Recognizing digraphs of Kelly-width 2

Daniel Meister and Jan Arne Telle and Martin Vatshelle
Department of Informatics, University of Bergen, 5020 Bergen, Norway

19th August 2008

Abstract

Kelly-width is a parameter of digraphs recently proposed by Hunter and Kreutzer as a directed analogue of treewidth. We give an alternative characterization of digraphs of bounded Kelly-width in support of this analogy, and the first polynomial-time algorithm recognizing digraphs of Kelly-width 2. For an input digraph $G = (V, A)$ the algorithm outputs a vertex ordering and a digraph $H = (V, B)$ with $A \subseteq B$ witnessing either that G has Kelly-width at most 2 or that G has Kelly-width at least 3, in time linear in H.

1 Introduction

The tractability of large classes of NP-complete problems when parameterized by the treewidth of the input graph counts among the strongest results in algorithmic graph theory. The algorithms behind this tractability have two stages: first an algorithm computing treewidth, then an algorithm solving the specific problem using the tree-structure discovered in the first stage. See for example [3] for a recent overview of these algorithms. For directed graphs (digraphs) there have been several proposals for a parameter analogous to treewidth: 'directed treewidth' of Johnson, Robertson, Seymour, Thomas [7], 'D-width' of Safari [10], 'DAG-width' of Berwanger, Dawar, Hunter, Kreutzer [2] and independently Obdržálek [9], and 'Kelly-width' of Hunter and Kreutzer [6]. Which of these proposed parameters is the better analogue of treewidth? In this paper we give evidence in support of the Kelly-width parameter.

The success of a model depends on a balance between the modeling power, which measures how general its domain of application is, and the analytical power, which measures how good it is as an analytical tool. The two are typically in conflict. This is also the case for the above proposals for tree-like parameters of digraphs. The better the modeling power, e.g. the larger the class of digraphs that have bounded parameter value, the worse the analytical power, e.g. the smaller the chance of successfully emulating both stages of the algorithmic results for treewidth. We do not go into details of the modeling and analytical powers of each of the proposed digraph parameters, but simply note that from a purely algorithmic point of view there is no clear winner. How then to choose the digraph parameter which is the most natural directed analogue of treewidth? Note that while some concepts of undirected graphs have unambiguous natural translations to directed graphs, e.g. from paths to directed paths, there are other concepts, e.g. cliques and separators, for which the translation is less clear. The treewidth parameter is known to have many equivalent characterizations. If we start with a characterization of treewidth that uses only concepts that have unambiguous translations to directed graphs then we should arrive at a directed graph parameter which is a natural analogue of treewidth. Hunter and Kreutzer did just this for an elimination process of undirected graphs that defines treewidth to obtain a characterization of digraphs of bounded Kelly-width. We establish a second such correspondence in Section 3, a characterization of digraphs of Kelly-width at most k that is an analogue of a characterization of treewidth [11, 4].

We also enhance the algorithmic argument in favour of Kelly-width. Digraphs of Kelly-width 1 are the directed acyclic graphs and recognizable by a simple algorithm. For all larger values of k the only algorithms that were known for recognizing digraphs of Kelly-width k had running time exponential in the size of the input digraph [6]. We present a fast algorithm recognizing digraphs of Kelly-width 2 in Section 4. For an input digraph $G = (V, A)$ this algorithm will output a vertex ordering and a digraph $H = (V, B)$ with $A \subseteq B$ witnessing either that G has Kelly-width at most 2 or that G has Kelly-width at least 3, in time linear in H. In the positive case the witness can be used to easily find a decomposition of the digraph into a tree-like structure. Section 5 discusses Kelly-width k recognition for $k > 2$ before some final remarks in Section 6.

2 Graph preliminaries and digraphs of bounded Kelly-width

We consider simple finite directed graphs, "digraphs" for short. A digraph is denoted as $G = (V, A)$ where V is the vertex set and A the arc set of G. In particular, G does not have loops. When we deal with undirected graphs, we will explicitly mention it. An arc of graph G is denoted as (u, v) with u an in-neighbor of v and v an out-neighbor of u. Further definitions are given when they are needed.

Hunter and Kreutzer introduced the notion of Kelly-width [6]. Kelly-width is a parameter for digraphs, and it is the least width of a so-called *Kelly-decomposition*. We will not define Kelly-decompositions here, since we will not use this notion. The authors gave several characterizations of digraphs of bounded Kelly-width by: elimination process, inductive construction, cops-and-robber game. We will study digraphs of bounded Kelly-width starting from the elimination process.

Undirected graphs of bounded treewidth have a nice characterization using an elimination process. Let x be a vertex of an undirected graph G. The operation *reducing G by x* yields graph G' that is obtained from G by deleting vertex x and adding an edge between each pair of non-adjacent neighbors of x, in other words, making its neighborhood into a clique.

Theorem 2.1 (folklore) *An undirected graph G has treewidth at most k if and only if G can be reduced to the empty graph by repeatedly reducing by a vertex of degree at most k.*

The operation of reducing by a vertex was translated by Hunter and Kreutzer into the world of digraphs as follows: *reducing digraph G by x* yields digraph G' that is obtained from G by deleting vertex x and adding arcs from all in-neighbors of x to all out-neighbors of x, but no loops or multiple edges. The following result will for our purposes serve as the definition of Kelly-width.

Theorem 2.2 ([6]) *A digraph G has Kelly-width at most $k + 1$ if and only if G can be reduced to the empty digraph by repeatedly reducing by a vertex of out-degree at most k.*

The elimination process of Theorem 2.2 defines a vertex ordering, where the first vertex is the vertex eliminated first. We call this an *elimination ordering* of width at most k. Note that it is implicit in Theorem 2.2 that for every digraph G and every subgraph H of G, the Kelly-width of H is at most the Kelly-width of G.

3 An analogy between Kelly-width and treewidth

In this section, we show that digraphs of bounded Kelly-width are the digraphs whose vertices can be arranged in a linear order to satisfy special conditions, without using the notion of vertex reductions. We start with a lemma that is implicit, but not proven, in [6]. When we speak of paths, we mean paths that visit a vertex at most once.

Lemma 3.1 *Let $G = (V, A)$ be a digraph, and let $X \subseteq V$. Reduce G by the vertices in X, taken in any order, and obtain H. Then, (v, w) is arc in H if and only if there exists a v, w-path in G using only vertices from $X \cup \{v, w\}$.*

Proof We show the lemma by induction over the cardinality of X. If $|X| = 0$ then the claim obviously holds, since G has a v, w-path using only vertices from $\emptyset \cup \{v, w\}$ if and only if $(v, w) \in A$. Now, let the claim be true for every subset of V of cardinality at most $k \geq 0$. Let $X \subseteq V$ where $|X| = k + 1$. Let $x \in X$. Obtain G' by reducing G by the vertices in $X \setminus \{x\}$. Obtain H by reducing G' by x. Let v, w be two vertices of H.

($\Rightarrow$) Let $(v, w) \in A(H)$. If $(v, w) \in A(G')$ then there is a v, w-path in G using only vertices from $X \cup \{v, w\}$ by induction hypothesis. Otherwise, by the definition of the reduction operation, $(v, x), (x, w) \in A(G')$. Due to induction hypothesis, there are v, x- and x, w-paths in G using only vertices from $(X \setminus \{x\}) \cup \{v, x\}$ and $(X \setminus \{x\}) \cup \{x, w\}$, respectively. Hence, there is a v, w-path in G using only vertices from $(X \setminus \{x\}) \cup \{v, x\} \cup \{x, w\} = X \cup \{v, w\}$ in G.

($\Leftarrow$) Let there be a v, w-path P in G using only vertices from $X \cup \{v, w\}$. If P does not contain x then P contains only vertices from $(X \setminus \{x\}) \cup \{v, w\}$, and $(v, w) \in A(G') \subseteq A(H)$ due to induction hypothesis. Otherwise, if x is a vertex on P, then there are v, x- and x, w-paths in G using only vertices from $(X \setminus \{x\}) \cup \{v, x\}$ and $(X \setminus \{x\}) \cup \{x, w\}$, respectively. By induction hypothesis, $(v, x), (x, w) \in A(G')$, and hence $(v, w) \in A(H)$.

Thus, the lemma follows. □

Note that Lemma 3.1 yields as a result that the graph obtained from reducing by a set of vertices does not depend on the actual order in which the reduction is executed.

The following characterization of undirected graphs of bounded treewidth was first given explicitly by Thorup [11] (and is implicit in the work of Dendris et al. [4]):

Lemma 3.2 ([11, 4]) *An undirected graph G has treewidth at most k if and only if it has a vertex ordering $\sigma = v_1, v_2, \ldots, v_n$ and, for each $1 \leq i \leq n$, a set $S_i \subseteq \{v_1, v_2, \ldots, v_{i-1}\}$ of at most k vertices such that there is no path in $G \setminus S_i$ from v_i to a vertex in $\{v_1, v_2, \ldots, v_{i-1}\}$.*

We prove an analogous result for digraphs and Kelly-width:

Lemma 3.3 *A digraph G has Kelly-width at most $k+1$ if and only if it has a vertex ordering $\sigma = v_1, v_2, \ldots, v_n$ and, for each $1 \leq i \leq n$, a set $S_i \subseteq \{v_1, v_2, \ldots, v_{i-1}\}$ of at most k vertices such that there is no path in $G \setminus S_i$ from v_i to a vertex in $\{v_1, v_2, \ldots, v_{i-1}\}$.*

Proof ($\Rightarrow$) Let G have Kelly-width at most $k + 1$. According to Theorem 2.2, there is an elimination ordering of width at most k for G; let $\sigma = v_1, \ldots, v_n$ be the reversal of such an ordering. Let $i \in \{1, \ldots, n\}$. Let P be a v_i, w-path for $w \in \{v_1, \ldots, v_{i-1}\}$. Let u be the first vertex from $\{v_1, \ldots, v_{i-1}\}$ on P. Let H be obtained from G by reducing by the vertices in $\{v_{i+1}, \ldots, v_n\}$. Due to Lemma 3.1, (v_i, u) is an arc in H. Let S_i be the set of out-neighbors of v_i in H. Due to Theorem 2.2 and Lemma 3.1, $|S_i| \leq k$. Then, there is no v_i, w-path for $w \in \{v_1, \ldots, v_{i-1}\}$ in $G \setminus S_i$.

($\Leftarrow$) Let $\sigma = v_1, \ldots, v_n$ be a vertex ordering for G and $S_1, \ldots, S_n$ the corresponding sets of vertices satisfying the claim. We show that $v_n, \ldots, v_1$ is an elimination ordering for G of width at most k. Let $i \in \{1, \ldots, n\}$ and let H_i be the result of reducing G by $\{v_{i+1}, \ldots, v_n\}$. For every out-neighbor x of v_i in H, there is a v_i, x-path in G using only vertices from $\{x, v_i, \ldots, v_n\}$ according to Lemma 3.1. Thus, $x \in S_i$. Hence, v_i has at most k out-neighbors in H_i, and the claim follows. $\square$

4 Algorithm for recognition of digraphs of Kelly-width 2

Theorem 2.2 gives an easy non-deterministic algorithm for recognition of digraphs of Kelly-width $k+1$: reduce by some vertex of out-degree at most k and halt with a positive answer if the result is an empty graph. A polynomial-time algorithm does not evolve from this, since it is not clear which of the possible vertices to choose. However, we show that for Kelly-width 2 any vertex of out-degree at most 1 can safely be chosen.

Algorithm 1 Kelly-width 2 recognition

Input: Digraph G
Output: 'Yes', if G has Kelly-width at most 2, and 'No' otherwise

```
while G has a vertex v of out-degree at most 1 do
  reduce by v
end while;
if G is the empty digraph then return 'Yes' else return 'No' end if
```

Theorem 4.1 *The Kelly-width 2 recognition algorithm is correct.*

Proof By Theorem 2.2 it is clear that the algorithm will never answer 'Yes' unless G has Kelly-width at most 2. We can thus assume that G has an elimination ordering establishing that it has Kelly-width at most 2. We show that any vertex v of out-degree at most 1 can be the first in such an elimination ordering. The correctness of the algorithm will follow by induction.

Assume v of out-degree at most 1 and let σ be an elimination ordering establishing Kelly-width at most 2 in which v is reduced the earliest possible. If v is first we are done, so assume for a contradiction that there is a vertex u reduced immediately before v. The vertex u has out-degree at most 1 when it is about to be reduced. By assumption the vertex v has out-degree at most 1 in G and since reducing vertices of out-degree at most 1 cannot increase the out-degree of any vertex, v has out-degree at most 1 also when u is about to be reduced. Let σ' be the elimination order we get from σ by simply swapping the order of u and v. Thus, reducing G in the order given by σ' will up to reduction of u obey the rule of always reducing by vertices of out-degree at most 1. By Lemma 3.1 the reduced graphs after reducing either by σ up to v or by σ' up to u are equal, and thus reducing G by σ' is also an elimination ordering establishing Kelly-width at most 2. This contradicts the choice of σ as v is reduced later in σ than in σ'. $\square$

The algorithm can be provided with an output certificate guaranteeing correctness of an implementation, as follows. Let $\sigma = v_1, v_2, \ldots, v_i$ be the reduction sequence found by the algorithm, with $i = n$ if and only if the input graph had Kelly-width at most 2. Let F be the set of edges added during vertex reduction. If the input graph has Kelly-width at most 2 then certainly by Theorem 2.2 the order σ and the edges F together with the input graph form a certificate of this. Also in the negative case this is a certificate, as the proof of Theorem 2.2 implies that reducing by vertices of out-degree at most 1 in any order is safe, and σ does just this adding edges F to result in a reduced graph with no vertices of out-degree at most 2.

Theorem 4.2 *The Kelly-width 2 recognition algorithm augmented with output σ and $H = G \cup F$ is a certifying algorithm that, given a digraph G, decides whether G has Kelly-width at most 2. The running time and the working space of the algorithm are linear in the size of the output graph.*

Proof We have already argued that the algorithm is correct and that the output can be used as a certificate. For the running time of the reduction algorithm note that, after every reduction step, the resulting graph is a subgraph of H. Reducing by a vertex takes time linear in the number of its neighbors, since the in-neighbors of the reduced vertex become in-neighbors of the single out-neighbor, if there is one. Let a be eliminated and let b be its only out-neighbor. The crucial point is to find the in-neighbors of a that are in-neighbors also of b. These are exactly the vertices whose out-degree is decreased by 1. We assume that the adjacency lists of the vertices are ordered with respect to some ordering. The intersection of the two in-neighborhoods is computed by just scanning the two lists. The reason that this is linear time even if the in-neighborhood list of b is larger than that of a is that the in-degree of a in H is not smaller than the in-degree of b. This gives linear running time and working space in the size of input and output graph. □

For deciding whether a graph has Kelly-width exactly 2, it suffices to run the algorithm only for non-acyclic digraphs. Acyclic digraphs are exactly the digraphs of Kelly-width 1. The algorithm has running time linear in the output certificate. We leave it as an open problem if the algorithm can be implemented to have running time linear in the input digraph. We see at least two reasons why this will not be trivial.

Firstly, consider the digraph G constructed from a bi-directional path $v_1, v_2, \ldots, v_n$ by adding arcs from every vertex into v_1. It is easy to check that $\sigma = v_1, v_2, \ldots, v_n$ is the one and only elimination ordering showing that G has Kelly-width 2. When reducing by vertex v_i in this order, we will add $n - i - 2$ new arcs. Thus, any elimination process showing that G has Kelly-width 2 adds $\Theta(n^2)$ arcs even though the input graph has only $\Theta(n)$ arcs.

Secondly, not all digraphs of Kelly-width 2 have an elimination ordering $\sigma = v_1, v_2, \ldots, v_n$ with no added arcs going 'forward' from a lower-numbered to a higher-numbered vertex. Consider the digraph G depicted in Figure 1. Elimination order d, b, f, a, c, e shows that G has Kelly-width 2, but note that a forward arc (f, a) is then added. Let us argue that every elimination order showing Kelly-width 2 will have such forward arcs added. G has three vertices with out-degree at most 1, namely a, b and d. Furthermore, vertices c and e have out-degree 5, which means that c and e are the last two vertices in every such order. We distinguish two cases. If a is the first vertex then the arc (f, e) is added which must be a forward arc as we have already argued that e must be reduced after f. If b or d is the first vertex then the arc (f, a) is added so a would need to be reduced before f but then after reducing by a (and the other of b or d) we would have a reduced graph with every vertex having at least two out-neighbors.

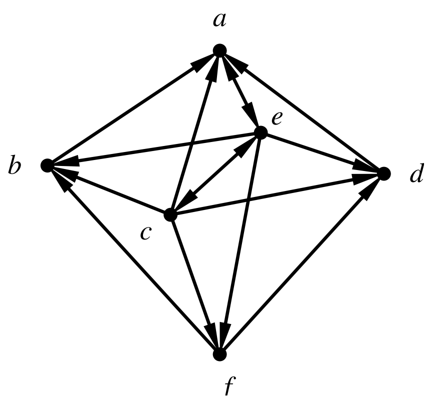

Figure 1: Elimination order d, b, f, a, c, e shows that this digraph has Kelly-width 2

5 Recognition of digraphs of Kelly-width k

As for recognition of graphs of Kelly-width k for any $k > 2$, we do not believe these ideas will generalize easily. The main reason for this is that when reducing by a vertex of out-degree larger than 1 the out-degree of remaining vertices can increase. Nevertheless, it is easy to see that for any k we can safely reduce by a vertex of in- and out-degree 1. As another simplification, we show in the following that it suffices to consider only strongly-connected components for determining the Kelly-width. For a digraph $G = (V, A)$, a *directed cut* (L, R) is a partition of the vertex set into non-empty sets L and R such that there is no arc (u, v) in G with $u \in R$ and $v \in L$.

Lemma 5.1 *Let $G = (V, A)$ be a digraph and let $k \geq 0$. Let (L, R) be a directed cut in G, and let G_L and G_R be the subgraphs of G induced by L and R, respectively. Then, the Kelly-width of G is at most $k + 1$ if and only if the Kelly-width of G_L and G_R is at most $k + 1$.*

Proof Since the Kelly-width does not increase by taking subgraphs, Kelly-width of G at most $k+1$ implies that the Kelly-width of G_L and G_R is at most $k+1$. For the other direction, let the Kelly-width of G_L and G_R be at most $k + 1$. Due to Theorem 2.2, there are elimination orderings σ_L and σ_R of width at most k for G_L and G_R, respectively. We show that the concatenation of both orderings, $\sigma_R \circ \sigma_L$, is an elimination ordering of width at most k for G. Since there is no path from vertices in R to vertices in L in G, all arcs that are added to G when reducing according to σ_R are arcs that are added to G_R when reducing according to σ_R. Thus, when reducing G according to σ_R, no vertex has out-degree more than k when it is chosen. Furthermore, due to Lemma 3.1, the result of reducing G by R is equal to G_L. Since σ_L is an elimination ordering for G_L of width at most k, we conclude that $\sigma_R \circ \sigma_L$ is an elimination ordering for G of width at most k, and thus G has Kelly-width at most $k + 1$. □

Corollary 5.2 *Let $G = (V, A)$ be a digraph. Then, the Kelly-width of G is equal to the maximum Kelly-width taken over the strongly-connected components of G.*

Proof If G is strongly-connected the claim obviously holds. Now, let G not be strongly-connected. Then, there is a strongly-connected component H of G such that no vertex of H has an out-neighbor outside H. Hence, $(V(G) \setminus V(H), V(H))$ is a directed cut in G. We apply Lemma 5.1 and obtain the result by induction. □

6 Final remarks

We gave an easy algorithm for recognition of digraphs of Kelly-width 2. Our algorithm for Kelly-width 2 can be considered a directed version of the undirected counterpart: graphs of treewidth 2 are the graphs reducible to the empty graph by iteratively reducing vertices of degree at most 2. Also graphs of treewidth 3 have a recognition algorithm based on vertex reduction, albeit with more complicated rules for choosing which vertex to reduce, as discovered by Arnborg et al. [1]. It is an interesting open problem if also recognition of graphs of Kelly-width 3 can be done by reduction rules. A more general open problem is to decide if Kelly-width is FPT.

Acknowledgements. We thank an anonymous referee for suggesting simplified proofs.

References

[1] S. Arnborg, A. Proskurowski, and D. Corneil. Forbidden minors characterization of partial 3-trees. *Discrete Mathematics* 80(1): 1-19 (1990)

[2] D. Berwanger, A. Dawar, P. Hunter, and St. Kreutzer. DAG-Width and Parity Games. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science, STACS 2006*, volume 3884 of *Lecture Notes in Computer Science*, pages 524–536. Springer-Verlag, 2006.

[3] H.L. Bodlaender. Treewidth: Characterizations, Applications, and Computations. In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2006*, volume 4271 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 2006.

[4] N.D. Dendris, L.M. Kirousis, and D.M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoretical Computer Science*, 172:233–254, 1997.

[5] G.A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1962.

[6] P. Hunter and St. Kreutzer. Digraph Measures: Kelly Decompositions, Games, and Orderings. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, pages 637–644, 2007.

[7] Th. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *Journal of Combinatorial Theory, Series B*, 82:138–154, 2001.

[8] D. Meister, J.A. Telle, and M. Vatshelle. Characterization and recognition of digraphs of bounded kelly-width. Technical Report 351, Institutt for Informatikk, Universitetet i Bergen, 2007.

[9] J. Obdržálek. DAG-width – Connectivity Measure for Directed Graphs. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006*, pages 814–821, 2006.

[10] M.A. Safari. D-Width: A More Natural Measure for Directed Tree Width. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science, MFCS*

2005, volume 3618 of *Lecture Notes in Computer Science*, pages 745–756. Springer-Verlag, 2005.

[11] M.Thorup. All Structured Programs have Small Tree-Width and Good Register Allocation. In *Information and Computation*, 142(2): 159-181 (1998)