On Digraph Width Measures in Parameterized Algorithmics

Robert Ganian[1], Petr Hliněný[1], Joachim Kneis[2], Alexander Langer[2],
Jan Obdržálek[1], and Peter Rossmanith[2]

[1] Faculty of Informatics, Masaryk University, Brno, Czech Republic
{xganian1,hlineny,obdrzalek}@fi.muni.cz
[2] Theoretical Computer Science, RWTH Aachen University, Germany
{kneis,langer,rossmani}@cs.rwth-aachen.de

Abstract. In contrast to undirected width measures (such as tree-width or clique-width), which have provided many important algorithmic applications, analogous measures for digraphs such as DAG-width or Kelly-width do not seem so successful. Several recent papers, e.g. those of Kreutzer–Ordyniak, Dankelmann–Gutin–Kim, or Lampis–Kaouri–Mitsou, have given some evidence for this. We support this direction by showing that many quite different problems remain hard even on graph classes that are restricted very beyond simply having small DAG-width. To this end, we introduce new measures K-width and DAG-depth. On the positive side, we also note that taking Kanté's directed generalization of rank-width as a parameter makes many problems fixed parameter tractable.

1 Introduction

The very successful concept of graph *tree-width* was introduced in the context of the Graph Minors project by Robertson and Seymour [RS86,RS91], and it turned out to be very useful for efficiently solving graph problems. Tree-width is a property of *undirected graphs.* In this paper we will be interested in *directed graphs* or *digraphs.*

Naturally, a width measure specifically tailored to digraphs with all the nice properties of tree-width would be tremendously useful. The properties of such a measure should include at least the following:

i) The width measure is small on many interesting instances.
ii) Many hard problems become easy if the width measure is bounded.

Obviously, there is a conflict between these goals, and consequently we can expect some trade-off. On the search for such a digraph measure, several suggestions were made, starting with directed tree-width [JRST01], and being complemented recently with several new approaches including DAG-width [Obd06,BDHK06], Kelly-width [HK08], entanglement [BG04], D-width [Saf05], directed path-width [Bar06] (defined by Reed, Seymour, and Thomas), and — although quite different — bi-rank-width [Kan08] (see Section 2).

Some positive results were encouraging: The Hamiltonian path problem can be solved in polynomial time (XP) if the directed tree width, the DAG-width, or the Kelly-width are bounded by a constant [JRST01]. More recently, it has been shown that parity games can be solved in polynomial time on digraphs of bounded DAG-width [BDHK06] and Kelly-width [HK08].

Are more results just waiting around the corner and do we just have to wait until we get more familiar with these digraph measures? It is the aim of this paper to answer this question, at least partially.

Unfortunately, as encouraging as the first positive results are, there is also the negative side. Hamiltonian path is W[2]-hard on digraphs of bounded DAG-width [LKM08], and some other natural problems even remain NP-hard on digraphs of low widths [KO08,DGK08,LKM08]. 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 graphs structure.

We introduce two digraph measures for this purpose: K-width and DAG-depth. While K-width (Section 2.3) restricts the number of different simple paths between pairs of vertices, DAG-depth (Definition 2.6) is the directed analog of tree-depth [NdM06]. K-width and DAG-depth are very restrictive digraph measures; at least as high as DAG-width, and often much higher.

The problems we consider in this paper (and formally define in Section 3) are Hamiltonian path (HAM), Disjoint paths (k-Path), Directed Dominating Set (DiDS), unit cost Directed Steiner Tree (DiSTP), Directed Feedback Vertex Set (DFVS), Kernel (Kernel), Maximum Directed Cut (MaxDiCut), Oriented Colouring (OCN), MSO$_1$ model checking (ϕ-MSO$_1$mc), solving Parity Games (Parity) and LTL-model checking (ϕ-LTLmc). See Table 1 in Section 3.

It turns out that most of the aforementioned problems are not only hard for DAG-width, but even for constant K-width and DAG-depth, or on DAGs. This can be seen as a strong indication that DAG-width or related measures are not yet the right parameters for dealing with standard digraph problems.

On the other hand, one width measure that fares much better in Table 1 is bi-rank-width (Definition 2.4), a width measure generalizing the rank-width of undirected graphs [Kan08]. 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 [HO08,Kan08].

2 Digraph Width Measures

The first wave of directed measures to appear shared the following features:

i) On bidirected orientations of graphs they coincided with the tree-width.
ii) These measures were strongly based on some variant of the directed *cops-and-robber game* on a digraph: There are k cops and a robber. Each cop can either occupy a vertex, or move around in a helicopter, and the robber occupies a vertex. The robber can, however, see the helicopter landing, and

can move at a great speed along a cop-free directed path to another vertex. The objective of the cops is to capture the robber by landing on the vertex currently occupied by him, the objective of the robber is to avoid capture.

iii) Point (ii) implied that DAGs and other graphs where vertices could be ordered in such a way that edges between them point mainly in one direction, and only a few point backwards, have a very low width.

iv) The last feature (iii) also made the algorithms to be XP, instead of FPT, because of the need to remember the partial results for all vertices with incoming edges from the outside, of which there could be $|V|$.

Directed tree-width. The first explicit directed measure was that of *directed tree-width (dtw)* [JRST01]. In the related cops-and-robber game the robber has to stay in the same cop-free strongly connected component, however the relationship between the number of cops needed and the directed tree-width is not strict. [JRST01] also contains XP algorithms for solving the Hamiltonian cycle, k-path, and related problems on graphs of bounded directed tree-width.

DAG-width. First defined in [Obd06] and, independently, in [BDHK06], *DAG-width (dagw)* was the next attempt to come up with a directed tree-width counterpart. This time the robber does not have to stay in the SCC, but the cop strategy has to be monotone, i.e., a cop cannot be placed on a previously vacated vertex. This game fully characterizes DAG-width. Note that monotone and non-monotone strategies are not equivalent [KO08].

Theorem 2.1 ([Obd06,BDHK06]). *For any graph G, there is a DAG-decomposition of G of width k if, and only if, the cop player has a monotone winning strategy in the k-cops-and-robber game on G.*

Kelly-width. Defined a year later, *Kelly-width (kellyw)* [HK08] aimed to solve an existing problem with DAG-decompositions: the number of nodes can be polynomially larger then the number of vertices in the original graph (the size depends on the width). The idea of Kelly-decompositions is based on the elimination ordering for tree-width, and therefore the size of the decomposition is linear in the size of the graph. The game characterizing Kelly-width is as for DAG-width, but with two important differences: 1) the cops cannot see the robber, and 2) the robber can move only when a cop is about to land on his vertex.

Cycle rank. This is perhaps the oldest definition of a digraph connectivity measure, given in 60's by Eggan and Büchi [Egg63].

Definition 2.2 (Cycle rank). The *cycle rank* $cr(G)$ of a digraph G is defined inductively as follows: For DAGs, $cr(G) = 1$. If G is strongly connected and $E(G) \neq \emptyset$, then $cr(G) = 1 + \min\{\, cr(G - v) : v \in V(G)\,\}$. Otherwise, $cr(G)$ is the maximum over the cycle rank of the strongly connected components of G.

Measure comparison. All the measures presented above are closely related to each other. The following theorem in a summary shows that if a problem is hard for graphs of bounded cycle rank, then it is hard for all the other measures.

Theorem 2.3. *Let G be a digraph. Then* (dpw *[Bar06] is the directed path-width):*

$$1/3(dtw(G)-1) \leq_{\text{[BDHK06]}} dagw(G) \leq dpw(G) \leq_{\text{[Gru08]}} cr(G)$$
$$1/6(dtw(G)+2) \leq_{\text{[HK08]}} kellyw(G) \leq dpw(G) \leq_{\text{[Gru08]}} cr(G)$$

Moreover, when DAG-width is bounded, so is Kelly-width [HO06].

2.1 Directed rank-width

The rank-width of undirected graphs was introduced by Oum and Seymour in relation to graph clique-width. While the definition of clique-width works "as is" also on digraphs, the following straightforward generalization of rank-width to digraphs (related to clique-width again) has been proposed by Kanté [Kan08].

Definition 2.4 (Bi-rank-width). Consider a digraph G, and vertex subsets $X \subseteq V(G)$ and $Y = V(G) \setminus X$. Let $\boldsymbol{A}_X^+$ denote the $X \times Y$ 0,1-matrix with the entries $a_{i,j} = 1$ $(i \in X,\ j \in Y)$ iff $(i,j) \in E(G)$, and let $\boldsymbol{A}_X^- = (\boldsymbol{A}_Y^+)^T$. The *bi-cutrank* function of G is defined as the sum of the ranks of these two matrices $brk_G(X) = rk(\boldsymbol{A}_X^+) + rk(\boldsymbol{A}_X^-)$ over the binary field $GF(2)$. The *bi-rank-width* $brwd(G)$ of G then equals the branch-width of this bi-cutrank function brk_G.

We remind the readers that the *branch-width* [RS91] of an arbitrary symmetric submodular function $\lambda\colon 2^E \to \mathbb{N}$ is defined as the minimum width over all branch-decompositions of λ over E, where a *branch-decomposition* is a pair T,τ satisfying the following: T is a tree of degree at most three, and τ is a bijection from E to the leaves of T. If f is an edge of T, then let $X_f \subseteq V(T)$ be the vertex set of one of the two connected components of $T - f$, and let the *width* of f be $\lambda(\tau^{-1}(X_f))$. The width of T,τ is the largest width over all edges of T.

Importantly, as proved by Kanté [Kan08], the rank-decomposition algorithm of [HO08] can also be used to find an optimal bi-rank-decomposition of a digraph.

Theorem 2.5 ([HO08] and [Kan08]). *Let $t \in \mathbb{N}$ be constant. There exists an algorithm that in time $O(n^3)$, for a given n-vertex graph (digraph) G, either outputs a rank-decomposition (bi-rank-decomposition, respectively) of G of width at most t, or certifies that the rank-width (bi-rank-width) is more than t.*

A rank-decomposition is, actually, not so suitable for designing dynamic programming algorithms. Yet, there is an efficient alternative characterization of a rank-decomposition via algebraic terms (or *parse trees*) over the bilinear graph product, which has been proposed by Courcelle and Kanté [CK07] and further extended towards algorithmic applications by [GH08] (see also an independent similar approach of [BXTV08]). As shown in [Kan08], an analogous "dynamic programming friendly" parse-tree view (of bi-rank-width) exists for digraphs, and we will apply this later, e.g. in Theorems 3.7 and 3.12.

2.2 DAG-depth

This part is inspired by the tree-depth notion of Nešetřil and Ossona de Mendez. [NdM06, Lemma 2.2] gives an inductive definition of the *tree-depth* $td(G)$ of undirected G as follows (compare to Def. 2.2). If G has one vertex, then $td(G) = 1$. If G is connected, then $td(G) = 1 + \min\{\, td(G - v) : v \in V(G) \,\}$. Otherwise, $td(G)$ equals the maximum over the tree-depth of the components of G.

We propose a new "directed" generalization of this definition. For a digraph G and any $v \in V(G)$, let G_v denote the subdigraph of G induced by the vertices reachable from v. The maximal elements of the poset $\{\, G_v : v \in V(G) \,\}$ in the graph-inclusion order are called *reachable fragments* of G. Notice that reachable fragments in the undirected case coincide with connected components.

Definition 2.6 (DAG-depth). The *DAG-depth* $ddp(G)$ of a digraph G is inductively defined: If $|V(G)| = 1$, then $ddp(G) = 1$. If G has a single reachable fragment, then $ddp(G) = 1 + \min\{\, ddp(G - v) : v \in V(G) \,\}$. Otherwise, $ddp(G)$ equals the maximum over the DAG-depth of the reachable fragments of G.

Comparing Definitions 2.2 and 2.6, one can see that DAG-depth equals cycle rank on bidirected orientations of graphs. Furthermore, the following useful game characterization of this new measure can be proved along Definition 2.6.

Theorem 2.7. *The DAG-depth of a digraph G is at most k if, and only if, the cop player has a "lift-free" winning strategy in the k-cops and robber game on G, i.e., a strategy that never moves a cop from a vertex once he has landed.*

Corollary 2.8 (cf. Theorem 2.1, Def. 2.2). *For any digraph G, the DAG-depth of G is greater than or equal to the DAG-width and the cycle rank of G.* □

Another claim tightly relates our new measure to directed paths in a digraph.

Proposition 2.9. *Consider a digraph G of DAG-depth t, and denote by ℓ the number of vertices of a longest directed path in G. Then $\lfloor \log_2 \ell \rfloor + 1 \le t \le \ell$.*

2.3 K-width

Moreover, applications in various "directed path" problems, see e.g. Section 3.1, inspired the following width measure: The *K-width* (a shortcut of "Kenny width") of a digraph G is the maximum number of distinct (not necessarily disjoint) simple s–t paths in G over all pairs of distinct vertices $s, t \in V(G)$.

Similarly to DAG-depth in Proposition 2.9, K-width can be arbitrarily large on DAGs. By giving a suitable search strategy for the cop player in a digraph G based on a DFS tree of G, we show that K-width is lower-bounded by DAG-width, but K-width is generally incomparable with cycle-rank which is unbounded on bidirected paths.

Theorem 2.10 (cf. Theorem 2.1). *For any digraph G, the K-width of G is greater or equal to the DAG-width of G minus one.*

Furthermore, an easy algorithm enumerating all paths leads to:

Proposition 2.11. *The K-width k of a given digraph G can be computed in time $k \cdot \mathrm{poly}(|V(G)|)$.*

3 Summary of Complexity Results

Table 1. Old and new (in boldface) complexity results on digraph measures (*-marked results assume a decomposition is given in advance; p-NPC is a shortcut for the complexity class para-NPC; and c and ϕ are fixed parameters of the respective problems).

Problem	K-width	DAG-depth	DAG-width	Cycle-rank	DAG	Bi-rank-width
HAM	**FPT**	**FPT**	XP[a] * W[2]-hard[c]	XP[a] *	P	XP[b] W[2]-hard[d]
c-Path	**FPT**	**FPT**	XP[a] *	XP[a] *	P[a]	**FPT**
k-Path	**p-NPC**	**p-NPC**	**NPC**	**NPC**	**NPC**	p-NPC[e]
DiDS	**p-NPC**	**p-NPC**	**NPC**	**NPC**	**NPC**	**FPT**
DiSTP	**p-NPC**	**p-NPC**	**NPC**	**NPC**	**NPC**	**FPT**
MaxDiCut	p-NPC[c]	p-NPC[c]	NPC[c]	NPC[c]	NPC[c]	**XP**
c-OCN	**p-NPC**	**p-NPC**	NPC[f]	NPC[f]	NPC[f]	**FPT**
DFVS	*open*	*open*	p-NPC[g]	p-NPC[g]	P	**FPT**
Kernel	p-NPC[h]	p-NPC[h]	p-NPC[g,h]	p-NPC[g,h]	P	**FPT**
ϕ-MSO$_1$mc	**p-NPH**	**p-NPH**	NPH	NPH	NPH	**FPT**[i]
ϕ-LTLmc	**p-coNPH**	**p-coNPH**	**coNPH**	**coNPH**	**coNPC**	**p-coNPH**
Parity	XP[j]	XP[j]	XP[j] *	XP[j] *	P	XP[k]

References [a][JRST01] [b][GH09] [c][LKM08] [d][FGLS09] [e][GW06] [f][CD06] [g][KO08] [h][vL76] [i][CMR00] [j][BDHK06] [k][Obd07] . Refer to the respective following sections for details and the new results.

3.1 Hamiltonian Path (HAM) and Disjoint Paths (k-Path)

The classical NP-hard *Hamiltonian Path* (HAM) problem [GJ79] is to find a directed path that visits each vertex of a digraph exactly once. A natural generalization of HAM is the *Longest Path* problem (Longest Path), where one is asked to find the longest simple path in a given digraph.

It is easy to see that HAM can be solved on DAGs in polynomial time. When using the parameter DAG-width, HAM belongs to XP [JRST01], but was also proven to be W[2]-hard [LKM08]. We prove our new FPT results for the parameters K-width and DAG-depth on more general Longest Path. Using a simple enumeration of all distinct paths in the case of bounded K-width, or applying Proposition 2.9 and any FPT-algorithm for Longest Path in the standard parameterization (e.g. [CKL$^+$09]) when DAG-depth is bounded, we get:

Theorem 3.1. *There is a fixed parameter tractable algorithm solving the* Longest Path *problem on a digraph G*

a) in time $O\big(t \cdot |V(G)| \cdot |E(G)|\big)$ if G is of K-width at most t;

b) in time $O\big(4^{2^t+O(t^3)} \cdot |V(G)| \cdot |E(G)|\big)$ if G is of DAG-depth at most t.

Another well-known problem is *Disjoint Paths* (k-Path); given a digraph and k pairs of nodes (s_i, t_i), $1 \le i \le k$, the task is to find pairwise disjoint directed paths from each s_i to the respective t_i. This problem is NP-complete [FHW80] even when k is bounded by any constant $c \ge 2$ (c-Path). Moreover, a "mixed" generalization of c-Path remains NP-complete [BJK09] even on DAGs, and k-Path is NP-complete [GW06] even on digraphs of bounded bi-rank-width.

If the digraph of an instance of k-Path has K-width ≤ 2, then it can be expressed as a 2-SAT formula, and if DAG-depth is ≤ 2, then it is equivalent to an SDR instance (system of distinct representatives). If, however, we slightly relax the restrictions as follows, the problem becomes NP-complete again.

Theorem 3.2. *The k-Path problem (with k as part of input)*

a) can be solved in polynomial time on graphs of K-width or DAG-depth 2*;*

b) is NP-complete on DAGs of K-width 3 *and DAG-depth* 4*.*

Finally, since one can express an instance of c-Path for any fixed c in MSO$_1$ logic (Section 3.6), it follows from Theorem 3.12 that this problem is fixed parameter tractable on digraphs of bi-rank-width t with parameters c and t. The c-Path problem however also becomes easier for the other new measures:

Theorem 3.3. *There is a fixed parameter tractable algorithm (for constant c) solving the c-Path problem on a digraph G*

a) in time $O(t^c \cdot |E(G)|)$ if G is of K-width at most t;

b) in time $O\big((2c)^{ct4^t} \cdot |E(G)|^2\big)$ if G is of DAG-depth at most t.

3.2 Directed Dominating Set (DiDS) and Steiner Tree (DiSTP)

The well-known NP-hard *Dominating Set* (DS) and *Steiner Tree* (STP) problems both allow for natural directed counterparts. We consider them in their unweighted variants for simplicity. The *Directed Dominating Set* problem (DiDS) asks for a minimum cardinality vertex set X in a digraph G such that every vertex of G not in X is an outneighbour of X. The *Directed Steiner Tree* problem (DiSTP) [HRW92], given a digraph G and $T \subseteq V(G)$, $r \in V(G)$, asks for a minimum size tree in G spanning $\{r\} \cup T$ with all arcs oriented away from r.

While it is folklore that both of these problems are NP-hard in general, we show (with a simple reduction from Vertex Cover) that the same holds even on very restricted graph classes.

Theorem 3.4. DiDS *and* DiSTP *problems are NP-complete on a digraph G even if G is restricted to be a DAG of K-width* 2 *and DAG-depth* 3*.*

Applying the MSO$_1$ optimization framework described in Section 3.6 we get:

Proposition 3.5 (Theorem 3.12). *The (unit cost)* DiDS *and* DiSTP *problems are fixed parameter tractable when parameterized by bi-rank-width.*

3.3 Maximum directed cut (MaxDiCut)

Maximum directed cut (MaxDiCut) is an extensively studied problem on digraphs. Given a digraph G, the goal is to partition the vertex set $V(G)$ into V_0 and V_1 such that the cardinality of $\{\, (u,v) \in E(G) : u \in V_0, v \in V_1 \,\}$ is maximized. This problem is often stated with edge weights, but we consider only the unweighted (cardinality MaxDiCut) variant in our paper.

It is well known that the MaxDiCut optimization problem is NP-hard, and it has been shown that MaxDiCut stays NP-hard even on DAGs [LKM08]. A closer, yet quite nontrivial look, at the reduction reveals the resulting graph to have also bounded DAG-depth and K-width.

Theorem 3.6 ([LKM08]). *The* MaxDiCut *problem is NP-hard on a digraph G even if G is restricted to be a DAG of K-width* 4608 *and DAG-depth* 11*.*

The only new efficiently solvable case among our measures is the following:

Theorem 3.7. *The unweighted* MaxDiCut *problem on a digraph G of bi-rank-width t is polynomially solvable for every fixed t (i.e. it belongs to the class XP).*

3.4 Oriented Colouring (OCN)

A natural directed generalization of the ordinary graph colouring problem can be obtained as follows: The chromatic number $\chi(G)$ of a graph G equals the minimum c such that G has a homomorphism into the complete graph K_c. The *Oriented Chromatic Number* (OCN) $\chi_o(G)$ of a digraph G is defined as the minimum c such that G has a homomorphism into some(!) orientation of K_c.

In other words, $\chi_o(G)$ equals minimum c such that the vertex set of G can be partitioned into c independent sets such that, between each pair of the sets, all arcs have the same direction. For instance, $\chi_o = 5$ for the directed 5-cycle.

It has been shown [KM04] that checking $\chi_o(G) \leq 3$ is easy, but determining whether $\chi_o(G) \leq 4$ is already NP-complete. Subsequently, [CD06] have shown that the problem $\chi_o(G) \leq 4$ remains NP-complete even on acyclic digraphs. Using a simpler and more powerful reduction than [CD06], we prove:

Theorem 3.8. *The problem (*4-OCN*) to decide whether a digraph G satisfies $\chi_o(G) \leq 4$ is NP-complete even if G is a DAG of K-width* 3 *and DAG-depth* 5*.*

On the other hand, it follows from the general framework of Theorem 3.12:

Proposition 3.9. *The problem (c-*OCN*) to decide $\chi_o(G) \leq c$ on an input digraph G of bi-rank-width t is fixed parameter tractable with parameters c and t.*

3.5 Directed Feedback Vertex Set (DFVS) and Kernel (Kernel)

The *directed feedback vertex set* (DFVS) problem is to find a minimum cardinality set S of vertices of a digraph G whose removal leaves $G \setminus S$ acyclic. This problem is trivial for acyclic digraphs, and it is FPT with the parameter $k = |S|$. We hence consider only the optimization variant of DFVS with unbounded k.

Kreutzer and Ordyniak [KO08] gave a reduction showing NP-hardness of the DFVS optimization problem on digraphs of DAG-width 4. A closer look at this reduction reveals that all the produced graphs are moreover of cycle rank 4, but they have unbounded K-width and DAG-depth.

The *kernel* of a digraph G is defined as an independent set $S \subseteq V(G)$ such that for every $x \in V(G) \setminus S$ there is an arc from x into S. Notice that a kernel may not always exist. However, on acyclic digraphs, a kernel can be easily found. Having a closer look at the NP-completeness reduction of van Leeuwen [vL76], one discovers the following claim (cf. also [KO08]).

Theorem 3.10 (van Leeuwen [vL76]). *It is NP-complete to decide whether a digraph G has a kernel, even if G is restricted to have (all at once) DAG-width and K-width* 2*, cycle rank also* 2*, and DAG-depth* 4*.*

Finally, by Example 3.11 and Theorem 3.12, both the Kernel and DFVS problems are fixed parameter tractable on digraphs of bounded bi-rank-width.

3.6 MSO$_1$ Model Checking (ϕ-MSO$_1$MC)

Monadic second order (MSO) logic is a language often used for description of combinatorial algorithmic problems. When applied to a one-sorted relational graph structure (i.e. to a set V with a symmetric relation $edge(u,v)$), this language is abbreviated as MSO$_1$. We use the same abbreviation MSO$_1$ also for digraphs with a relation $arc(u,v)$.

Example 3.11. The following properties are expressible in MSO$_1$ on digraphs

- a directed dominating set X as $\forall z\big(z \in X \vee \exists x \in X\ arc(x,z)\big)$,
- the existence of a kernel S as $\exists S\,\forall x\big[x \notin S \leftrightarrow \big(\exists y \in S\ arc(x,y)\big)\big]$, or
- a feedback vertex set Z as $\forall X\big[X \cap Z = \emptyset \rightarrow \big(\exists x \in X\, \forall y \in X\ \neg arc(x,y)\big)\big]$.

On the other hand, MSO$_1$ cannot express Hamiltonian cycle, for instance.

The MSO$_1$ *model checking problem* (ϕ-MSO$_1$MC), where ϕ is a fixed formula, is FPT on (undirected) graphs of bounded clique-width or rank-width [CMR00,CK07]. Not surprisingly, this extends to digraphs parameterized by bi-rank-width. More generally, the *LinEMSO$_1$ optimization framework* includes all problems which can be expressed as maximization of a linear evaluational term over all tuples of sets $X_1,\dots,X_j$ satisfying $\psi(X_1,\dots,X_j)$ where ψ is an MSO$_1$ formula — see [CMR00] for details. Analogously to [CMR00] (or [GH08]) we get:

Theorem 3.12 (cf. [CMR00], and [Kan08,GH08]).
Every ψ-LinEMSO$_1$ optimization problem is fixed parameter tractable when restricted to digraphs of bi-rank-width t, with parameters t and ψ.

Theorem 3.12 particularly implies that the problems listed in Example 3.11 (and many others) are FPT on digraphs of bi-rank-width t. No analogous results, however, seem possible for our other directed width measures since one can interpret ϕ-MSO$_1$MC of arbitrary undirected graphs via subdividing each edge and giving the two new edges opposite orientations, leading to:

Proposition 3.13. *The ϕ-MSO$_1$MC problem is NP-hard even when restricted to DAGs that are of K-width* 1 *and DAG-depth* 2*.*

3.7 LTL Model Checking (ϕ-LTLMC) and Parity Games (PARITY)

Another useful language that allows to express properties of digraphs is *Linear Temporal Logic* (LTL) — see, e.g., [BK08]. LTL model checking remains hard for a fixed formula ϕ and all of the directed width measures we considered here, including bi-rank-width (as opposed to MSO_1 model checking).

Theorem 3.14. *The ϕ-LTLMC problem is coNP-hard even when the input digraph is restricted to have K-width* 1*, DAG-depth* 4*, and bi-rank-width* 2*.*

Theorem 3.15. *The ϕ-LTLMC problem is coNP-complete on DAGs.*

Parity games— see e.g. [GTW02] for a reference, play an important role in the field of model-checking and formal verification. There are many reasons for this. First, solving parity games is equivalent to model-checking the modal μ-calculus, an important modal logic subsuming many other logics (e.g. CTL). Moreover, the modal μ-calculus is a bisimulation invariant fragment of MSO_1.

Second, the exact complexity of solving a parity game is a long-standing open problem. It is known to be in NP $\cap$ co-NP, and widely believed to be in P. It is trivially in P for acyclic digraphs. Moreover, it was shown that solving a parity game is in XP for digraphs of bounded tree-width [Obd03], bounded DAG-width [BDHK06] (hence also on bounded K-width, DAG-depth, and cycle rank) and bounded Kelly-width [HK08], and of bounded clique-width [Obd07] (implying the same for bi-rank-width).

4 Conclusion

Table 1, and the related results in this paper, have left several interesting open problems and questions. Just to specifically mention a few:

1) We suggest there exist FPT algorithms solving the DFVS problem for bounded K-width or DAG-depth (two of the open table entries).
2) For some entries in the table, we neither expect an FPT algorithm, nor have an NP-hardness estimate. E.g., MAXDICUT or k-PATH for bi-rank-width, or c-PATH for cycle rank. Can we then, at least, show a W-hardness result?
3) While we have given FPT and XP, respectively, algoritms solving the unit-cost variants of DISTP and MAXDICUT, these problems are usually considered in their weighted variants and then we expect their complexity to be higher. We, however, have no further results in this direction.
4) Some suggest that the DFVS number (see in Section 3.5) perhaps can be a good directed width measure. However, since majority of our sample problems in Table 1 remain hard even on DAGs, there is not much room left for applications of the DFVS parameter. Interestingly though, KERNEL becomes FPT when parametrized by DFVS.

Theorem 4.1. *If a digraph G is given with a directed feedback vertex set of size k, then the* KERNEL *problem can be solved in time $O(2^k \cdot |V(G)|^2)$.*

Finally, we try to formulate the overall impression coming from Table 1: Robber-and-cops based width measures do not seem to be very useful for parameterized algorithms on digraphs. One reason might be that cops "give" good graph separators in the undirected case, but that does not work any more for digraphs. Considering the DFVS number as a width parameter does not seem to help either. We perhaps need something new to move on. At this moment, bi-rank-width seems like a good alternative.

Acknowledgments. This work has been supported by a Czech–German bilateral grant of GAČR and DFG (201/09/J021 and RO 927/9). Moreover, P. Hliněný has been supported by the Czech research grant GAČR 201/08/0308.

References

[Bar06] J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

[BDHK06] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-width and parity games. In *STACS'06*, volume 3884 of *LNCS*, pages 524–536. Springer, 2006.

[BG04] D. Berwanger and E. Grädel. Entanglement – a measure for the complexity of directed graphs with applications to logic and games. In *LPAR 2004*, volume 3452 of *LNCS*, pages 209–223. Springer, 2004.

[BJK09] J. Bang-Jensen and M. Kriesell. Disjoint directed and undirected paths and cycles in digraphs. Technical Report PP-2009-03, University of South Denmark, 2009.

[BK08] C. Baier and J.-P. Katoen. *Principles of Model Checking.* The MIT Press, Cambridge, Massachusetts, 2008.

[BXTV08] B.-M. Bui-Xuan, J. Telle, and M. Vatshelle. H-join and algorithms on graphs of bounded rank-width. submitted, November 2008.

[CD06] J.-F. Culus and M. Demange. Oriented coloring: Complexity and approximation. In *SOFSEM'06*, volume 3831 of *LNCS*, pages 226–236. Springer, 2006.

[CK07] B. Courcelle and M. Kanté. Graph operations characterizing rank-width and balanced graph expressions. In *WG'07*, volume 4769 of *LNCS*, pages 66–75. Springer, 2007.

[CKL$^+$09] J. Chen, J. Kneis, S. Lu, D. Mölle, S. Richter, P. Rossmanith, S. Sze, and F. Zhang. Randomized divide-and-conquer: Improved path, matching, and packing algorithms. *SIAM Journal on Computing*, 38(6):2526–2547, 2009.

[CMR00] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[DGK08] P. Dankelmann, G. Gutin, and E. Kim. On complexity of minimun leaf out-branching. arXiv:0808.0980v1, August 2008.

[Egg63] L. Eggan. Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10(4):385–397, 1963.

[FGLS09] F. Fomin, P. Golovach, D. Lokshtanov, and S. Saurab. Clique-width: On the price of generality. In *SODA'09*, pages 825–834. SIAM, 2009.

[FHW80] S. Fortune, J. E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.

[GH08] R. Ganian and P. Hliněný. Automata approach to graphs of bounded rank-width. In *IWOCA'08*, pages 4–15, 2008.

[GH09] R. Ganian and P. Hliněný. Better polynomial algorithms on graphs of bounded rank-width. In *IWOCA'09*, LNCS. Springer, 2009. to appear.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.

[Gru08] H. Gruber. Digraph complexity measures and applications in formal language theory. In *MEMICS'08*, pages 60–67, 2008.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002.

[GW06] F. Gurski and E. Wanke. Vertex disjoint paths on clique-width bounded graphs. *Theor. Comput. Sci.*, 359(1-3):188–199, 2006.

[HK08] P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008.

[HO06] P. Hliněný and J. Obdržálek. Escape-width: Measuring "width" of digraphs. Presented at Sixth Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications, 2006.

[HO08] P. Hliněný and S. Oum. Finding branch-decomposition and rank-decomposition. *SIAM J. Comput.*, 38:1012–1032, 2008.

[HRW92] F. Hwang, D. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics. Noth-Holland, 1992.

[JRST01] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

[Kan08] M. Kanté. The rank-width of directed graphs. arXiv:0709.1433v3, March 2008.

[KM04] W. Klostermeyer and G. MacGillivray. Homomorphisms and oriented colorings of equivalence classes of oriented graphs. *Discrete Mathematics*, 274:161–172, 2004.

[KO08] S. Kreutzer and S. Ordyniak. Digraph decompositions and monotonicity in digraph searching. In *WG'08*, volume 5344 of *LNCS*, pages 336–347. Springer, 2008.

[LKM08] M. Lampis, G. Kaouri, and V. Mitsou. On the algorithmic effectiveness of digraph decompositions and complexity measures. In *ISAAC*, volume 5369 of *LNCS*, pages 220–231. Springer, 2008.

[NdM06] J. Nešetřil and P. Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European J. Combin.*, 27(6):1024–1041, 2006.

[Obd03] J. Obdržálek. Fast mu-calculus model checking when tree-width is bounded. In *CAV 2003*, volume 2725 of *LNCS*, pages 80–92. Springer, 2003.

[Obd06] J. Obdržálek. DAG-width – connectivity measure for directed graphs. In *SODA'06*, pages 814–821. ACM-SIAM, 2006.

[Obd07] Jan Obdržálek. Clique-width and parity games. In *CSL'07*, volume 4646 of *LNCS*, pages 54–68. Springer, 2007.

[RS86] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, September 1986.

[RS91] N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory B*, 52(2):153–190, 1991.

[Saf05] M. Safari. D-width: A more natural measure for directed tree-width. In *MFCS'05*, volume 3618 of *LNCS*, pages 745–756. Springer, 2005.

[vL76] J. van Leeuwen. Having a Grundy-numbering is NP-complete. Technical Report 207, The Pennsylvania State University, September 1976.