

Computational results on the traceability of oriented graphs of small order

AP Burger*

Department of Logistics
University of Stellenbosch
South Africa

apburger@sun.ac.za

Submitted: Jun 10, 2011; Accepted: Nov 19, 2013; Published: Nov 29, 2013

Mathematics Subject Classifications: 05C20, 05C38

Abstract

A digraph D is *traceable* if it contains a path visiting every vertex, and *hypotraceable* if D is not traceable, but $D - v$ is traceable for every vertex $v \in V(D)$. Van Aardt, Frick, Katrenič and Nielsen [Discrete Math. 11(2011), 1273-1280] showed that there exists a hypotraceable oriented graph of order n for every $n \geq 8$, except possibly for $n = 9$ or 11 . These two outstanding existence questions for hypotraceable oriented graphs are settled in this paper — the first in the negative and the second in the affirmative. Furthermore, D is *k-traceable* if D has at least k vertices and each of its induced subdigraphs of order k is traceable. It is known that for $k \leq 6$ every k -traceable oriented graph is traceable and that for $k = 7$ and each $k \geq 9$ there exist nontraceable k -traceable oriented graphs of order $k + 1$. The Traceability Conjecture states that for $k \geq 2$ every k -traceable oriented graph of order $n \geq 2k - 1$ is traceable. In this paper it is shown via computer searches that all 7-traceable and 8-traceable oriented graphs of orders 9, 10 and 11 are traceable, and that all 9-traceable oriented graphs of order 11 are traceable. All hypotraceable graphs of order 10 are also found. Recently, these results are used to prove that the Traceability Conjecture also holds for $k = 7, 8$ and 9 , except possibly when $k = 9$ and $22 \leq n \leq 32$.

Keywords: oriented graph, tournament, hypotraceable, k -traceable, traceability

*Research towards this paper was supported financially by the South African National Research Foundation under grant number GUN 77248.

1 Introduction

Denote the *vertex set*, *arc set*, *minimum degree* and *order* of a digraph D by $V(D)$, $A(D)$, $\delta(D)$ and $n(D)$, respectively. If D is a digraph and $X \subseteq V(D)$, then the subdigraph induced by X in D is denoted by $\langle X \rangle_D$. A directed path (walk, cycle) in a digraph is simply called a *path* (*walk*, *cycle*). The order of a longest path in D is denoted by $\lambda(D)$, and if $n(D) - \lambda(D) = p$, then D is said to be *p-deficient*. A digraph is *hamiltonian* if it contains a cycle that visits every vertex, and *traceable* if it contains a path visiting every vertex. An *oriented graph* is a digraph without 2-cycles. A digraph D is *hypotraceable* if it is not traceable, but $D - v$ is traceable for every $v \in V(D)$. Furthermore, D is *k-traceable* for some $k \in \mathbb{N}$ if D has at least k vertices and each of its induced subdigraphs of order k is traceable. Note, therefore, that a nontraceable oriented graph of order $k + 1$ is *k-traceable* if and only if it is hypotraceable.

It is clear that an oriented graph is 2-traceable if and only if it is a tournament (a complete oriented graph). Thus, for $k = 2$, every *k-traceable* oriented graph is traceable (since it is well known that every tournament contains a hamiltonian path). Van Aardt *et al.* [3] showed that this is also true for $k = 3, 4, 5, 6$. Hence there is no hypotraceable oriented graph of order $3 \leq n \leq 7$. However, the following result was established in [6].

Theorem 1. *There exists a hypotraceable oriented graph of order n for every $n \geq 8$, except possibly for $n = 9$ or 11.*

The existence of hypotraceable oriented graphs seems to be inextricably linked with the *Asymmetric Travelling Salesman Problem* (ATSP)¹ and also play an important role in the so-called *Traceability Conjecture* (TC) [1, 4, 5, 9].

Conjecture 1 (TC). *For $k \geq 2$, every k -traceable oriented graph of order at least $2k - 1$ is traceable.*

The TC is, in turn, motivated by the celebrated *Path Partition Conjecture for Digraphs* (DPPC) [15] dating back to 1982, which has been proven correct for various special cases [8, 2], but seems to be very difficult to settle in general — even the case $a = 1$ has not yet been settled [10, 11, 14].

Conjecture 2 (DPPC). *If D is a digraph and (a, b) is a pair of positive integers such that $a + b = \lambda(D)$, then $V(D)$ has a partition (A, B) such that $\lambda(\langle A \rangle_D) \leq a$ and $\lambda(\langle B \rangle_D) \leq b$.*

The connection between the TC and the DPPC may be seen by considering a 1-deficient oriented digraph D and a pair of positive integers (a, b) satisfying $a \leq b$ and $a + b = \lambda(D)$. Then $n(D) = a + b + 1$ and so $n(D) \geq 2(a + 1) - 1$. Therefore, if the TC is true, D would not be $(a + 1)$ -traceable, *i.e.* D would have a subdigraph H of order

¹Grötschel and Wakabayashi [12, 13] have shown that certain hypotraceable digraphs induce facets of the monotone ATSP polytope, and they speculate that, due to the complicated nature of these facets, it seems unlikely that an explicit characterization of these polytopes will be found.

$a + 1$ for which $\lambda(H) \leq a$, and hence $(V(H), V(D) \setminus V(H))$ would be an (a, b) -partition of D . Consequently the truth of the TC would imply the truth of the DPPC for 1-deficient oriented graphs.

As noted in [3], it may well be that a *stronger* result than the TC holds — as indeed is the case for $k = 2, 3, 4, 5, 6$, mentioned above. As a natural consequence, the following problem was posed in [3].

Problem 1. For $k \geq 2$, find the smallest value $t(k)$ for which every k -traceable oriented graph of order at least $t(k)$ is traceable.

The following partial solution to the above problem is due to Van Aardt *et al.* [3].

Theorem 2. *The parameter $t(k)$ in Problem 1 is well-defined for every $k \geq 2$. Moreover,*

- (a) $t(k) = k$ for $k \leq 6$,
- (b) $t(k) \geq k + 2$ for all $k \geq 7$, except possibly for $k = 8, 10$,
- (c) $9 \leq t(7) \leq 22$,
- (d) $8 \leq t(8) \leq 28$,
- (e) $t(k) \leq 2k^2 - 20k + 59$ for all $k \geq 9$.

The objective in this paper is to investigate the traceability of k -traceable oriented graphs of order at most 11 via computer searches. As a result we strengthen Theorem 1 by settling its two outstanding cases, thereby giving rise to the following existence result.

Theorem 3. *There are no hypotraceable oriented graphs of order $2 < n \leq 7$ or $n = 9$. For all other values of $n > 2$ there exist hypotraceable oriented graphs of order n .*

It is also shown that all 7-traceable and 8-traceable oriented graphs of orders 9, 10 and 11 are traceable and that all 9-traceable graphs of order 11 are traceable. These results improve on Theorem 2(b), since they imply that $t(k) \geq k + 2$ for all $k \geq 7$, except for $k = 8$. We can also now conclude that $t(7) \neq 10, 11, 12$, $t(8) \neq 10, 11, 12$ and that $t(9) \neq 12$. Furthermore, the following two theorems are proved in [1]:

Theorem 4. *Let $k \geq 5$ and suppose n_1 and n_2 are integers such that $k \leq n_1 \leq n_2$ and every k -traceable oriented graph of order n_i is traceable for $i = 1, 2$. If $n = n_1 + n_2 - j$; $j = 1$ or 2 , and*

$$k - 9 \leq n_2 - n_1 \leq 5 \text{ if } j = 1,$$

$$k - 9 < n_2 - n_1 < 5 \text{ if } j = 2,$$

then every k -traceable oriented graph of order n with independence number at least 3 is traceable.

Theorem 5. *Let $7 \leq k \leq 10$ and suppose there exist integers n_1, n_2 , such that $k \leq n_1 \leq n_2$ and every k -traceable oriented graph with independence number 2 and order n_i is traceable for $i = 1, 2$. Then every k -traceable oriented graph with independence number 2 and order $n_1 + n_2 - j$ is traceable, for $j = 1, 2, 3$.*

Using these theorems together with the results in this paper it is proved in [1] that $t(7) = 9$, $t(8) \leq 14$ and $t(9)$ either equals 11 or $13 \leq t(9) \leq 33$.

The remainder of this paper is organised as follows. A number of general, known properties of hypotraceable digraphs are briefly reviewed in §2; these properties may be used to narrow down the search for hypotraceable oriented graphs. An exhaustive search paradigm for hypotraceable graphs is then outlined in §3. In the following section the search results are reported. In §5 we determine all non-isomorphic hypotraceable oriented graphs of order 10, and also show two hypotraceable oriented graphs of order 11. Validation of the search results and some ideas with respect to possible future work follow in §6 and §7.

2 Properties of hypotraceable oriented graphs

The following two simple properties of hypotraceable oriented graphs follow from the definition of hypotraceability [6].

Lemma 1 ([6]). *If D is a hypotraceable digraph, then D has no vertex of in-degree 1 and no vertex of out-degree 1.*

Lemma 2 ([5]). *Let $k \geq 2$ and suppose D is a k -traceable oriented graph of order $n > k$. Then $\delta(D) \geq n - k + 1$.*

The following lemma, due to Frick and Katrenič [9], illustrates the importance of hypotraceable oriented graphs and is key to the results of this paper.

Lemma 3. *If $n > k \geq 3$ and D is a nontraceable k -traceable oriented graph of order n , then D contains a hypotraceable oriented graph of order m as induced subgraph for some $m \in \{k + 1, k + 2, \dots, n\}$.*

Note from Theorem 2 that k -traceable graphs of order at least k are traceable when $2 \leq k \leq 6$. Thus we start our search at $k = 7$. We search for k -traceable graphs of order $n > k$ that are nontraceable. If no such graphs are found, then we have established that all k -traceable graphs of order n are traceable.

In order to make use of Lemma 3 it makes sense to start searching for hypotraceable graphs (*i.e.* the case where $n = k + 1$). When searching for a nontraceable k -traceable graph of order n , the search may be achieved very rapidly when all the hypotraceable graphs up to order n are known, since the (hypotraceable) subgraphs concerned are induced. If it is known that the graphs for which we search are themselves hypotraceable, we may also employ Lemma 1 to accelerate the search slightly. Lemma 2 may also be used, because any vertex in a k -traceable graph cannot have too many non-edges. Another way to accelerate the search, is to divide the search into essentially three possible cases:

Case 1: D has both a sink and a source,

Case 2: D has either a sink or a source, and

Case 3: D has neither a sink nor a source.

The advantage here is, from Lemma 1, that each vertex which is neither a sink nor a source must have in-degree at least 2, and out-degree at least 2. The disadvantage is that one must know which vertices are the sink and the source.

3 The search algorithm

Two versions of essentially the same algorithm are employed:

Version A: In this version the user must specify whether there is a source and/or a sink. Vertex 1 will always be taken as the source (if there is one), and vertex n as the sink (if there is one). We also assume that the graphs for which we search are hypotraceable and therefore contain a directed path of order $n - 1$, namely v_1, \dots, v_{n-1} without loss of generality.

Version B: In this version the algorithm searches for a nontraceable k -traceable graph D of order n by testing, for each graph H in a file, whether there exists a graph D in which H is an induced subgraph.

A more detailed description follows of the essence of the two versions of the algorithm mentioned above.

A computerised exhaustive depth-first ternary tree search approach is adopted to establish the existence or otherwise of nontraceable k -traceable oriented graphs of orders n . The search is performed by calling the recursive algorithm `Hypotraceable(\mathcal{D}, \mathcal{F})` for which a pseudocode listing is shown in Algorithm 1. The algorithm takes as input a set \mathcal{D} of decided arc directions and a set \mathcal{F} of forbidden arc directions — both specified locally. In addition, the parameter n (the order for which a hypotraceable oriented graph is sought) and two $n \times n$ variable matrices \mathbf{A} and \mathbf{B} containing an initial, partial description of a graph H_0 are specified. The algorithm output is a boolean value — the value `True` if a k -traceable oriented graph of order n is found containing the initial graph H_0 , or `False` whenever a branch terminates.

For each recursive call of the algorithm, a node in the search tree with current edge lists \mathcal{D} and \mathcal{F} is investigated as follows. In lines 1–5 all *undecided* arcs are tested for possible assignment of one of the options *up*, *down* or *nonadjacent* without violating any condition for the current graph to be completable to a nontraceable k -traceable graph by calling the function `CheckUndecidedArcs`. If there is an edge that cannot be assigned any of the three options, a contradiction is encountered and `False` is returned whereupon the node terminates. If there are edges that can only be assigned one of the options, the relevant (non)edges are added to \mathcal{D} or \mathcal{F} . This process is repeated until no (non)arcs can be added (or a contradiction is encountered). Lemmas 1 and 2 are incorporated into the function `CheckUndecidedArcs`, but additional rules may also be employed in the function.

Next, a list of undecided arcs \mathcal{U} are created. If \mathcal{U} is empty, a graph has been found and `True` is returned (line 7). Otherwise any edge e from \mathcal{U} is chosen. The allowable

Algorithm 1 Hypotraceable(\mathcal{D}, \mathcal{F})

Input: A graph order n and two $n \times n$ matrices \mathbf{A} and \mathbf{B} containing an initial graph specification (all global variables), a set \mathcal{D} of decided arcs and a set \mathcal{F} of forbidden arcs (local variables).

Output: **True** if a hypotraceable oriented graph of order n exists containing the arcs in \mathcal{D} and not containing the edges in \mathcal{F} , or **False** when a contradiction is found or a branch terminates.

```
1: repeat
2:    $t \leftarrow |\mathcal{D}| + |\mathcal{F}|$ 
3:    $(\mathcal{D}, \mathcal{F}) \leftarrow \text{CheckUndecidedArcs}(\mathcal{D}, \mathcal{F})$ 
4:   if  $\text{Contradiction} = \text{True}$  then return(False)
5: until  $|\mathcal{D}| + |\mathcal{F}| = t$ 
6:  $\mathcal{U} \leftarrow$  set of undecided arcs
7: if  $\mathcal{U} = \emptyset$  then print('graph found'); return(True)
8:  $e \leftarrow$  any edge in  $\mathcal{U}$ 
9:  $\text{edgecanbe} \leftarrow \text{EdgeUpDown}(e)$ 
10: if  $\text{edgecanbe.up}$  and  $\text{Hypotraceable}(\mathcal{D} \cup \{e\}, \mathcal{F})$  then  $\text{count} \leftarrow \text{count} + 1$ 
11: if  $\text{edgecanbe.down}$  and  $\text{Hypotraceable}(\mathcal{D} \cup \{\text{Reverse}(e)\}, \mathcal{F})$  then  $\text{count} \leftarrow \text{count} + 1$ 
12: if  $\text{edgecanbe.nonadjacent}$  and  $\text{Hypotraceable}(\mathcal{D}, \mathcal{F} \cup \{e\})$  then  $\text{count} \leftarrow \text{count} + 1$ 
13: return(False)
```

directions for e are determined via the function `EdgeUpDownEmpty`, and are stored in the boolean variables `edgecanbe.up`, `edgecanbe.down` and `edgecanbe.nonadjacent`. Lines 10–12 contain the recursive calls responsible for the branching in the algorithm. Finally, line 13 is reached whenever the last level of the tree is reached (*i.e.* a graph is found) or when all branches from lines 10–12 have terminated with contradictions, ensuring that backtracking takes place.

Two adjacency matrices \mathbf{A} and \mathbf{B} are maintained as variables during the search process in order to store the current oriented graph. The matrix \mathbf{A} contains the initial graph H_0 together with all assigned arcs and is used to test for the nontraceability of D . The matrix \mathbf{B} contains all allowable arc directions (including the undecided arcs, in which case both directions are allowed) and is used to verify that $D - v$ remains traceable for all $v \in V(D)$. So, essentially, arcs are assigned that keeps the graph stored in \mathbf{A} nontraceable as well as the graph stored in \mathbf{B} minus any vertex traceable.

The algorithm is initialised with $\mathcal{D} = \mathcal{F} = \emptyset$ and by specifying an initial graph H_0 which is stored in the global variables \mathbf{A} and \mathbf{B} . Algorithm 1 was implemented in C++ and was run mostly in parallel on a few Intel(R) Core(TM) 2 Duo processors, each running at a clock speed of 2.4 GHz with 1 GB of RAM, under a Windows XP Pro operating system, but some runs were done in Linux on a Intel(R) Core(TM) 2 Duo processor at 3.3GHz with 4 GB of RAM. The source code for the algorithm can be viewed at [7].

4 Search results

The search results obtained via the algorithmic approach described in §3 are reported in this section. In particular, it is shown that there are no hypotraceable oriented graphs of order 9, that all 7-traceable and 8-traceable oriented graphs of orders 9, 10 and 11 are traceable, and that all 9-traceable graphs of order 11 are traceable. Table 1 contains a survey summary of these results. Note that table entries on the main diagonal represent hypotraceable cases and that a ‘Yes’ on the main diagonal means that there are no hypotraceable graphs.

	$n = 8$	$n = 9$	$n = 10$	$n = 11$
$k = 7$	No	Yes	Yes	Yes
$k = 8$	—	Yes	Yes	Yes
$k = 9$	—	—	No	Yes
$k = 10$	—	—	—	No

Table 1: Traceable orders (n) of k -traceable oriented graphs.

All hypotraceable graphs of order 10 were determined, but only some of order 11. These results are reported in Section 5. Knowledge of all hypotraceable graphs up to order 10 together with Lemma 3 helped considerably to accelerate the search up to order 10 for the cases where n and k differ by more than one, because only Version B of the algorithm was required. For $n = 11$ both versions were required. For example, for $n = 11$ and $k = 7$, graphs that have hypotraceable graphs of orders 8 and 10 as an induced subgraphs were searched for first (there are no hypotraceable graphs of order 9). Having found no such graphs, we conclude from Lemma 3 that if there exists a nontraceable 7-traceable graph of order 10, it must itself be hypotraceable. Knowing that the graphs we are searching for are also hypotraceable, Lemma 1 was used to accelerate the search slightly. The run times of the algorithm are reported in Table 2.

	$n = 8$	$n = 9$	$n = 10$	$n = 11$
$k = 7$	6	1	4	13 290
$k = 8$	—	1607	4	60 759
$k = 9$	—	—	720 946	8 013 292
$k = 10$	—	—	—	—

Table 2: Run times in seconds for establishing the existence of k -traceable oriented graphs of order n .

The results of this section are summarised in the following theorem.

Theorem 6.

- (a) *All 7-traceable and 8-traceable oriented graphs of orders 9, 10 and 11 are traceable.*
- (b) *All 9-traceable oriented graphs of order 11 are traceable.*

5 Hypotraceable oriented graphs of orders 10 and 11

Since the inclusion or exclusion of the arc joining the source with the sink has no effect on the (non)traceability of an oriented graph, it is unnecessary to branch on the concerned arc. We elected to exclude this arc when listing the non-isomorphic hypotraceable graphs of order 10, but it should be kept in mind that for each such graph there are also a graph with the arc included, thus doubling the number of listed non-isomorphic graphs. The algorithm found 128 220 hypotraceable oriented graphs of order 10. Some of these graphs are isomorphic in pairs, and all of them have both a source and a sink. The isomorphism test of *Mathematica* were used to eliminate isomorphic copies of graphs, and it was found that there are 15 719 non-isomorphic oriented hypotraceable graphs of order 10 (with the arc from source to sink excluded). These graphs are listed in [7]. Useful subsets of these graphs may be minimal or maximal hypotraceable graphs. There are 185 non-isomorphic minimal and 681 non-isomorphic maximal hypotraceable oriented graphs of order 10. Note that for the maximal graphs the arc from the source to sink are included. These graphs are also listed in [7]. The algorithm seems too slow to determine all hypotraceable graphs of order 11. However, running Version A of the algorithm for more than a day produced no hypotraceable graphs of order 11.

One way of improving the probability of finding a hypotraceable oriented graph G of order 11 quickly via the algorithmic approach described in §3 is to provide the search algorithm with a fairly large initial oriented graph that is likely to be a subgraph of G . Due to the near-circulant structure of many of the known hypotraceable oriented graphs of other orders, the near-circulant subgraph structure of order 11 in Figure 1(a) shown in bold was provided to Algorithm 1. This resulted in the full hypotraceable oriented graph of order 11 shown in Figure 1(a). The smaller oriented graph of order 11 shown in Figure 1(b) is a subgraph of the graph in (a), and is also hypotraceable. This smaller graph was obtained from the graph in (a) by removing as many arcs as possible without violating its hypotraceability.

6 Validation and verification of the algorithm

The basic working of the algorithm is simple, namely building a tree of graphs by branching on undecided edges until either a graph is found that has no undecided edges (and violates no traceability condition), or until all branches lead to contradictions. Thus the main concern is the correct implementation of the algorithm. It was verified that the algorithm finds all known hypotraceable oriented graphs of order 8. Furthermore, since the algorithm does not take into account isomorphisms of graphs, several orientations of the same graph are found, making the chance of missing a graph much smaller. For example, for order 10 the algorithm counts each hypotraceable graph slightly more than 8 times, on average.

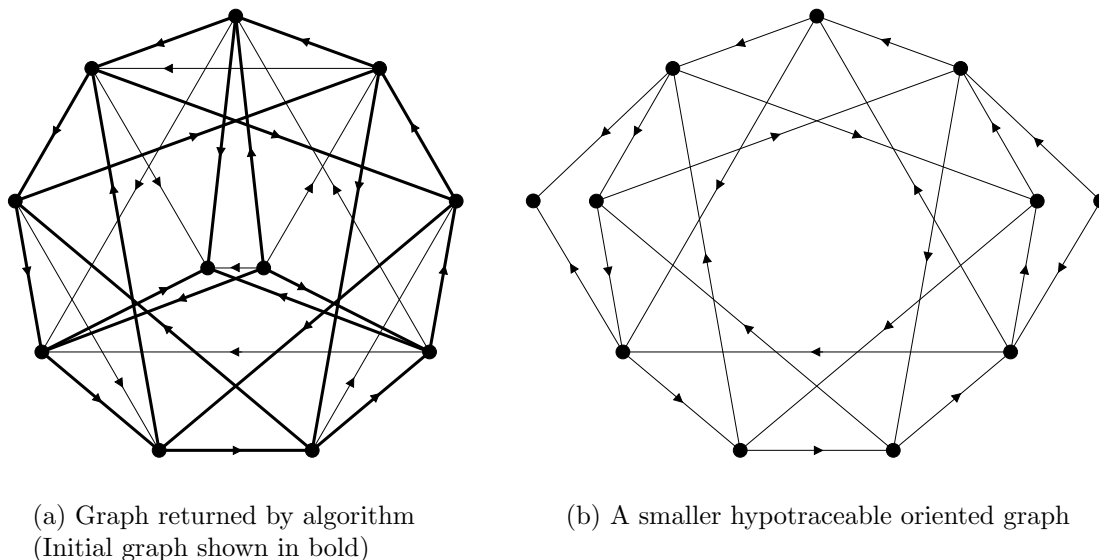


Figure 1: Hypotraceable oriented graphs of order 11.

7 Further work

Perhaps the most obvious avenue of further investigation is to attempt to improve the bounds on $t(k)$ in Theorem 2. For instance, so far there is no evidence to answer the following question in the affirmative.

Question 1. Is there a value of $k \in \mathbb{N}$ for which $t(k) > k + 2$?

It is also not known whether the following question may be answered in the affirmative.

Question 2. Is it true that if all k -traceable oriented graphs of order n for some $n > k$ are traceable, then all k -traceable oriented graphs of orders larger than n are also traceable?

If strong enough properties of untraceable k -traceable oriented graphs are employed, computer searches may be used in an attempt to answer Questions 1 and 2. Answering these questions will constitute significant progress towards solving the TC.

Algorithm 1 may also be adapted for digraphs (as opposed to oriented graphs) by branching into four possibilities instead of three.

References

- [1] S.A. van Aardt, A.P. Burger, J.E. Dunbar, M. Frick, J. Harris & J. Singleton. *An Iterative Approach to the Traceability Conjecture for Oriented Graphs*, Electron. J. Combin., **20(1)** (2013), P59.
- [2] S.A. van Aardt, G. Dlamini, J.E. Dunbar, M. Frick & O.R. Oellermann. *The directed path partition conjecture*, Discuss. Math. Graph Theory, **25** (2005), 331–343.

- [3] S.A. van Aardt, J.E. Dunbar, M. Frick, P. Katrenič, M.H. Nielsen & O.R. Oellermann. *Traceability of k -traceable oriented graphs*, Discrete Math., **310** (2010), 1325–1333.
- [4] S.A. van Aardt, J.E. Dunbar, M. Frick & M.H. Nielsen. *Cycles in k -traceable oriented graphs*, Discrete Math., **311** (2011), 2085–2094.
- [5] S.A. van Aardt, J.E. Dunbar, M. Frick, M.H. Nielsen & O.R. Oellermann. *A traceability conjecture for oriented graphs*, Electron. J. Combin., **15(1)** (2008), R150.
- [6] S.A. van Aardt, M. Frick, P. Katrenič & M.H. Nielsen. *The order of hypotraceable oriented graphs*, Discrete Math., **311** (2011), 1273–1280.
- [7] A.P. Burger. 2011. *A repository of hypotraceable oriented graphs*, [Online], [Cited November 22, 2013], Available at <http://www.vuuren.co.za> → Repositories.
- [8] J. Bang-Jensen, M.H. Nielsen & A. Yeo. *Longest path partitions in generalisations of tournaments*, Discrete Math., **306** (2006), 1830–1839.
- [9] M. Frick & P. Katrenič. *Progress on the traceability conjecture*, Discrete Math. and Theor. Comp. Science, **10(3)** (2008), 105–114.
- [10] H. Galeana-Sánchez & R. Gómez. *Independent sets and nonaugmentable paths in generalisations of tournaments*, Discrete Math., **308(12)** (2008), 2460–2472.
- [11] H. Galeana-Sánchez & H.A. Rincón-Mejía *Independent sets which meet all longest paths*, Discrete Math., **152** (1996), 141–145.
- [12] M. Grötschel & Y. Wakabayashi. *On the structure of the monotone asymmetric traveling salesman polytope I: Hypo-hamiltonian facets*, Discrete Math., **34** (1981), 43–59.
- [13] M. Grötschel & Y. Wakabayashi. *On the structure of the monotone asymmetric traveling salesman polytope II: Hypotraceable facets*, Math. Programming Stud., **14** (1981), 77–97.
- [14] F. Havet. *Stable set meeting every longest path*, Discrete Math., **289** (2004), 169–173.
- [15] J.M. Laborde, C. Payan & N.H. Xuong. 1982. *Independent sets and longest directed paths in digraphs*, pp. 173–177 in *Graphs and other combinatorial topics*, Teubner-Texte Math., **59**.