On the End-Vertex Problem

Jesse Beisegel — **BTU Cottbus-Senftenberg, Cottbus, Germany**
Carolin Denkert — **BTU Cottbus-Senftenberg, Cottbus, Germany**
Matjaž Krnc — **University of Primorska, Koper, Slovenia**
Ekkehard Köhler — **BTU Cottbus-Senftenberg, Cottbus, Germany**
Nevena Mitrović — **University of Primorska, Koper, Slovenia**
Robert Scheffler — **BTU Cottbus-Senftenberg, Cottbus, Germany**
Martin Strehler — **BTU Cottbus-Senftenberg, Cottbus, Germany**

Abstract

For a given graph search algorithm and a graph G, a vertex v is said to be an *end-vertex* if there exists a corresponding search ordering of the vertices such that v is the last vertex in this ordering. It is known that the complexity of recognizing the end-vertices in general graphs is $\mathcal{NP}$-hard for several search algorithms, such as Breadth First search, Depth First Search, and their lexicographic variants. This motivated the study of end-vertices with respect to some other search algorithms. In this work we consider Maximum Cardinality Search (MCS) and Maximum Neiborhood Search (MNS) and present results concerning the complexity of the end-vertex problem with respect to these search methods.

1 Introduction

Graph algorithms frequently use various graph traversal algorithms as an important part of their execution. In general, a graph search represents a method of ordering vertices of a graph, and depending on the way our method picks the next visited vertex, various search methods are defined. Arguably the two most known methods are Depth-First Search (DFS) and Breath-First Search (BFS). Since every search method produces some ordering of the vertices, it is desired to understand the structure of that ordering – the correctness of algorithms that are based on some search method can be proved using the structural characterization of the ordering. In 1976 Rose, Tarjan and Lueker introduced *Lexicographic Breadth-First Search* (LBFS) [6] which, unlike BFS and DFS, admits a characterization that was used to obtain a linear-time algorithm for recognition of chordal graphs [6]. This algorithm was further simplified in 1984, by the introduction of yet another graph search method, called *Maximal Cardinality Search* (MCS), see Tarjan [7]. At an arbitrary step of iteration, this method chooses a non-visited vertex with the largest number of its visited neighbors.

The above-mentioned notion of LBFS was naturally extended to Lexicographic Depth First Search (LDFS), which was introduced as well as characterized with forbidden structures by Corneil et al. [5]. The authors identified the core common property of both lexicographic search methods, which led to the definition of a new graph search called *Maximal Neighborhood Search* (MNS), where each vertex is assumed to be uniquely labeled, and in the process it maintains a subset of its visited neighboring labels, see [5]. At an arbitrary step of iteration, this method chooses a non-visited vertex with the maximal label under set inclusion, and then appends its (unique) label to the lists of its non-visited neighbors.

In 2010, Corneil et al. [4] described a related *end-vertex problem*, which asks whether a certain vertex can correspond to the last visited vertex in a chosen graph search algorithm. The authors

*This work is supported in part by the Slovenian Research Agency (bilateral project BI-DE/17-19-18 and a Young Researchers Grant).

showed that the LBFS end-vertex problem is $\mathcal{NP}$-complete for general graphs. A few years later, Charbit et al. proved a similar result for DFS, BFS and LDFS [3]. The end-vertex problem remained unsolved for MNS and MCS. We give a proof that it is $\mathcal{NP}$-complete in both cases, obtaining a complete overview of the problems' complexities. Moreover, we present polynomial algorithms for certain classes of graphs. For a more specific overview of the mentioned results, see Table 1.

Search algorithm	BFS	LBFS	DFS	LDFS	MCS	MNS
All Graphs	NPC	NPC	NPC	NPC	**NPC**	**NPC**
Weakly Chordal	NPC [3]	NPC [4]	NPC	NPC [3]	?	**NPC**
Chordal	?	?	NPC	?	?	P [1]
Interval	?	P [4]	?	?	**P**	P
Split	P [3]	P [3]	NPC [3]	P [3]	**P**	P

Table 1: Complexity of the end-vertex problem. Bold results are ours.

The rest of this extended abstract is structured into two parts concerning MNS and MCS, respectively. In Section 2 we describe a reduction from 3-SAT to the MNS end-vertex problem, for the class of weakly chordal graphs. In Section 3, we show a reduction from 3-SAT to the MCS end-vertex problem as well as some additional polynomial results for certain graph familes. For graph-theoretic notions used throughout the paper we refer the reader to [2].

2 Maximum Neighborhood Search

In this section we describe an $\mathcal{NP}$-completeness proof for the end-vertex-problem of MNS in weakly chordal graphs, by using a reduction from 3-SAT. Let $\mathcal{I}$ be an instance of 3-SAT. We construct the corresponding graph $G(\mathcal{I})$ as follows (see Fig. 1 for an example). The variables (together with their negations) are represented by the complement of a matching. Non-adjacent vertices correspond to the two literals of the same variable. Each clause is represented by one vertex. All the clause vertices correspond to an independent set. Furthermore the clause vertices are connected to all literal vertices, except to the literals that are elements of the clause. Additionally, we have a vertex b that is connected to all literal vertices, a vertex s connected to all literal vertices and all clause vertices , as well as a vertex t, which is connected to all vertices but s.

Lemma 1. *For an instance $\mathcal{I}$ of 3-SAT and graph $G(\mathcal{I})$ as above, the following holds:*

i) If MNS chooses a clause vertex before vertex b or b before s, then t is not the end vertex.

ii) If t is the end vertex of MNS, then the algorithm has to choose s and an assignment at the beginning of the search. Any of these choices can be made.

iii) For each instance $\mathcal{I}$ of 3-SAT, the graph $G(\mathcal{I})$ is weakly chordal.

From the above statements, it clearly follows that the 3-SAT instance $\mathcal{I}$ has a satisfying assignment if and only if t can be the end-vertex of a MNS-search on $G(\mathcal{I})$. We state the main theorem concerning the end-vertex problem for the Maximum Neighborhood Search.

Theorem 2. *The MNS end-vertex-problem is $\mathcal{NP}$-complete for weakly chordal graphs.*

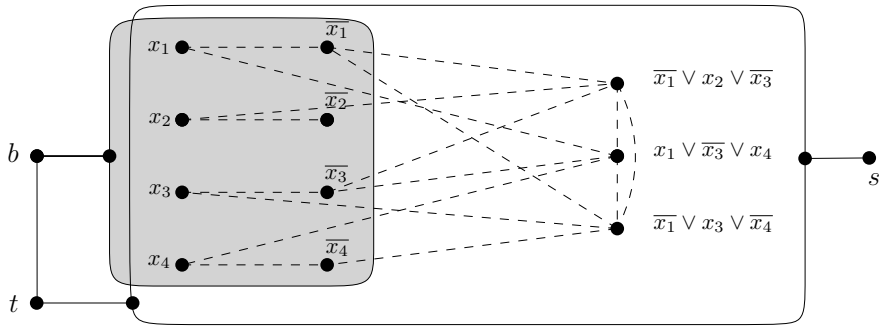


Figure 1: The $\mathcal{NP}$-completeness construction for the MNS end-vertex-problem. The depicted graph is $G(\mathcal{I})$ for $\mathcal{I} = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_3 \vee \overline{x_4})$. In both boxes only non-edges are displayed by dashed lines. The connection of a vertex with a box means that the vertex is connected to all vertices in this box.

3 Maximum Cardinality Search

In this section we show $\mathcal{NP}$-completeness of the end-vertex problem for the MCS, where we construct a polynomial reduction from 3-SAT. Let $\mathcal{I}$ be an arbitrary instance of 3-SAT, with $x_1, \ldots, x_\ell$ being literals of $\mathcal{I}$, respectively. We construct a corresponding graph $G(\mathcal{I})$ as follows (see Fig. 2 for an example). Each literal x_i of $\mathcal{I}$ is represented by two disjoint copies of K_2, where the first edge a_ib_i represents a literal, the second edge c_id_i represents its negation, while each clause is represented by a triangle. Vertices belonging to consecutive literals are fully adjacent and each edge connecting two vertices of consecutive literals is subdivided. For each literal x_i one of the two vertices c_i or d_i (resp. a_i or b_i) is adjacent to the three vertices of any clause which contains the literal x_i (resp. the negation of x_i). We add the vertices s and s', where $N(s) = \{a_1, b_1, c_1, d_1\}$ and $N(s') = \{s\}$. Additionally, the graph contains an "almost" clique C with $3(4l + 8(l-1)) + 4$ vertices: three vertices for each literal and auxiliary vertex and 4 connector vertices (inducing a subgraph isomorphic to $2K_2$ which are each completely connected to the vertices a_l, b_l, c_l, d_l) – thus all but 4 connector vertices form a clique. Finally, we add the vertex t which is adjacent to all clause vertices.

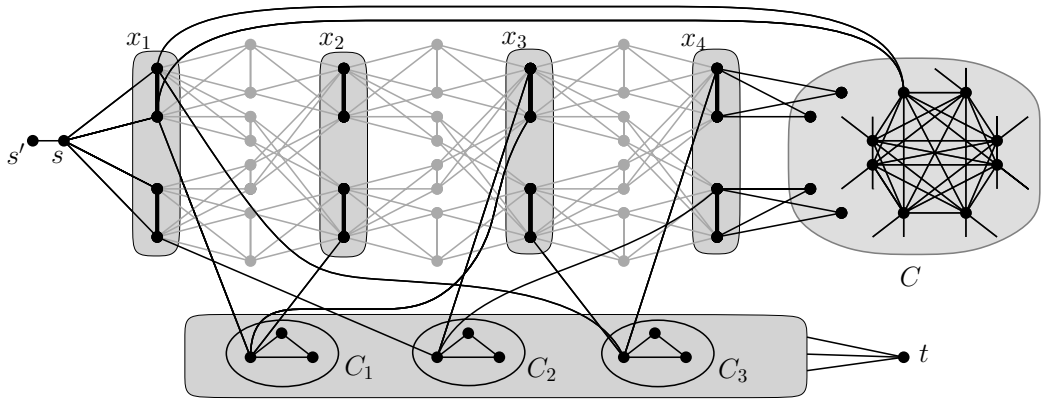


Figure 2: The $\mathcal{NP}$-completeness construction for the end-vertex-problem of MCS with $\mathcal{I} = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_3 \vee \overline{x_4})$.

By inspecting the construction of $G(\mathcal{I})$, observe the following properties.

Lemma 3. *For an instance $\mathcal{I}$ of 3-SAT and graph $G(\mathcal{I})$ as above, the following holds.*

i) If $\mathcal{I}$ has a satisfying assignment $\mathcal{A}$, then there is an MCS on $G(\mathcal{I})$ that ends in t.

ii) Any maximum cardinality search that ends in t must begin in s or s'.

iii) If $\mathcal{I}$ does not admit a satisfying assignment, then t can not be an end-vertex of $G(I)$.

The statements from the above lemma clearly imply that any instance $\mathcal{I}$ of 3-SAT has a satisfying assignment if and only if t is a possible MCS end-vertex of $G(\mathcal{I})$, concluding our reduction.

Theorem 4. *The MCS end-vertex problem is $\mathcal{NP}$-complete.*

For the families of split graphs and interval graphs, we give the following characterization of the possible end-vertices.

Theorem 5. *Let $G = (V, E)$ be a split graph. Then $t \in V$ is the last vertex of some MCS-ordering σ of G if and only if t is simplicial and the neighborhoods of the vertices with smaller degree than t are totally ordered by the set inclusion.*

Theorem 6. *Let $G = (V, E)$ be an interval graph and let $C_1, C_2, \ldots, C_k$ be a linear order of the maximal cliques of G. Then $t \in V$ is the last vertex of some MCS-ordering σ of G if and only if*

i) t is simplicial, and

ii) If C_i is the unique clique containing t, then either $i = 1$, $i = k$ or $C_{i-1} \cap C_i \subseteq C_i \cap C_{i+1}$, and

$$|C_i \cap C_{i+1}| \leq |C_j \cap C_{j+1}| \text{ for all } j > i,$$

or the same holds for the reverse order $C_k, C_{k-1}, \ldots, C_1$.

Corollary 7. *The MCS end-vertex problem is solvable in polynomial time for split graphs and for interval graphs.*

References

[1] Anne Berry, Jean R.S. Blair, Jean-Paul Bordat, and Genevieve Simonet. Graph extremities defined by search algorithms. *Algorithms*, 3(2):100–124, 2010.

[2] Andreas Brandstädt, Jeremy P. Spinrad, and Van Bang Lee. *Graph classes: A Survey*, volume 3. SIAM, 1999.

[3] Pierre Charbit, Michel Habib, and Antoine Mamcarz. Influence of the tie-break rule on the end-vertex problem. *Discrete Mathematics and Theoretical Computer Science*, 16(2):57, 2014.

[4] Derek G. Corneil, Ekkehard Köhler, and Jean-Marc Lanlignel. On end-vertices of lexicographic breadth first searches. *Discrete Applied Mathematics*, 158(5):434–443, 2010.

[5] Derek G. Corneil and Richard M. Krueger. A unified view of graph searching. *SIAM Journal on Discrete Mathematics*, 22(4):1259–1276, 2008.

[6] Donald J Rose, R Endre Tarjan, and George S Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.

[7] Robert E. Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566–579, 1984.