Enumerating k-arc-connected orientations

Sarah Blind* Kolja Knauer†‡ Petru Valicov†§

August 6, 2019

Abstract

We give simple algorithms to enumerate the α-orientations of a graph G in delay $O(m^2)$ and to enumerate the outdegree sequences attained by k-arc-connected orientations of G in delay $O(knm^2)$. Combining both yields an algorithm that enumerates all k-arc-connected orientations of G in delay $O(knm^2)$ and amortized time $O(m^2)$. The latter improves over another approach using submodular flows and moreover is much simpler, since it is basically a combination of BFS searches.

1 Introduction

Given an undirected (not necessarily simple) graph G, we consider the problem of enumerating its orientations with given properties, i.e., outputting each orientation exactly once. Since generally, the set to be enumerated is very large, one wants to control the *delay*, i.e., the maximum time between two consecutive outputs (including the time before the first and after the last output) in terms of the input G or at least the average over these, called the *amortized time*. In the present paper we use the term *enumeration* instead of the sometimes used terms *generation* or *listing*.

The set of all orientations of $G = (V, E)$ can be identified with the set of vectors $\{0, 1\}^m$ where $m = |E|$. Thus, exploring it using a gray code, enumerating all possible orientations of an undirected graph can be done with constant delay [11]. It gets more interesting when enumerating *acyclic* orientations, i.e. those that have no directed cycles. In [37] an algorithm for enumerating all acyclic orientations with delay $O(n^3)$ but linear amortized time $O(n)$ was given, where $n = |V|$. Later, the delay was reduced to $O(mn)$ with an increase in amortized time to $O(m + n)$, [3]. Another improvement was obtained in [7], where an algorithm of delay $O(m)$ and the resulting amortized time $O(m)$, is given. Many more types of orientations have been studied with respect to their enumeration complexity, see [6] for an overview. A concept dual to acyclic orientations are *strongly connected* orientations i.e., those that for any two vertices $u, v \in V$ there is a directed path from u to v. In [8] an enumeration algorithm with delay $O(m)$ is given. The main objective of the present paper is to give a simple algorithm for the enumeration of the *k-arc-connected orientations* of G, i.e., those where at least k arcs have to be removed in order to destroy the strong connectivity. In the following, when there is no ambiguity, we will abbreviate k-arc-connected by saying *k-connected*. The concept of k-connectivity is a classic in the theory of directed graphs, but to our knowledge their enumeration has not been studied explicitly for $k \geq 2$. A theorem of Nash-Williams [31] asserts that G admits a k-connected orientation if and only if G is $2k$-edge connected. An independent proof of Lovász [29] of this result yields an easy algorithm to produce a k-connected orientation of G (if there is one) with runtime $O(n^6)$. More involved techniques have been developed to improve this runtime, see [19,28]. Important contributions to the theory of k-connected orientations come from Frank. In particular, he showed that any two k-connected orientations of G can be transformed into each other by reversing directed paths and directed cycles [16]. Disassembling this result lies at the core of our algorithms. Another important

*Université de Lorraine, LGIPM, F-57000 Metz, France
Email: sarah.blind@univ-lorraine.fr
†Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
Email: {kolja.knauer,petru.valicov}@lis-lab.fr
‡Departament de Matemàtiques i Informàtica, Universitat de Barcelona (UB), Barcelona, Spain
§LIRMM, CNRS, Université de Montpellier, Montpellier, France.

contribution of Frank is the integration of k-connected orientations into the theory of submodular flows, see [15]. In particular, finding a submodular flow (and hence a k-connected orientation) can be done in polynomial time. There is a considerable amount of literature proposing different algorithmic solutions for this task on simple graphs, multigraphs, and mixed graphs, see e.g. [15, 17, 18, 25].

Our general approach to enumeration will always be *backtrack search.* It consists of exploring a tree whose vertices are partial solutions and the leaves are solutions. Going through such a graph by a depth-first search allows the enumeration of all the solutions. In order to obtain a polynomial delay, it is important that no branch of the tree is searched in vain, that is, an internal node of the tree must lead to at least one final solution. In this case the delay of the algorithm depends on the depth of the tree and the time spent at each node.

Using a min-cost k-connected orientation algorithm due to [25] and following ideas of [2] one can design an algorithm that decides in $O(n^3k^3 + kn^2m)$ if a given mixed graph $G = (V, E \cup A)$, where some edges are oriented and others are not, can be extended to an orientation that is k-connected. This yields an enumeration algorithm for k-connected orientations with delay $O(m(n^3k^3 + kn^2m))$ The idea is to simply start with $G = (V, E)$ as the root vertex of the tree and at each node create two children by picking an edge, fixing it to be oriented in one way or the other, and verifying if a k-connected extension exists. Since this tree has depth in $O(m)$ the previously stated runtime follows from [25]. Note that any other algorithm finding a k-connected extension of a partial orientation could be used in this approach (for example algorithms from [15, 17, 18]). See Section 3 for a detailed description.

Here, we propose an alternative approach to solve this task. The main advantages of our algorithm are its simplicity (while algorithms for submodular flows tend to be rater intricate) and that the two parts of it enumerate objects of independent interest. The idea basically comes from Franks above mentioned result [16] that reversals of directed cycles and directed paths are enough to enumerate all k-connected orientations. We split this into two algorithms, one reversing cycles, the other paths.

Concerning cycle reversals, we give an algorithm that enumerates all orientations of G with a prescribed outdegree sequence also known as *α-orientations.* Given $\alpha : V \to \mathbb{N}$ an orientation D of G is an α-orientation if $\delta^+_D(v) = \alpha(v)$ for all $v \in V$, where δ^+_D denotes the outdegree. These orientations are of current interest with respect to computational properties (see e.g. [1]) and model many combinatorial objects such as domino and lozenge tilings of a plane region [36, 39], spanning trees of a planar graph [20], perfect matchings (and d-factors) of a bipartite graph [12, 27, 34], Schnyder woods of a planar triangulation [4], Eulerian orientations of a graph [12], and contact representations of planar graphs with homothetic triangles, rectangles, and k-gons [13, 14, 23].

Our enumeration algorithm for α-orientations has a delay of $O(m^2)$, is based on BFS searches, and is explained in Section 4. Note that the set of α-orientations of a planar graph can be endowed with a natural distributive lattice structure [12] and therefore, in this case the enumeration can be done in linear amortized time [24]. It is a famous question (also open in this setting) whether the enumeration of the elements of a distributive lattice can be done in constant amortized time [35].

The second part of our approach enumerates all outdegree sequences that are attained by k-connected orientations of a given graph. This algorithm has a delay of $O(m^2kn)$ and is explained in Section 5. In Section 6 we combine these algorithms, leading to an enumeration algorithm for k-connected orientations of delay $O(m^2kn)$ and amortized time $O(m^2)$. We close the paper in Section 7.

2 Preliminaries

In this section we introduce some digraph basics. All graphs we consider in this paper are multigraphs and consequently their orientations also may have multiple parallel or anti-parallel arcs. We will also consider *mixed (multi)graphs.* These are of the form $G = (V, E \cup A)$, where E is a (multi)set of undirected edges and A is a (multi)set of directed arcs. Analogously to undirected graphs, an orientation of a mixed graph consists in fixing a direction for each of its undirected edges.

For a vertex $u \in V$ of $D = (V, A)$ or more generally a subset $X \subseteq V$ we will denote its *outdegree* $\delta^+_D(X) = |\{a = (u, v) \in A \mid u \in X \not\ni v\}|$. The digraph obtained from $D = (V, A)$ by reversing a set of arcs $B \subseteq A$ is denoted by D^B. If $A = B$ we might write D^- instead of D^B. The reversed arc set of B is denoted $B^- = \{(u, v) \mid (v, u) \in B\}$. In particular, the inverse of an arc a is denoted a^-.

With respect to arc-connectivity in digraphs there is a generalization of Menger's Theorem [30], that we will make frequent use of. Denoting by $\lambda(u,v)$ the maximum number of arc-disjoint directed paths from u to v we can formulate it as:

Theorem 1 (Local Menger Theorem for digraphs)**.** *Let $D = (V, A)$ be a digraph and $u, v \in V$. We have $\lambda(u,v) = \min\{\delta^+_D(X) \mid u \in X \not\ni v\}$.*

Note that by the definition of k-connectivity D is k-connected if and only $\delta^+_D(X) \geq k$ for every non-empty proper subset $X \subset V$. Thus, as a consequence of Theorem 1, D is k-connected if and only $\lambda(u,v) \geq k$ for all $u, v \in V$. The main result of the article will be an algorithm for the following:

k-ARC-CONNECTED-ORIENTATIONS
Input : A graph $G = (V, E)$
Output : The set of all k-arc-connected orientations of G

3 A first enumeration algorithm

We present a more detailed description of the first enumeration algorithm mentioned in the introduction. In [2] using a min-cost k-connected orientation due to [25] the following result is given:

Theorem 2 ([2, 25])**.** *Deciding whether a given mixed graph $G = (V, E \cup A)$ admits a k-connected orientation can be done in time $O(k^3n^3 + kn^2m)$.*

The idea is just unorienting the whole graph G and finding a min-cost k-connected orientation of it, where the cost on A^- is set to be very high. We can use a backtrack search algorithm (Algorithm 1) which checks at each step when orienting some new edge, if there exists a k-arc-connected orientation respecting the partial fixed orientation.

Algorithm 1: enumeration of k-connected orientations via submodular flows

Input: A graph $G = (V, E)$ and an integer $k \in \mathbb{N}$
Output: The k-connected orientations of G

begin
  Fix any linear ordering on E;
  `Enumerate`$(G = (V, E, \emptyset))$;
end

Function `Enumerate`$(G = (V, E, F))$:
  if $E \neq \emptyset$ **then**
    Take the smallest $e = \{u, v\}) \in E$;
    if $G' := (V, E \setminus \{e\}, A \cup \{(u, v)\})$ *admits a k-connected orientation* **then**
      `Enumerate`(G');
    if $G' := (V, E \setminus \{e\}, A \cup \{(v, u))\}$ *admits a k-connected orientation* **then**
      `Enumerate`(G');
  else
    Output G;
end

This algorithm takes as input an undirected graph $G = (V, E)$. Clearly, all k-connected orientations are produced and since each node built by this algorithm gives rise to disjoint branches it does not repeat the same solution twice. The depth of the binary execution tree is m and at each node we check the orientability of a mixed graph which is solvable in $O(k^3n^3 + kn^2m)$ by Theorem 2. We conclude

Proposition 3. *Let G be a graph and $k \in \mathbb{N}$. Algorithm 1 enumerates all k-connected orientations of G with delay $O(m(k^3n^3 + kn^2m))$.*

The downside of Algorithm 1 and the main motivation for the rest of the paper is that the proof of Theorem 2, is rather intricate and based on finding a submodular flow. In the following we will give two simple algorithms of independent interest that together give an alternative solution for the enumeration of k-connected orientations. Moreover, the resulting algorithm has a better delay and even better amortized runtime.

4 Orientations with prescribed outdegree sequence

Let $G = (V, E)$ be a graph and $\alpha : V \to \mathbb{N}$. We say that an orientation D of G is an *α-orientation* if $\delta^+_D \equiv \alpha$, i.e., $\delta^+_D(v) = \alpha(v)$ for all $v \in V$. We will denote by $\mathcal{O}_\alpha(G)$ the set of all α-orientationsof G. The following is the enumeration problem to be discussed in the present section.

α-ORIENTATIONS
Input : A graph $G = (V, E)$ and a function $\alpha : V \to \mathbb{N}$
Output : The elements of $\mathcal{O}_\alpha(G)$

Before going into this algorithm let us give a folklore results about α-orientations:

Lemma 4. *Let G be a graph and D and D' be two orientations of G. We have $\delta^+_D = \delta^+_{D'}$ if and only if orientation D' can be obtained from D by reversing a set of arc-disjoint directed cycles.*

Proof. Reversing the direction of a directed cycle does not change the outdegree function. Therefore, if D' is obtained from D by reversing a set of arc-disjoint directed cycles, then $\delta^+_D = \delta^+_{D'}$. Now, let D and D' be orientations such that $\delta^+_D = \delta^+_{D'}$ and let us show that D' can be obtained from D by reversing some arc-disjoint directed cycles. Consider the arcs of D whose direction differ in D'. Observe that in the directed subgraph formed by these arcs the indegree and the outdegree coincide at each node (otherwise we would have $\delta^+_D \neq \delta^+_{D'}$). In other words this directed subgraph is Eulerian which is an arc-disjoint union of directed cycles, so we are done. □

Our algorithm enumerating $\mathcal{O}_\alpha(G)$ takes as an input a graph $D = (V, A) \in \mathcal{O}_\alpha(G)$ and a set of *fixed* arcs $F \subseteq A$. Initially D is an arbitrary α-orientation and $F = \emptyset$. The algorithm recursively constructs all orientations of $\mathcal{O}_\alpha(G)$ such that at each recursive call all possible α-orientations contain the current F. At each recursive call, an arc $a \notin F$ will be fixed with the two possible cases to consider: On the one hand, the α-orientations with $a = (u, v)$: this branch is known to be non-empty since D is a possible extension, thus we recurse with D and $F \cup a^+$. On the other hand, the α-orientations with $a^- = (v, u)$ (i.e. reversed a): if there is a directed path P in $D \setminus F$ from v to u, then the directed cycle a, P is reversed and we recurse with the new α-orientation$D^{a^- \cup P}$ and $F \cup a^-$.

If all arcs are fixed, then we output the current orientation. We give a presentation of our algorithm in pseudo-code, where the set of fixed arcs is extended along an arbitrary linear ordering of the arcs:

Algorithm 2: Backtrack search for *α-orientations*

Input: A graph $G = (V, E)$ and $\alpha : V \to \mathbb{N}$
Output: All elements of $\mathcal{O}_\alpha(G)$

begin
  if *there exists* $D \in \mathcal{O}_\alpha(G)$ **then**
    Fix an arbitrary linear order on A;
    `EnOPODS`(D, $\emptyset$);
end

```
Function EnOPODS(D, F):
    if F ≠ A then
        Take the smallest a = (u, v) ∈ A \ F;
        EnOPODS(D, F ∪ {a});
        if D \ F has a directed path P from v to u then
            EnOPODS(D^{a∪P}, F ∪ {a^-});
    else
        Output D;
end
```

Algorithm 2 takes an undirected graph $G = (V, E)$ as input, but after finding one initial α-orientation D, it just consists of recursive calls of the function EnOPODS(D, F).

Lemma 5. *Let $D \in \mathcal{O}_\alpha(G)$ and $F \subseteq A$. The function* EnOPODS(D, F) *enumerates each α-orientation that coincide with D on F exactly once with a delay of $O(m^2)$.*

Proof. We start by proving by induction on $|A \setminus F|$ that each of the claimed α-orientations is enumerated exactly once. If $|A \setminus F| = 0$, then EnOPODS(D, F) = EnOPODS(D, A) = $\{D\}$ and we are done. Let now $|A \setminus F| > 0$ and $a = (u, v) \in A \setminus F$. By induction hypothesis EnOPODS $(D, F \cup \{a\})$ enumerates each α-orientation that coincides with D on $F \cup \{a\}$ exactly once and EnOPODS($D^{a\cup P}, F \cup \{a^-\}$) enumerates each α-orientation that coincides with $D^{a\cup P}$ on $F \cup \{a^-\}$ exactly once. Clearly, both sets are disjoint since they differ on the orientation of a, i.e., no repetitions are produced. Moreover, since $a \cup P$ is a directed cycle, the digraph $D^{a\cup P}$ also is an α-orientation of G by Lemma 4 that fixes F, since $P \cap F = \emptyset$.

Let us proof that if there is no directed path P from u to v in $D \setminus F$, then there exists no α-orientation fixing F and reversing a. By contraposition, suppose that D' is an α-orientation that coincides with D on F but differs on a. Then by Lemma 4, there is a set of arc disjoint directed cycles in D' whose union is $D' \setminus D$. Since both digraphs coincide on F, these cycles are disjoint from F. Since both digraphs differ on a, one of the directed cycles C contains a^- in D'. Thus, the path $P = (C \setminus \{a^-\})^-$ is a directed path in $D \setminus F$ from v to u.

Finally, for the analysis of complexity, note that in each recursion step the algorithm only needs to check the presence of a directed path, which can be done by a single BFS from the source vertex u towards the target v. In general the complexity of a BFS algorithm is $O(m + n)$, however in our case the BFS tree will be constructed only on the strongly connected component of D containing u and thus the complexity is in $O(m)$. Moreover, the depth of our recursion tree is bounded by m. Thus, the total delay is bounded by $O(m^2)$. $\square$

Observe that Algorithm 2 has to use a separate method for finding a first element $D \in \mathcal{O}_\alpha(G)$. It is well-known that this problem can be reduced to a flow-problem, see e.g. [12]. Therefore, this preprocessing step can be done in $O(mn)$ time, see [33]. With Lemma 5 we obtain:

Theorem 6. *Let G be a graph and $\alpha : V \to \mathbb{N}$. Algorithm 2 enumerates $O_\alpha(G)$ with delay $O(m^2)$.*

5 Outdegree sequences

In this section we will present an algorithm to enumerate the possible outdegree sequences among the k-connected orientations of a graph G.

An easy consequence of Lemma 4 is the following that can also be found in [16].

Lemma 7. *If $D, D' \in \mathcal{O}_\alpha(G)$, then D is k-connected if and only if D' is k-connected.*

Proof. Let D be a k-arc-connected orientation of some graph G such that $\delta^+_D = \alpha$ and thus $\forall X \subseteq V$ we have $\delta^+_D(X) \geq k$. Since $\delta^+_D = \delta^+_{D'}$, by Lemma 4 the orientation D' can be obtained from D by reversing a set of disjoint directed cycles. On the other hand, reversing the direction of a directed cycle in D does not change the outdegree of the subsets of vertices of G. Therefore, after reversing a set of directed cycles to obtain D', we have that $\forall X \subseteq V$, $\delta^+_{D'}(X) \geq k$. $\square$

Given $G = (V, E)$, Lemma 7 allows to define a function $\alpha : V \to \mathbb{N}$ to be *k-connected* if there is some k-connected $D \in \mathcal{O}_\alpha(G)$. In this case we will sometimes call α a k-connected outdegree sequence. Having in mind that we want to enumerate all k-connected orientations of G and already are able to enumerate $\mathcal{O}_\alpha(G)$ for any given α, we are left with enumerating the k-connected outdegree sequences.

k-CONNECTED OUTDEGREE SEQUENCES
Input : A graph $G = (V, E)$, $k \in \mathbb{N}$
Output : The k-connected functions $\alpha : V \to \mathbb{N}$.

Here in order to change the outdegree sequence of D we will reverse a directed path P_{uv} from u to v and thus only increase $\delta^+_D(u)$ by one and decrease $\delta^+_D(v)$ by one. More generally we have:

Observation 1. *Let D be an orientation of graph G and $X \subseteq V(G)$. If D' is obtained from D by reversing a path from a vertex u to a vertex v, then we have*

1. $\delta^+_{D'}(X) = \delta^+_D(X)$ *if* $u, v \in X$ *or* $u, v \notin X$
2. $\delta^+_{D'}(X) = \delta^+_D(X) + 1$ *if* $u \notin X$ *and* $v \in X$
3. $\delta^+_{D'}(X) = \delta^+_D(X) - 1$ *if* $u \in X$ *and* $v \notin X$

We have to check how reversing paths affects the number of arc-disjoint directed paths between pairs of vertices. The following will be useful:

Lemma 8. *Let P_{uv} be a directed path from a vertex u to a vertex v in some orientation D. Then for all vertices u', v', we have $\lambda_{D^{P_{uv}}}(u'v') \geq \min(\lambda_D(u, v) - 1, \lambda_D(u', v'))$.*

Proof. With Menger's theorem (Theorem 1) and Observation 1, Lemma 8 can be easily proved:

$$\begin{aligned}
\lambda_{D^{P_{uv}}}(u', v') &= \min\{\delta^+_{D^{P_{uv}}}(X) \mid X \subseteq V, u' \in X, v' \notin X\} && \text{(Theorem 1)}\\
&\geq \min\{\min\{\delta^+_D(X) - 1 \mid X \subseteq V, u', u \in X, v', v \notin X\}, && \text{(Observation 1)}\\
&\quad \min\{\delta^+_D(X) \mid X \subseteq V, u', u, v \in X \text{ and } v' \notin X \text{ or } u' \in X \text{ and } v', u, v \notin X\},\\
&\quad \min\{\delta^+_D(X) + 1 \mid X \subseteq V, u', v \in X, v', u \notin X\}\}\\
&\geq \min\{\lambda_D(u, v) - 1, \lambda_D(u', v')\} && \text{(Theorem 1)}
\end{aligned}$$

□

In a k-connected digraph D we call a directed path P *flippable* if D^P is k-connected. Lemma 8 implies that if some path from u to v is flippable, then all of them are. Therefore, we call the pair (u, v) *flippable* if there is a flippable path P_{uv} from u to v, i.e., $\lambda(u, v) > k$.

Lemma 9. *Let D be k-connected, it can be decided in time $O(km)$ if (u, v) is flippable.*

Proof. By Lemma 8 a simple algorithm consists in finding a directed path P_{uv} in D, reverse it and iterate in $D^{P_{uv}} \setminus P^-_{uv}$. We have $\lambda_D(u, v) > k$ if and only if this procedure can be applied $k + 1$ times. Each execution is a BFS, which yields the claimed runtime. □

The following is a slight refinement of a result of Frank [16]:

Lemma 10. *Let $G = (V, E)$ be a graph and D, D' be two k-connected orientations of G. For every vertex v such that $\delta^+_D(v) < \delta^+_{D'}(v)$, there exists a vertex u such that $\delta^+_D(u) > \delta^+_{D'}(u)$ and (u, v) is flippable in D.*

Proof. Let v be such that $\delta^+_D(v) < \delta^+_{D'}(v)$. We will prove that there is u such that $\delta^+_D(u) > \delta^+_{D'}(u)$ and $\delta^+_D(X) > k$ whenever $u \in X$ and $v \notin X$. By Menger's Theorem (Theorem 1) this will imply that (u, v) is flippable. Notice that because we have v such that $\delta^+_D(v) < \delta^+_{D'}(v)$ and because $\sum_{w \in V} \delta^+_D(w) = |E| = \sum_{w \in V} \delta^+_{D'}(w)$, there exists a vertex u such that $\delta^+_D(u) > \delta^+_{D'}(u)$. By contradiction suppose that each such vertex u, for which $\delta^+_D(u) > \delta^+_{D'}(u)$, is contained in a set X of outdegree exactly k in D and

$v \notin X$. Among all these sets for all such u, we consider those that are inclusion maximal and collect them in $\mathcal{X}$.

First of all notice that all such sets are disjoint. Indeed consider X, X' with $\delta_D^+(X) = \delta_D^+(X') = k$ and $v \notin X \cup X'$ with $X \cap X' \neq \emptyset$. Since $X \cup X'$, $X \cap X'$ are both not empty, $X \cup X' \neq V$ and D is k-connected, we know that $\delta_D^+(X \cup X') \geq k$ and $\delta_D^+(X \cap X') \geq k$. Therefore, we have

$$k + k = \delta_D^+(X) + \delta_D^+(X') \geq \delta_D^+(X \cup X') + \delta_D^+(X \cap X') \geq k + k$$

This gives $\delta_D^+(X \cup X') = k$, i.e., X and X' were not maximal.

Now, let us count the number c of edges of G not contained in any subgraph of G induced by $X \in \mathcal{X}$ and denote by $Y = V \setminus \bigcup_{X \in \mathcal{X}} X$. Observe that since the elements of $\mathcal{X}$ are disjoint, in any orientation of G the number c is just the sum of outdegrees of $X \in \mathcal{X}$ plus the outdegrees of each vertex of Y. Thus, if we furthermore denote $t = |\mathcal{X}|$ we can do this counting with respect to D and D' and obtain the following contradiction

$$c = \sum_{X \in \mathcal{X}} \delta_D^+(X) + \sum_{w \in Y} \delta_D^+(w) = kt + \sum_{w \in Y} \delta_D^+(w) < \sum_{X \in \mathcal{X}} \delta_{D'}^+(X) + \sum_{w \in Y} \delta_{D'}^+(w) = c$$

More precisely, since D' is k-connected we have that $kt \leq \sum_{X \in \mathcal{X}} \delta_{D'}^+(X)$. Recall that we supposed that if w is such that $\delta_D^+(w) > \delta_{D'}^+(w)$, then $w \notin Y$. Therefore, we have $\delta_D^+(w) \leq \delta_{D'}^+(w)$ for all $w \in Y$. Adding this to the fact that $\delta_D^+(v) < \delta_{D'}^+(v)$ and $v \in Y$, we obtain the strict inequality claimed above. This concludes the proof. □

Note that a complete analogue of Lemma 10 holds for the case $\delta_D^+(v) > \delta_{D'}^+(v)$:

Lemma 11. *Let $G = (V, E)$ be a graph and D, D' be k-connected orientations of G. For every $v \in V$ with $\delta_D^+(v) > \delta_{D'}^+(v)$, there exists $u \in V$ such that $\delta_D^+(u) < \delta_{D'}^+(u)$ and (v, u) is flippable in D.*

Proof. Observe that an equivalent statement of the lemma is that for every vertex v such that $\delta_D^-(v) < \delta_{D'}^-(v)$, there exists a vertex u such that $\delta_D^-(u) > \delta_{D'}^-(u)$ and (v, u) is flippable in D. By fully reorienting all the arcs of D and D', we obtain graphs D^- and D'^- and get a further equivalent statement: for every vertex v such that $\delta_{D^-}^+(v) < \delta_{D'^-}^+(v)$, there exists a vertex u such that $\delta_{D^-}^+(u) > \delta_{D'^-}^+(u)$ and (u, v) is flippable in D^-. This is precisely the statement of Lemma 10, so we are done. □

Now we are ready de describe the general algorithm for k-CONNECTED OUTDEGREE SEQUENCES:

Algorithm 3: Enumeration of k-connected outdegree sequences

Input: A graph $G = (V, E)$, an integer k
Output: All k-connected outdegree sequences of G

begin
  if *there exists a k-connected orientation D for G* **then**
    Fix an arbitrary linear order on V;
    `EnODS`$(D, \emptyset)$;
end

Function `EnODS`(D, F):
  if $F \neq V$ **then**
    take the smallest $v \in V \setminus F$;
    `Reverse`$^-(D, F, v)$;
    `Reverse`$^+(D, F, v)$;
    `EnODS`$(D, F \cup \{v\})$;
  else
    Output δ_D^+;
end

Function Reverse$^-$(D, F, v):
  if *there exists* $u \in V \setminus F$ *such that* (v,u) *is flippable* **then**
    Take a directed path P_{vu} from v to u ;
    Reverse$^-$($D^{P_{vu}}$, F, v) ;
    EnODS($D^{P_{vu}}$, $F \cup \{v\}$);

Function Reverse$^+$(D, F, v):
  if *there exists* $u \in V \setminus F$ *such that* (u,v) *is flippable* **then**
    Take a directed path P_{uv} from u to v ;
    Reverse$^+$($D^{P_{uv}}, F, v$) ;
    EnODS($D^{P_{uv}}, F \cup \{v\}$);

Lemma 12. *Let D be a k-connected orientation of $G = (V, E)$ and $F \subseteq V$. The function* EnODS(D,F) *enumerates the k-connected outdegree sequences coinciding with D on F with delay is $O(knm^2)$.*

Proof. We show the first part of the lemma by induction on $|V \setminus F|$. If $|V \setminus F| = 0$, then EnODS(D,F) outputs δ^+_D and the claim holds. Consider now the case $|V \setminus F| > 0$ and let $v \in V \setminus F$ be the next vertex. By induction EnODS(D,$F \cup \{v\}$) enumerates every k-connected outdegree sequence coinciding with D on $F \cup \{v\}$ exactly once. We have to show that Reverse$^+$(D,F,v) (resp. Reverse$^-$(D,F,v)) enumerates all k-connected outdegree sequences coinciding with D on F and having outdegree of v larger (resp. smaller) than $\delta^+_D(v)$. Also, we have to show that each of these outdegree sequences will be enumerated exactly once. Note that this implies that globally each solution is produced exactly once.

Let us prove this for Reverse$^+$(D,F,v). So let D' be a k-connected orientation of G such that $\delta^+_D(F) \equiv \delta^+_{D'}(F)$ and $\delta^+_D(v) < \delta^+_{D'}(v)$. By Lemma 10 there exists a vertex $u \in V \setminus F$ such that (u, v) is flippable, i.e., for any path P_{uv} the orientation $D^{P_{uv}}$ is k-connected, its outdegree sequence coincides with D on F and $\delta^+_D(v) + 1 = \delta^+_{D^{P_{uv}}}(v)$.

We proceed by induction on $\delta^+_{D'}(v) - \delta^+_D(v)$ to show that $\delta^+(D')$ is enumerated exactly once. So for the base case $\delta^+_{D'}(v) - \delta^+_D(v) = 1$ we have $\delta^+_{D'}(v) = \delta^+_{D^{P_{uv}}}(v)$ and by induction $\delta^+_{D'}$ will be enumerated exactly once by the next call of EnODS($D^{P_{uv}}$,$F \cup \{v\}$) and not at all by Reverse$^+$($D^{P_{uv}}$, F, v) since the latter outputs degree sequences with $\alpha(v) > \delta^+_{D^{P_{uv}}}(v)$. Suppose now that $\delta^+_{D'}(v) - \delta^+_D(v) > 1$. We have $\delta^+_{D'}(v) - \delta^+_{D^{P_{uv}}}(v) < \delta^+_{D'}(v) - \delta^+_D(v)$, so by induction hypothesis Reverse$^+$($D^{P_{uv}}$,F,v) enumerates the outdegree sequence of $\delta^+_{D'}(v)$ exactly once.

The analogue proof works for Reverse$^-$ using Lemma 11.

For the analysis of complexity note that in each call of Reverse$^+$ or Reverse$^-$ for at most n times it has to be checked if a pair (u, v) is flippable. The latter can be done in time $O(km)$ by Lemma 9, finding a directed path from u to v is done in $O(m)$. So a call costs $O(knm)$. Finally, the depth of the recursion tree is in $O(m)$. To see this compare the $\delta^+_{D'}$ of a leaf orientation with the δ^+_D of orientation D at the root. Between any two calls of EnODS, there will be a sequence of at most $\deg(v)$ calls of Reverse$^+$ or Reverse$^-$. This way δ^+_D will be approached to $\delta^+_{D'}$ coordinate by coordinate, where previous coordinates are not affected by modifications on latter coordinates. Thus, there are at most $\sum_{v \in V} \deg(v) = O(m)$ calls and we get an overall delay of $O(knm^2)$. ☐

Note that, Algorithm 4 has to use a separate method for finding a first k-connected orientation D of G. This preprocessing step can be done in $O(k^3n^3 + kn^2m)$ [25]. On the other hand, recall that in a k-connected orientation we have $kn \leq m$. Therefore, together with Lemma 12 we obtain:

Theorem 13. *Let G be a graph and $k \in \mathbb{N}$. Algorithm 4 enumerates all k-connected outdegree sequences of G in delay $O(knm^2)$.*

6 k-connected orientations

Putting the above together we obtain an algorithm to enumerate k-connected orientations.

Algorithm 4: Simple enumeration of k-connected orientations

Input: A graph $G = (V, E)$, an integer k
Output: All k-connected orientations of G

begin
  if *there exists a k-connected orientation D for G* **then**
    Fix an arbitrary linear order on V;
    `EnODS'`$(D, \emptyset)$;
end

Function `EnODS'`(D, F)**:**
  if $F \neq V$ **then**
    take the smallest $v \in V \setminus F$;
    `Reverse`$^-(D, F, v)$;
    `Reverse`$^+(D, F, v)$;
    `EnODS'`$(D, F \cup \{v\})$;
  else
    `EnOPODS`$(D, \emptyset)$;
end

We need the following easy result for analyzing the amortized complexity .

Lemma 14. *Let $G = (V, E)$ be a graph and α k-connected, then $|\mathcal{O}_\alpha(G)| \geq (k-1)n + 1$.*

Proof. Let $D \in \mathcal{O}_\alpha(G)$. We will show that D contains at least $(k-1)n$ directed cycles. Since for each directed cycle C, the orientation D^C is a different element of $\mathcal{O}_\alpha(G)$, we obtain the result.

The cycle space of D is the set of vectors in $\mathbb{R}^m$ that are linear combinations of signed incidence vectors of cycles of D, where the latter assigns to a cycle C with a direction of traversal a vector which has an entry for every arc of D, which is 1 if a is traversed forward, -1 if it is backward and 0 if $a \notin C$. It is well known, that the cycle space of a strongly connected D can be enumerated by linear combinations of the vectors associated to its directed cycles, see e.g. [21]. Moreover, it can be found in most books on (algebraic) graph theory that the dimension of the cycle space of a weakly connected digraph is $m - n + 1$, see e.g. [22, 26], i.e., D has at least $m - n + 1$ directed cycles. Since D is k-connected it has at least kn edges and the result follows. □

Theorem 15. *Let G be a graph and $k \in \mathbb{N}$. Algorithm 6 enumerates all k-connected orientations of G with delay $O(knm^2)$. If $k \geq 2$ the amortized time is in $O(m^2)$.*

Proof. The correctness and the delay follow directly from Theorem 13 and Theorem 6. Let us compute the amortized time complexity as an average over the delays. Let s be the number of solutions, i.e., the total number of k-connected orientations and t be the number of k-connected outdegree sequences of G. Since for every k-connected outdegree sequence α there are at least $(k-1)n + 1$ orientations, we have that $t \leq \frac{s}{(k-1)n+1}$. Thus there exists some constants c and c', such that the overall runtime of our algorithm is bounded by $cknm^2 t + c'm^2 s \leq cknm^2 \frac{s}{(k-1)n} + c'm^2 = O(m^2)s$, where for the last equality we use $k \geq 2$. Hence the amortized complexity is in $O(m^2)$. □

7 Discussion

We have given a very simple algorithm for enumerating the k-arc-connected orientations of a graph with a considerably low amortized time complexity. The weakness of our approach is that finding the initial solution, i.e., finding a k-connected orientation of a graph, is algorithmically the most complicated part.

The best implementations using splitting off techniques are rather simple and lead to computation time of roughly $O(n^5)$ [19, 28], it would be interesting to find simpler methods.

Our original motivation was the hunt for counterexamples to a conjecture of Neumann-Lara [32] stating that the vertex set of every oriented planar graph (that is no directed cycles of length 2) can be vertex-partitioned into two subsets each inducing an acyclic digraph. It is not hard to see that a minimal counterexample to this conjecture has to be a 2-arc-connected (planar) digraph. However, with a little more work one can actually prove that a minimal counterexample has to be 2-*vertex-connected*, i.e., at east two vertices have to be removed in order to destroy strong connectivity. Deciding if a graph admits a k-vertex-connected orientation can be done in polynomial time if and only if $k \leq 2$ (unless P=NP), see [10, 38]. Thus, this enumeration problem would mostly be viable for $k = 2$ or even moreover restricted to planar graphs. Note that also in [2] it is posed as open problem whether it can be decided in polynomial time, if a mixed graph can be oriented such that it is 2-vertex-connected. A positive answer would allow polynomial time enumeration as in Algorithm 1.

A dual analogue of k-arc-connectivity could be called k-*acyclicity*, where at least k arcs of D have to be contracted in order to destroy its acylicity. As mentioned in the introduction, acyclic orientations can be enumerated with polynomial delay. On the other hand, it is easy to see that a graph G admits a 2-acyclic orientation if and only if G is the cover graph of a poset. The corresponding recognition problem is NP-complete [5] and the proof can be extended to see that testing whether G admits a k-acyclic orientation is NP-complete for any $k \geq 2$.

Using [9, Theorem 13] one can show that enumerating k-acyclic orientations (as well as k-vertex connected orientations for $k > 2$) are DelNP-hard under D-reductions and IncNP-hard under I-reduction. Since with an NP-oracle one can decide if a given partial orientation extends to one of these types, a backtrack search algorithm like Algorithm 1 yields that the above problems can be solved with delay a polynomial number of calls of an NP-oracle. Thus these problems are in the class DelNP (and a fortiori IncNP), and therefore are indeed complete in both settings.

Acknowledgements:

We wish to thank Nadia Creignou and Frédéric Olive for fruitful discussions in an early stage of this paper. Some results related to the ones of this paper were presented at *WEPA 2018: Second Workshop on Enumeration Problems and Applications* held in Pisa in November 2018. The authors wish to thank the organizers. The first author was supported by the French ANR project GaphEn: ANR-15-CE40-0009. The second and third authors were supported by the French ANR project DISTANCIA: ANR-17-CE40-0015. The second author was also supported by ANR grant GATO: ANR-16-CE40-0009-01 and by the Spanish Ministerio de Economía, Industria y Competitividad through grant RYC-2017-22701.

References

[1] O. Aichholzer, J. Cardinal, T. Huynh, K. Knauer, T. Mütze, R. Steiner, and B. Vogtenhuber, *Flip distances between graph orientations*, arXiv e-prints, (2019), p. arXiv:1902.06103.

[2] J. Bang-Jensen, J. Huang, and X. Zhu, *Completing orientations of partially oriented graphs*, Journal of Graph Theory, 87 (2018), pp. 285–304.

[3] V. C. Barbosa and J. L. Szwarcfiter, *Generating all the acyclic orientations of an undirected graph*, Information Processing Letters, 72 (1999), pp. 71 – 74.

[4] E. Brehm, *3-orientations and Schnyder-3-tree-decompositions*, 2000. Diploma Thesis, FU Berlin.

[5] G. Brightwell, *On the complexity of diagram testing*, Order, 10 (1993), pp. 297–303.

[6] A. Conte, *Enumeration Algorithms for Real-World Networks: Efficiency and Beyond*, PhD thesis, Università di Pisa, 2018.

[7] A. Conte, R. Grossi, A. Marino, and R. Rizzi, *Efficient enumeration of graph orientations with sources.*, Discrete Appl. Math., 246 (2018), pp. 22–37.

[8] A. Conte, R. Grossi, A. Marino, R. Rizzi, and L. Versari, *Directing Road Networks by Listing Strong Orientations*, Springer International Publishing, 2016, pp. 83–95.

[9] N. Creignou, M. Kröll, R. Pichler, S. Skritek, and H. Vollmer, *On the complexity of hard enumeration problems*, CoRR, abs/1610.05493 (2016).

[10] O. D. de Gevigney, *On Frank's conjecture on k-connected orientations*, arXiv:1212.4086v1, (2012).

[11] G. Ehrlich, *Loopless algorithms for generating permutations, combinations, and other combinatorial configurations*, J. ACM, 20 (1973), pp. 500–513.

[12] S. Felsner, *Lattice structures from planar graphs*, Electron. J. Combin., 11 (2004). Research Paper 15.

[13] S. Felsner, *Rectangle and square representations of planar graphs*, in Thirty essays on geometric graph theory, Springer, New York, 2013, pp. 213–248.

[14] S. Felsner, H. Schrezenmaier, and R. Steiner, *Pentagon contact representations*, Electron. J. Combin., 25 (2018). Paper 3.39, 38.

[15] A. Frank, *An algorithm for submodular functions on graphs*, Ann. Discrete Math., 16 (1982), pp. 97–120.

[16] A. Frank, *A note on k-strongly connected orientations of an undirected graph*, Discrete Math., 39 (1982), pp. 103–104.

[17] H. Gabow, *Centroids, representations, and submodular flows*, J. Algorithms, 18 (1995), pp. 586 – 628.

[18] H. N. Gabow, *A framework for cost-scaling algorithms for submodular flow problems*, in 34th Annual Symposium on Foundations of Computer Science, Palo Alto, USA, November 1993, 1993, pp. 449–458.

[19] H. N. Gabow, *Efficient splitting off algorithms for graphs*, in Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing, STOC '94, New York, NY, USA, 1994, ACM, pp. 696–705.

[20] P. M. Gilmer and R. A. Litherland, *The duality conjecture in formal knot theory*, Osaka J. Math., 23 (1986), pp. 229–247.

[21] P. M. Gleiss, J. Leydold, and P. F. Stadler, *Circuit bases of strongly connected digraphs.*, Discuss. Math., Graph Theory, 23 (2003), pp. 241–260.

[22] C. Godsil and G. Royle, *Algebraic graph theory.*, vol. 207, New York, NY: Springer, 2001.

[23] D. Gonçalves, B. Lévêque, and A. Pinlou, *Triangle contact representations and duality*, Discrete Comput. Geom., 48 (2012), pp. 239–254.

[24] M. Habib, R. Medina, L. Nourine, and G. Steiner, *Efficient algorithms on distributive lattices*, Discrete Applied Mathematics, 110 (2001), pp. 169 – 187.

[25] S. Iwata and Y. Kobayashi, *An algorithm for minimum cost arc-connectivity orientations*, Algorithmica, 56 (2010), pp. 437–447.

[26] U. Knauer and K. Knauer, *Algebraic graph theory. Morphisms, monoids and matrices (to appear). 2nd revised and extended edition.*, vol. 41, Berlin: De Gruyter, 2nd revised and extended edition ed., 2019.

[27] P. C. B. Lam and H. Zhang, *A distributive lattice on the set of perfect matchings of a plane bipartite graph*, Order, 20 (2003), pp. 13–29.

[28] L. C. Lau and C. K. Yung, *Efficient edge splitting-off algorithms maintaining all-pairs edge-connectivities*, SIAM J. Comput., 42 (2013), pp. 1185–1200.

[29] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland, 1979.

[30] K. Menger, *Zur allgemeinen Kurventheorie*, in Fundamental Mathematics, 1927, pp. 96–115.

[31] C. S. J. A. Nash-Williams, *On orientations, connectivity and odd-vertex-pairings in finite graphs*, Canadian Journal of Mathematics, 12 (1960), p. 555–567.

[32] V. Neumann-Lara, *Vertex colourings in digraphs, Some Problems*, tech. report, Waterloo, Canada, 1985.

[33] J. B. Orlin, *Max flows in $O(nm)$ time, or better.*, in Proceedings of the 45th annual ACM symposium on theory of computing, STOC '13. Palo Alto, CA, USA, June 1–4, 2013, 2013, pp. 765–774.

[34] J. Propp, *Lattice structure for orientations of graphs.* ArXiv: math/0209005, Sept. 2002.

[35] G. Pruesse and F. Ruskey, *Gray codes from antimatroids*, Order, 10 (1993), pp. 239–252.

[36] E. Rémila, *The lattice structure of the set of domino tilings of a polygon*, Theoret. Comput. Sci., 322 (2004), pp. 409–422.

[37] M. B. Squire, *Generating the acyclic orientations of a graph*, J. Algorithms, 26 (1998), pp. 275 – 290.

[38] C. Thomassen, *Strongly 2-connected orientations of graphs*, Journal of Combinatorial Theory, Series B, 110 (2015), pp. 67 – 78.

[39] W. P. Thurston, *Conway's tiling groups*, Amer. Math. Monthly, 97 (1990), pp. 757–773.