Vol.2012-AL-138 No.6
2012/1/28

A polynomial time algorithm for bounded directed pathwidth

Hisao Tamaki[†1]

We give a polynomial time algorithm for bounded directed pathwidth. Given a positive integer k and a digraph G with n vertices and m edges, it runs in $O(mn^{k+1})$ time and constructs a directed path-decomposition of G of width at most k if one exists and otherwise reports the non-existence.

1. Introduction

According to Barát[3)], the notion of directed pathwidth of digraphs was introduced by Reed, Thomas, and Seymour around 1995. It is a generalization of pathwidth[18)], which is defined for undirected graphs, in the sense that if G is an undirected graph and G' is a digraph obtained from G by replacing each edge by a pair of directed edges in both directions, then the directed pathwidth of G' equals the pathwidth of G.

Following the tremendous success of the notion of treewidth[19)] of undirected graphs, as a key tool for the graph minor theory[20)] and for designing efficient algorithm[2)], several authors have proposed extensions of this notion to digraphs. Johnson, Robertson, Seymour, and Thomas introduced directed treewidth[12)], and showed that some NP-hard problems on digraphs including the directed Hamilton cycle problem can be solved in polynomial time if the given digraph has bounded directed treewidth. Since then, several variants of directed treewidth have been proposed: D-width[21)], DAG-width[7),17)], and Kelly-width[11)]. It is the subject of ongoing active research to compare respective power of these variants and other related digraph measures[10)].

†1 明治大学
Meiji University

In contrast, the extension of the notion of pathwidth to digraphs seems stable. Only one parameter, the directed pathwidth, has been proposed, which enjoys several equivalent formulations just as undirected treewidth and pathwidth do.

Although the applicability of these digraph parameters in designing efficient algorithms is provably limited in the sense that directed graph counterparts of some fixed parameter tractable problems on undirected graphs are hard to solve when parameterized by these width parameters[15)], they are nonetheless fundamental digraph parameters that deserve further explorations for algorithmic applications. For example, in[22)], the present author used directed pathwidth in a heuristic algorithm for exactly identifying the set of attractors of a given boolean network and experimentally showed the effectiveness of the approach.

Since it is NP-complete to decide, given a positive integer k and an undirected graph G, whether the pathwidth of G is at most k[13)], the same holds for the directed pathwidth. The situation is quite different between these problems if k is fixed. In this case, the problem of deciding if an undirected graph G has pathwidth at most k (and of constructing the associated path-decomposition) can be solved in linear time[5),6)]. In contrast, no polynomial time algorithm for fixed k (even for $k = 2$) was previously known, that decides whether the directed pathwidth of a given digraph is at most k.

In the undirected case, the fact that there is a polynomial time algorithm for fixed k that decides whether a given graph has pathwidth at most k is an immediate consequence of the graph minor theorem due to Robertson and Seymour[20)]: since the class of graphs with pathwidth k or smaller is closed under taking minors, that class is characterized by a fixed set of forbidden minors and therefore the membership to that class can be tested by checking if the given graph contains any of the forbidden minors. This does not hold for the directed case. Although the class of digraphs with directed pathwidth at most k, for any fixed k, is closed under taking minors, with a suitable definition of digraph minors[12)], no counterpart of the graph minor theorem is known for digraphs.

The standard algorithmic approach for undirected pathwidth for fixed k that leads to the linear time algorithm mentioned above is to first obtain a tree-decomposition of width $O(k)$ of the given graph and then perform a dynamic programming on this tree-decomposition to optimally solve the problem. There

Vol.2012-AL-138 No.6
2012/1/28

are again difficulties in extending this approach to the directed case. Although there is a fast approximation algorithm[12] to obtain a directed tree-decomposition of G of width $O(k)$, given that G has directed pathwidth at most k, directed tree-decompositions do not seem to support dynamic programming solutions to the problem of exactly determining the directed pathwidth. We may try to use a tree-decomposition of the underlying undirected graph, but since the treewidth of the underlying undirected graph is not bounded by any function of the directed pathwidth of the original digraph, we do not obtain a time bound that is polynomial in the size of the digraph even if the directed pathwidth is bounded.

In this paper, we show that the directed pathwidth problem for fixed k can be solved in polynomial time. We denote the directed pathwidth of digraph G by $\mathrm{dpw}(G)$.

Theorem 1.1 Given a positive integer k and a digraph G of n vertices and m edges, it can be decided in $O(mn^{k+1})$ time whether $\mathrm{dpw}(G) \leq k$. Moreover, if $\mathrm{dpw}(G) \leq k$, a directed path-decomposition of width at most k can be constructed in the same amount of time.

Our algorithm is based on a lemma (Lemma 3.1), which enables us to prune the natural search tree of factorial size into one of polynomial size. This lemma, which we call the commitment lemma, asserts that if a descendant of a node satisfies certain conditions then all other descendants of the node in the same generation can be safely removed from the search tree.

Our algorithm is extremely simple and easy to implement. We remark that even for undirected pathwidth, for which a fixed parameter linear-time algorithm is known[6], our algorithm is a strong alternative for practical use, as the linear-time algorithm depends exponentially on k^3 and is considered highly impractical. To the best of the present author's knowledge, an explicit and implementable $n^{O(k)}$ time algorithm has been known for treewidth[1] but not for pathwidth for general fixed k. We also remark that, even for the ranges of n and k where the time bound in Theorem 1.1 is practically useless, the commitment lemma would be useful in designing heuristic algorithms.

The rest of this paper is organized as follows. After some preliminaries in Section 2, we describe some basic ideas underlying the pruning of search trees in Section 3, assuming the commitment lemma. The proof of this lemma is given in Section 4. Section 5 provides some details of the algorithm which are necessary to establish the exact running time bound stated in Theorem 1.1.

This work was originally reported in[23].

2. Preliminaries

Let G be a digraph. For each subset U of $V(G)$, we denote by $N^-_G(U)$ the set of in-neighbors of U, i.e., $N^-_G(U) = \{v \in V(G) \setminus U \mid \exists u \in U : (v,u) \in E(G)\}$, and $d^-_G(U) = |N^-_G(U)|$ the number of in-neighbors of U.

Rather than giving the standard definition of the directed pathwidth, we use an alternative formulation called the directed vertex separation number, defined below.

We call a sequence σ of vertices of G *non-duplicating* if each vertex of G occurs at most once in σ. We denote by $\Sigma(G)$ the set of all non-duplicating sequences of vertices of G. For each sequence $\sigma \in \Sigma(G)$, we denote by $V(\sigma)$ the set of vertices constituting σ and by $|\sigma| = |V(\sigma)|$ the length of σ. For brevity, we write $d^-_G(\sigma)$ and $N^-_G(\sigma)$ for $d^-_G(V(\sigma))$ and $N^-_G(V(\sigma))$, respectively.

For each pair of sequences $\sigma, \tau \in \Sigma(G)$ such that $V(\sigma) \cap V(\tau) = \emptyset$, we denote by $\sigma\tau$ the sequence in $\Sigma(G)$ that is σ followed by τ. If $\sigma' = \sigma\tau$ for some τ, then we say that σ is a *prefix* of σ' and that σ' is an *extension* of (or *extends*) σ; we say that σ is a *proper* prefix of σ' and that σ' is a *proper* extension of σ if τ is nonempty. For each non-empty sequence $\sigma \in \Sigma(G)$, we denote by $\pi(\sigma)$ the prefix of σ with length $|\sigma| - 1$.

For $\sigma, \tau \in \Sigma(G)$, we say σ is a *subsequence* of τ if $V(\sigma) \subseteq V(\tau)$ and, for each pair of distinct vertices u and v in $V(\sigma)$, u occurs before v in σ if and only if u occurs before v in τ.

Let G be a digraph and k a positive integer. We say $\sigma \in \Sigma(G)$ is *k-feasible* for G if $d^-_G(\sigma') \leq k$ for every prefix σ' of σ. We may drop the reference to G and say σ is k-feasible when G is clear from the context.

Definition 2.1 The *directed vertex separation number* of digraph G, denoted by $\mathrm{dvsn}(G)$, is the minimum integer k such that there is a k-feasible sequence $\sigma \in \Sigma(G)$ with $V(\sigma) = V(G)$.

Note that, because of the equivalence of the directed vertex separation number to the directed pathwidth stated below, this parameter is invariant under the simultaneous reversal of all the edges.

It is known that $\mathrm{dvsn}(G) = \mathrm{dpw}(G)$ for every digraph G[24] (see also[14] for the undirected case) and the conversions between the sequences achieving the directed vertex separation number and the optimal directed path decompositions are simple. In particular, the conversion from the former to the latter can be done in $O(m+kn)$ time, where $n = |V(G)|$, $m = |E(G)|$, and $k = \mathrm{dpw}(G)$. Based on these equivalence and conversion, we focus on computing the directed vertex separation number and the corresponding sequence in the following sections.

3. Search tree pruning

Let digraph G be fixed and let $n = |V(G)|$. A straightforward exponential time algorithm for deciding if $\mathrm{dvsn}(G) \leq k$ constructs a search tree in which each node at level i of the tree is a k-feasible sequence of length i and the parent of a non-empty sequence σ is $\pi(\sigma)$, the prefix of σ with length $|\sigma| - 1$. We show that this search tree can be pruned into one with $O(n^{k+1})$ nodes.

The key to this pruning is the notion of non-expanding extensions. We say that an extension τ of $\sigma \in \Sigma(G)$ is *non-expanding* if τ is a proper extension of σ and $d^-_G(\tau) \leq d^-_G(\sigma)$. Suppose σ is k-feasible and has an immediate non-expanding extension σv, where $v \in V(G) \setminus V(\sigma)$. Then it appears plausible to hope that committing to this child of σ in the search tree, discarding all the other children, is safe in the sense that if σ has a k-feasible extension of length n then so does σv and therefore we do not lose completeness of the search through this commitment. The following lemma states that this hope is true in a more general manner: we may safely commit not only to an immediate non-expanding extension but also to any shortest non-expanding extension. We say that an element of $\Sigma(G)$ is *strongly k-feasible* if it has a k-feasible extension of length n.

Lemma 3.1 (Commitment Lemma) Let σ be a strongly k-feasible sequence in $\Sigma(G)$ and let τ be a shortest non-expanding k-feasible extension of σ, that is,
(1) $d^-_G(\tau) \leq d^-_G(\sigma)$, and
(2) $d^-_G(\tau') > d^-_G(\sigma)$ for every k-feasible proper extension τ' of σ with $|\tau'| < |\tau|$.
Then, τ is strongly k-feasible.
The proof of this lemma is given in the next section.

In the following, we assume a fixed total ordering $<$ on $V(G)$ and use a standard lexicographic ordering $<$ on $\Sigma(G)$ based on this total ordering. Let σ and τ be sequences of equal length in $\Sigma(G)$. We say that σ is *preferable to* τ, if either $d^-_G(\sigma) < d^-_G(\tau)$ or $d^-_G(\sigma) = d^-_G(\tau)$ and $\sigma < \tau$. Clearly, this preferable-to relation is a total ordering on the subset of $\Sigma(G)$ consisting of sequences of length i, for each $0 \leq i \leq n$.

Let σ and τ be k-feasible sequences of equal length. We say that σ *suppresses* τ, if σ and τ has a common prefix σ' such that σ is a shortest non-expanding k-feasible extension of σ' and σ is preferable to τ.

Proposition 3.1 Let σ, τ, and η be k-feasible sequences of equal length. If σ suppresses τ and τ suppresses η, then σ suppresses η.

Proof: Under the assumptions of the lemma, σ is preferable to η, since σ is preferable to τ and τ is preferable to η. Therefore, it suffices to show that σ and η has a common prefix α such that σ is a shortest non-expanding k-feasible extension of α.

Since σ suppresses τ, there is a common prefix β of σ and τ such that σ is a shortest non-expanding k-feasible extension of β. Similarly, there is a common prefix γ of τ and η such that τ is a shortest non-expanding k-feasible extension of γ. Since both β and γ are prefixes of τ, one is a prefix of the other. If β is a prefix of γ, then we are done with $\alpha = \beta$. If γ is a prefix of β, then γ is a common prefix of σ and η. Since σ is preferable to τ, we have $d^-_G(\sigma) \leq d^-_G(\tau)$. This, together with the assumption that τ is a shortest non-expanding k-feasible extension of γ implies that σ is also a shortest non-expanding k-feasible extension of γ. We are done with $\alpha = \gamma$. []

It should be intuitively clear that suppressed sequences are not necessary in the search tree, as a consequence of the commitment lemma. To formalize this intuition, we define the set S_i of unsuppressed k-feasible sequences of length i, for each $0 \leq i \leq n$, inductively as follows.
(1) S_0 consists of the empty sequence.
(2) A k-feasible sequence σ of length $i > 0$ is in S_i if and only if $\pi(\sigma) \in S_{i-1}$ and there is no k-feasible sequence τ of length i such that $\pi(\tau) \in S_{i-1}$ and τ suppresses σ.

Lemma 3.2 If there is a k-feasible sequence of length n in $\Sigma(G)$, then there is at least one such sequence in S_n.

Proof: For each k-feasible sequence σ of length n, let i_σ denote the largest i, $0 \le i \le n$, such that the prefix of σ of length i is in S_i. If there is some k-feasible σ of length n with $i_\sigma = n$, then we are done. So, suppose otherwise and fix k-feasible σ of length n so that i_σ is the largest over all choices of σ. Let σ' be the prefix of σ of length $i_\sigma + 1$. Then, since $\sigma' \notin S_{i_\sigma+1}$ and $\pi(\sigma') \in S_{i_\sigma}$, σ' must be suppressed by some k-feasible sequence τ of length $i_\sigma+1$ such that $\pi(\tau) \in S_{i_\sigma}$. Choose τ so that it is the most preferable among all the candidates. Then, τ is not suppressed by any τ' with $\pi(\tau') \in S_{i_\sigma}$, since otherwise τ' suppresses σ' by Proposition 3.1 and is preferable to τ, contradicting the choice of τ. Therefore $\tau \in S_{i_\sigma+1}$. But since τ is a shortest non-expanding k-feasible extension of some prefix of σ', which is strongly k-feasible because of its extension σ, τ is strongly k-feasible by Lemma 3.1. This contradicts the choice of σ, since $i_\eta \ge i_\sigma + 1$, where η is a k-feasible extension of τ with length n. []

Thus, in our pruned search, we need only to generate k-feasible sequences in S_i, for $1 \le i \le n$.

To analyze the size of each set S_i, we assign a *signature* $\mathrm{sgn}(\sigma) \in \Sigma(G)$ to each k-feasible sequence $\sigma \in \Sigma(G)$ as follows. Call a non-expanding k-feasible extension τ of σ *locally shortest*, if no proper prefix of τ is a non-expanding extension of σ. We define $\mathrm{sgn}(\sigma)$ inductively as follows.

(1) If σ is empty then $\mathrm{sgn}(\sigma)$ is empty.

(2) If σ is non-empty and is a locally shortest non-expanding extension of some prefix of σ, then $\mathrm{sgn}(\sigma) = \mathrm{sgn}(\tau)$, where τ is the shortest prefix of σ such that σ is a locally shortest non-expanding k-feasible extension of τ.

(3) Otherwise $\mathrm{sgn}(\sigma) = \mathrm{sgn}(\pi(\sigma))v$, where v is the last vertex of σ (and hence $\sigma = \pi(\sigma)v$).

Proposition 3.2 For each k-feasible sequence $\sigma \in \Sigma(G)$, we have $|\mathrm{sgn}(\sigma)| \le d^-_G(\sigma)$.

Proof: The proof is by induction on the length of σ. The base case where σ is empty is trivial. Suppose rule 2 of the definition of signatures applies to σ: $\mathrm{sgn}(\sigma) = \mathrm{sgn}(\tau)$, where τ is the shortest prefix of σ such that σ is a locally shortest non-expanding k-feasible extension of τ. If $d^-_G(\tau) = d^-_G(\sigma)$ then we are done, since we have $|\mathrm{sgn}(\sigma)| = |\mathrm{sgn}(\tau)| \le d^-_G(\tau)$ by the induction hypothesis. So suppose $d^-_G(\tau) > d^-_G(\sigma)$. Let τ' be the shortest prefix of τ such that $d^-_G(\tau'') = d^-_G(\tau)$ for every prefix τ'' of τ that is an extension of τ', including τ' itself. Then, we have $\mathrm{sgn}(\sigma) = \mathrm{sgn}(\tau) = \mathrm{sgn}(\tau')$ by a repeated application of rule 2. Since $d^-_G(\tau') > 0$, τ' is non-empty and we have $d^-_G(\pi(\tau')) < d^-_G(\sigma)$ since σ is not a locally shortest non-expanding extension of $\pi(\tau')$ by the choice of τ. In this case, τ' cannot be a locally shortest non-expanding extension of any of its prefixes because $d^-_G(\pi(\tau')) < d^-_G(\tau)$. Therefore, rule 3 applies to τ' and we have $|\mathrm{sgn}(\tau')| = |\mathrm{sgn}(\pi(\tau'))| + 1 \le d^-_G(\pi(\tau')) + 1$ by the induction hypothesis and therefore $|\mathrm{sgn}(\sigma)| = |\mathrm{sgn}(\tau')| \le d^-_G(\sigma)$. Finally suppose that rule 3 applies to σ: $\mathrm{sgn}(\sigma) = \mathrm{sgn}(\pi(\sigma))v$, where v is the last vertex of σ. Since σ is not a non-expanding extension of $\pi(\sigma)$, we have $d^-_G(\sigma) > d^-_G(\pi(\sigma))$ and therefore $|\mathrm{sgn}(\sigma)| \le d^-_G(\sigma)$ follows from the induction hypothesis on $\pi(\sigma)$. []

The following observation is straightforward.

Proposition 3.3 Let σ be a k-feasible sequence of length i that belongs to S_i. Then $v \in V(\sigma)$ does not appear in $\mathrm{sgn}(\sigma)$ if and only if there are prefixes σ_1 and σ_2 of σ such that $v \notin V(\sigma_1)$, $v \in V(\sigma_2)$, and σ_2 is a locally shortest non-expanding k-feasible extension of σ_1.

Lemma 3.3 Let i, $1 \le i \le n$, be arbitrary. If σ and τ are distinct elements of S_i then neither $\mathrm{sgn}(\sigma)$ nor $\mathrm{sgn}(\tau)$ is a prefix of the other.

Proof: Let $\sigma, \tau \in S_i$ be distinct. For each j, $0 \le j \le i$, let σ_j (τ_j, resp.) denote the prefix of σ (τ, resp.) of length j. Let j_0 be the smallest integer such that $\sigma_{j_0} \ne \tau_{j_0}$. Let u_0 be the last vertex of σ_{j_0} and v_0 the last vertex of τ_{j_0}. We claim that there is no pair of integers j_1 and j_2 such that $0 \le j_1 < j_0 \le j_2 \le i$ and σ_{j_2} is a locally shortest non-expanding extension of σ_{j_1}. To see this, suppose such a pair of integers j_1 and j_2 exists. If there is a non-expanding k-feasible extension of σ_{j_1} shorter than σ_{j_2} then this extension is not a prefix of σ_{j_2} since σ_{j_2} is a locally shortest non-expanding k-feasible extension of σ_{j_1}. But this is impossible because then a prefix of σ would be suppressed and σ would not be in S_i. Therefore, σ_{j_2} is a shortest non-expanding k-feasible extension of σ_{j_1}. Since σ_{j_1} is a common prefix of σ_{j_2} and τ_{j_2}, τ_{j_2} is suppressed by σ_{j_2} if σ_{j_2} is preferable to τ_{j_2}. On the other hand, if τ_{j_2} is preferable to σ_{j_2}, then $d^-_G(\tau_{j_2}) \le d^-_G(\sigma_{j_2}) \le d^-_G(\sigma_{j_1})$ and, noting that $\sigma_{j_1} = \tau_{j_1}$ because $j_1 < j_0$, we see that τ_{j_2} is also a shortest non-expanding k-feasible extension of σ_{j_1} and hence suppresses σ_{j_2}. In either case, we have a contradiction to the fact that both σ_{j_2} and τ_{j_2} are in S_{j_2}. This verifies the claim that there is no such pair j_1, j_2.

It follows from this claim and Proposition 3.3 that:

(1) u_0 appears in $\mathrm{sgn}(\sigma)$ and
(2) each vertex in $V(\sigma_{j_0-1})$ appears in $\mathrm{sgn}(\sigma)$ if and only if it appears in $\mathrm{sgn}(\sigma_{j_0-1})$.

Similarly, we have:

(1) v_0 appears in $\mathrm{sgn}(\tau)$ and
(2) each vertex in $V(\tau_{j_0-1})$ appears in $\mathrm{sgn}(\tau)$ if and only if it appears in $\mathrm{sgn}(\tau_{j_0-1})$.

Thus, $\mathrm{sgn}(\sigma)$ and $\mathrm{sgn}(\tau)$ have a common prefix $\mathrm{sgn}(\sigma_{j_0-1}) = \mathrm{sgn}(\tau_{j_0-1})$, which is followed by u_0 in $\mathrm{sgn}(\sigma)$ and by v_0 in $\mathrm{sgn}(\tau)$. Since $u_0 \neq v_0$, neither $\mathrm{sgn}(\sigma)$ nor $\mathrm{sgn}(\tau)$ is a prefix of the other. [] Our desired bound on $|S_i|$ immediately follows from this lemma and Proposition 3.2.

Corollary 3.1 $|S_i| \leq n^k$ holds for $0 \leq i \leq n$.

From this corollary, it is clear that the directed pathwidth problem can be solved in $n^{k+O(1)}$ time. Some implementation details needed to obtain the specific time bound stated in Theorem 1.1 are given in Section 5.

4. Proof of the commitment lemma

The following observation that the function d^-_G is submodular is straightforward. For self-containedness, we include a proof.

Proposition 4.1 Let G be a digraph and let X and Y be two arbitrary subsets of $V(G)$. Then, we have

$$d^-_G(X) + d^-_G(Y) \geq d^-_G(X \cap Y) + d^-_G(X \cup Y). \quad (1)$$

Proof: For each vertex $v \in V(G)$, we show that the number of times v is counted in the right-hand side of (1) does not exceed the number of times it is counted in the left-hand side of (1). If v is counted both in $d^-_G(X \cap Y)$ and $d^-_G(X \cup Y)$ then $v \notin X \cup Y$ and v has an out-neighbor in $X \cap Y$ and, therefore, v is counted both in $d^-_G(X)$ and $d^-_G(Y)$. If v is counted in $d^-_G(X \cup Y)$ then $v \notin X \cup Y$ and v has an out-neighbor in $X \cup Y$ and, therefore, v is counted either in $d^-_G(X)$ or in $d^-_G(Y)$. If v is counted in $d^-_G(X \cap Y)$ then either $v \notin X$ or $v \notin Y$ and v has an out-neighbor in $X \cap Y$ and, therefore, v is counted either in $d^-_G(X)$ or in $d^-_G(Y)$. []

Lemma 3.1 is a direct consequence of the following two lemmas.

Lemma 4.1 Let G be a directed graph and k a positive integer. Let σ be a strongly k-feasible sequence in $\Sigma(G)$ and τ a k-feasible proper extension of σ such that $d^-_G(X) \geq d^-_G(\tau)$ for every X with $V(\sigma) \subseteq X \subseteq V(\tau)$. Then, τ is strongly k-feasible.

Proof: Let σ and τ be as in the statement of the lemma and σ' a k-feasible extension of σ of length n. Let α be the subsequence of σ' such that $V(\alpha) = V(G) \setminus V(\tau)$. Let $\tau' = \tau\alpha$. Note that $\tau' \in \Sigma(G)$ and $V(\tau') = V(G)$. We claim that τ' is k-feasible and therefore τ is strongly k-feasible.

Since the prefix τ of τ' is k-feasible, we only need to show that, for $1 \leq i \leq |\alpha|$, $d^-_G(V(\tau) \cup V_i(\alpha)) \leq k$, where we denote by $V_i(\alpha)$ the set of first i vertices of α.

For each i, $1 \leq i \leq |\alpha|$, let σ_i denote the minimal prefix of σ' such that $V(\sigma_i) \setminus V(\tau) = V_i(\alpha)$. Since each member of σ precedes each member of α in σ', σ is a prefix of σ_i for $1 \leq i \leq |\alpha|$. Fix i, $1 \leq i \leq |\alpha|$. By the submodularity of d^-_G, we have

$$d^-_G(\tau) + d^-_G(\sigma_i) \geq d^-_G(V(\tau) \cap V(\sigma_i)) + d^-_G(V(\tau) \cup V(\sigma_i)).$$

Since σ' is k-feasible, we have $d^-_G(\sigma_i) \leq k$. By the assumption on τ in the statement of the lemma, we also have $d^-_G(V(\tau) \cap V(\sigma_i)) \geq d^-_G(\tau)$ as $V(\sigma) \subseteq V(\tau) \cap V(\sigma_i) \subseteq V(\tau)$. Therefore we have $d^-_G(V(\tau) \cup V_i(\alpha)) = d^-_G(V(\tau) \cup V(\sigma_i)) \leq k$, which proves the claim. []

Lemma 4.2 Let G be a directed graph and k a positive integer. Let σ be a k-feasible sequence in $\Sigma(G)$ and τ a shortest non-expanding k-feasible extension of σ. Then, for every X such that $V(\sigma) \subseteq X \subseteq V(\tau)$, we have $d^-_G(X) \geq d^-_G(\tau)$.

Proof: Suppose to the contrary that there is some X, $V(\sigma) \subseteq X \subseteq V(\tau)$, such that $d^-_G(X) < d^-_G(\tau)$. Since $d^-_G(\sigma) \geq d^-_G(\tau)$, we have $V(\sigma) \subsetneq X \subsetneq V(\tau)$. We show that there is some non-expanding k-feasible extension η of σ that is shorter than τ. This contradicts the assumption that τ is a shortest such extension, and therefore we will be done.

Let α be the subsequence of τ such that $V(\alpha) = X$. Note that α extends σ since $V(\sigma) \subseteq X$. Let h be an integer, $|\sigma| < h \leq |X|$, such that $d^-_G(V_h(\alpha))$ is the largest, where we denote by $V_h(\alpha)$ the set of first h vertices of α. If $d^-_G(V_h(\alpha)) \leq k$ then α is k-feasible and we are done with $\eta = \alpha$: $|\alpha| < |\tau|$ holds since $V(\alpha) = X$ is a proper subset of $V(\tau)$.

Suppose $d^-_G(V_h(\alpha)) > k$. Since $d^-_G(X) < k$, we have $h < |X|$. For each i, $0 \leq i \leq X$, let τ_i denote the minimal prefix of τ such that $V(\tau_i) \cap X = V_i(\alpha)$. Since $V(\sigma) \subseteq X$, we have $\tau_{|\sigma|} = \sigma$.

We set $\eta = \tau_h\alpha'$, where α' is the subsequence of α consisting of its last $|X| - h$ elements, and verify that η is a non-expanding k-feasible extension of σ and is shorter than τ. Let i be an integer, $h \leq i \leq |X|$. By the submodularity of d^-_G, we have

$$d^-_G(\tau_h) + d^-_G(V_i(\alpha)) \geq d^-_G(V_h(\alpha)) + d^-_G(V(\tau_h) \cup V_i(\alpha)), \tag{2}$$

where we have used $V(\tau_h) \cap V_i(\alpha) = V_h(\alpha)$. We have $d^-_G(V_i(\alpha)) \leq d^-_G(V_h(\alpha))$ by the choice of h and moreover $d^-_G(\tau_h) \leq k$ since τ is k-feasible. Therefore, we have $d^-_G(V(\tau_h) \cup V_i(\alpha)) \leq k$. Since this holds for every i, $h \leq i \leq |X|$, η is k-feasible. Since $d^-_G(\tau_h) \leq k < d^-_G(V_h(\alpha))$, (2) also implies $d^-_G(V(\tau_h) \cup V_i(\alpha)) < d^-_G(V_i(\alpha))$. Letting $i = |X|$, we have $d^-_G(\eta) = d^-_G(V(\tau_h) \cup V(\alpha)) < d^-_G(\alpha) = d^-_G(X) < d^-_G(\tau) \leq d^-_G(\sigma)$. Thus, η is a non-expanding extension of σ. Finally, the inclusion $V(\eta) \subseteq V(\tau)$ and the strict inequality $d^-_G(\eta) < d^-_G(\tau)$ imply that η is shorter than τ. []

Proof: (of Lemma 3.1.) Let σ be a strongly k-feasible sequence in $\Sigma(G)$ and τ a shortest non-expanding k-feasible extension of σ. Then, by Lemma 4.2, we have $d^-_G(X) \geq d^-_G(\tau)$ for every X such that $V(\sigma) \subseteq X \subseteq V(\tau)$. Lemma 4.1 applies and τ is strongly k-feasible. []

5. Implementation details

In this section, we verify that our algorithm can be implemented to run in the time bound of $O(mn^{k+1})$ stated in Theorem 1.1, where $n = |V(G)|$ and $m = |E(G)|$. We assume that G is strongly connected and hence $m \geq n$.

Data structures

We represent each nonempty sequence $\sigma \in \Sigma(G)$ by a pair consisting of the last vertex of σ and a pointer to the prefix $\pi(\sigma)$ of σ of length $|\sigma| - 1$. Thus, the elements of the sets S_i, $0 \leq i \leq i$, naturally form a rooted tree in which the parent of each non-root node σ is $\pi(\sigma)$ and the set of nodes at the ith level is S_i. In addition, we represent the set S_i, for each $0 \leq i \leq n$, as a list sorted in the lexicographic ordering.

We assume the input digraph G is given in the in-neighbor list representation: each vertex v has a list in(v) of its in-neighbors ordered in the assumed total ordering $<$ on $V(G)$.

Constructing immediate extensions

In this step, we generate k-feasible extensions of each element of S_{i-1} and let the set of all those extensions be T_i. Let σ be an element of S_{i-1} being processed. We first construct the bit-vector representation of $N^-_G(\sigma)$ in $O(n)$ time. Then, we iterate through all the vertices in $V(G)$. If $v \in V(G)$ is not in σ, we compute $d^-_G(\sigma v)$ in $O(d^-_G(v))$ time, using the bit-vector for $N^-_G(\sigma)$. If $d^-_G(\sigma v) \leq k$ then we add σv to our list of feasible extensions. Doing this for all elements of S_{i-1} in the sorted order, we obtain the set T_i in the form of a sorted list. The time required for this step is $O(mn^k)$.

Identifying shortest non-expanding k-feasible extensions and inheritors

In this step, for each pair (τ, σ) such that $\sigma \in T_i$ and σ is the most preferable shortest non-expanding k-feasible extension of τ, we register σ as the *inheritor* of τ.

We first observe that $\sigma \in T_i$ can be a shortest non-expanding k-feasible extension of some proper prefix of σ only if $d^-_G(\sigma) \leq d^-_G(\pi(\sigma))$. Moreover, for each $\eta \in S_{i-1}$, among the extensions of η in T_i satisfying $d^-_G(\sigma) \leq d^-_G(\eta)$, only the most preferable one can be the most preferable shortest non-expanding k-feasible extension of some sequence. Based on this observation, we collect, for each $\eta \in S_{i-1}$, at most one extension $\sigma \in T_i$ of η: σ satisfies $d^-_G(\sigma) \leq d^-_G(\eta)$ and is the most-preferable over all extensions of η in T_i. We let the resulting set T'_i and scan its elements in the lexicographic ordering.

Let σ be an element of T'_i. For each proper prefix τ of σ, σ is a shortest non-expanding k-feasible extension of τ if and only if σ is a locally shortest non-expanding k-feasible extension of τ. The "only if" part is obvious. For the "if" part, suppose τ has a non-expanding k-feasible extension τ' that is shorter than σ but is not a prefix of σ. We assume τ' is the shortest among such and hence is a shortest non-expanding k-feasible extension of τ. Let τ'' be a prefix of σ of length $|\tau'|$. Since the presence of $\pi(\sigma)$ in S_{i-1} implies that τ' does not suppress τ'', it must hold that $d^-_G(\tau'') \leq d^-_G(\tau') \leq d^-_G(\tau)$ and therefore σ is not a locally shortest non-expanding k-feasible extension of τ. Since $d^-_G(\tau)$ has been calculated for every $\tau \in \bigcup_{j \leq i} S_j$, the above condition can be tested in $O(n)$ total time for all prefixes τ of σ.

When we find a prefix τ of σ such that σ is a shortest k-feasible non-expanding extension of τ, we check whether the inheritor of τ is already registered. If not, then register σ as such. Otherwise, let σ' be the registered extension. If $d^-_G(\sigma) < d^-_G(\sigma')$ then we replace σ' with σ; otherwise we retain σ'. Since we are processing the elements of T'_i in the lexicographic order, the registered inheritor is correctly the most-preferable shortest k-feasible non-expanding extension after all the elements of T'_i are processed. The time required for this registering process is also $O(n)$ for each $\sigma \in T'_i$. The overall processing time for this step is $O(n^{k+1})$.

Filtering out suppressed elements

In this step, we collect those elements of T_i that are not suppressed, obtaining the set S_i.

Let $\eta \in S_{i-1}$. Suppose first that η does not have an extension in T'_i, that is, $d^-_G(\sigma) > d^-_G(\eta)$ for every extension σ of η in T_i. In this case, if some prefix of η has some inheritor registered then all extensions of η in T_i are suppressed; otherwise, no prefix of η has a non-expanding k-feasible extension in T_i and therefore none of the extensions of η in T_i is suppressed. Suppose next that η has an extension σ in T'_i (which is unique). Then all extensions of η in T_i but σ are suppressed by σ. This extension σ is suppressed if and only if some prefix of η has an inheritor other than σ registered.

In either case, the processing time for each $\eta \in S_{i-1}$ is $O(n)$ and therefore the total time for this step is $O(n^{k+1})$.

Overall running time

We repeat the above construction of S_i for $i = 1, 2, \ldots, n$ in $O(mn^{k+1})$ total time. Checking whether S_n is empty is trivial. If it is not empty, any element of S_n achieves the directed vertex separation number at most k.

6. Concluding remarks

In the terminology of parameterized complexity theory[8),9),16)], the result of this paper puts the problem of deciding the directed pathwidth in class XP. It is open whether it is in FPT, that is, if there is an algorithm that, given positive integer k and digraph G, decides if $\mathrm{dpw}(G) \leq k$ in time $f(k)n^{O(1)}$ where f is some function independent of n.

It was pointed out by Sang-il Oum and by Hiroshi Nagamochi that the commitment lemma holds in a more general setting, where the in-degree function d^-_G is replaced by an arbitrary submodular function, and thus may be useful in other contexts. Exploring such applications of the lemma and the techniques in this work is also an attractive avenue of future research.

Acknowledgment

The author would like to thank Yuichiro Miyamoto, Ryuhei Uehara, Hirotaka Ono, Takehiro Ito, Katsuhisa Yamanaka, Yasuaki Kobayashi, and Fumihito Ohtaki for useful discussions. Thanks are also due to Hiroshi Nagamochi who read an earlier manuscript carefully and helped improve the presentation.

References

1) S.Arnborg, D.Corneil and A.Proskurowski. Complexity of finding embeddings in a k-tree. SIAM Journal on Matrix Analysis and Applications 8(2): 277-284.
2) S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. Discrete Applied Mathematics 23 (1): 11-24,
3) J.Barát. Directed path-width and monotonicity in digraph searching. Graphs and Combinatorics 22(2):161–172, 2006.
4) D.Berwanger, A.Dawar, P.Hunter, and S.Kreutzer DAG-Width and Parity Games. In *Proc. 23rd Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 3884, 524-536, 2006.
5) H.L.Bodlaender and T.Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of Graphs. Journal of Algorithms 21: 358–402, 1996.
6) H.L.Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing 25(6): 1305–1317, 1996.
7) D.Berwanger, A.Dawar, P.Hunter, and S.Kreutzer DAG-Width and Parity Games. In *Proc. 23rd Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 3884, 524-536, 2006.
8) R.G.Downey and M.R.Fellows. *Parameterized Complexity*, Springer, 1999.

9) J.Flum and M.Grohe. *Parameterized Complexity Theory*, Springer, 2006.
10) R.Ganian, P.Hliněný, J. Kneis, A.Langer, J.Obdržálek, and P.Rossmanith. On Digraph Width Measures in Parameterized Algorithmics. In *Proc. the 4th International Workshop on Parameterized and Exact Computation*, LNCS 5917, 185-197, 2009.
11) P.Hunter and St.Kreutzer. Digraph Measures: Kelly Decompositions, Games, and Orderings. In *Proc. the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 637-644, 2007.
12) T.Johnson, N.Robertson, P.D.Seymour and R.Thomas. Directed tree-width. Journal of Combinatorial Theory Series B 82(1): 138–154, 2001.
13) T. Kashiwabara and T. Fujisawa. NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph. In *Proc. International Symposium on Circuits and Systems* 657–660, 1979.
14) N.G.Kinnersley. The vertex separation number of a graph equals its path-width. Information Processing Letters 42:345-350, 1992.
15) M. Lampis, G. Kaouri, and V. Mitsou. On the Algorithmic Effectiveness of Digraph Decompositions and Complexity Measures. In Proc. of 19th International Symposium on Algorithms and Computation, 220–231, 2008.
16) R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.
17) J.Obdržálek. DAG-width - Connectivity Measure for Directed Graphs. In *Proc. the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 814-821, 2006.
18) N.Robertson and P.Seymour. Graph minors. I. Excluding a forest. Journal of Combinatorial Theory, Series B 35(1): 39–61, 1983.
19) N.Robertson and P.Seymour. Graph minors III: Planar tree-width. Journal of Combinatorial Theory, Series B 36(1): 49–64, 1984.
20) N.Robertson and P.Seymour. Graph Minors. XX. Wagner's conjecture. Journal of Combinatorial Theory, Series B 92(2): 325–35, 2004.
21) M.A.Safari. D-width: A more natural measure for directed tree width. In *Proc. the 30th International Symposium on Mathematical Foundations of Computer Science*, LNCS 3618, 745-756, Springer-Verlag, 2005.
22) H.Tamaki. A directed path-decomposition approach to exactly identifying attractors of boolean networks. In *Proc. 10th International Symposium on Communications and Information Technologies*, 844-849, 2010.
23) H.Tamaki. A polynomial time algorithm for bounded directed pathwidth. In *Proc. 37th International Workshop on Graph-Theoretic Concepts in Computer Science,* LNCS6986, 331-342, 2011.
24) B.Yang and Y.Cao. Digraph searching, directed vertex separation and directed pathwidth. Discrete Applied Mathematics 156(10):1822–1837, 2008.