Graphs where Search Methods are Indistinguishable

Matjaž Krnc and Nevena Pivač*
University of Primorska,
Faculty of Mathematics, Natural Sciences and Information Technologies,
Glagoljaška 8, 6000 Koper, Slovenia
`matjaz.krnc@upr.si nevena.pivac@iam.upr.si`

Abstract

Graph searching is one of the simplest and most widely used tools in graph algorithms. Every graph search method is defined using some particular selection rule, and the analysis of the corresponding vertex orderings can aid greatly in devising algorithms, writing proofs of correctness, or recognition of various graph families.

We study graphs where the sets of vertex orderings produced by two different search methods coincide. We characterise such graph families for ten pairs from the best-known set of graph searches: Breadth First Search (BFS), Depth First Search (DFS), Lexicographic Breadth First Search (LexBFS) and Lexicographic Depth First Search (LexDFS), and Maximal Neighborhood Search (MNS).

Keywords graph search methods, breadth first search, depth first search

1 Introduction

Graph search methods (for instance, Depth First Search and Breadth First Search) are among essential concepts classically taught at the undergraduate level of computer science faculties worldwide. Various types of graph searches have been studied since the 19th century, and used to solve diverse problems, from solving mazes, to linear-time recognition of interval graphs, finding minimal path-cover of co-comparability graphs, finding perfect elimination order, or optimal coloring of a chordal graph, and many others [1, 4, 7, 8, 10, 11].

In its most general form, a *graph search* (also *generic search* [5]) is a method of traversing vertices of a given graph such that every prefix of the obtained vertex ordering induces a connected graph. This general definition of a graph search leaves much freedom for a selection rule determining which node is chosen next. By defining some specific rule that restricts this choice, various different graph search methods are defined. Other search methods that we focus on in this paper are Breadth First Search, Depth First Search, Lexicographic Breadth First Search, Lexicographic Depth First Search, and Maximal Neighborhood Search.

We briefly present the studied graph search methods in Section 2, and then state the obtained results in Section 3. Due to lack of space we omit the proofs and provide some directions for further work in Section 4. All proofs are available in the full version of paper on www.arxiv.org.

*This work is supported in part by the Slovenian Research Agency (research programs P1-0285 and P1-0383, research projects N1-0102, J1-9110, N1-0160, N1-0209 and Young Researchers Grant).

2 Preliminaries

We now briefly describe the above-mentioned graph search methods, and give the formal definitions. Note that the definitions below are not given in a historically standard form, but rather as so-called *three-point conditions*, due to Corneil and Kruger [5] and also Brändstadt et. al. [3]. Two vertices $u, v \in V(G)$ satisfy the relation $u <_\sigma v$ if u appears before v in the ordering $\sigma : V(G) \to \{1, 2, \dots, n\}$ of vertices in G.

Breadth First Search (BFS), first introduced in 1959 by Moore [9], is a restriction of a generic search which puts unvisited vertices in a queue and visits a first vertex from the queue in the next iteration. After visiting a particular vertex, all its unvisited neighbors are put at the end of the queue, in an arbitrary order.

Definition 2.0.1 *An ordering σ of V is a BFS-ordering if and only if the following holds: if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then there exists a vertex d such that $d <_\sigma a$ and $db \in E$.*

Any BFS ordering of a graph G starting in a vertex v results in a rooted tree (with root v), which contains the shortest paths from v to any other vertex in G (see [6]). We use this property implicitly throughout the paper.

Depth First Search (DFS), in contrast with the BFS, traverses the graph as deeply as possible, visiting a neighbor of the last visited vertex whenever it is possible, and backtracking only when all the neighbors of the last visited vertex are already visited. In DFS, the unvisited vertices are put on top of a stack, so visiting a first vertex in a stack means that we always visit a neighbor of the most recently visited vertex.

Definition 2.0.2 *An ordering σ of V is a DFS-ordering if and only if the following holds: if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then there exists a vertex d such that $a <_\sigma d <_\sigma b$ and $db \in E$.*

Lexicographic Breadth First Search (LexBFS) was introduced in the 1970s by Rose, Tarjan and Lueker [10] as a part of an algorithm for recognizing chordal graphs in linear time. Since then, it has been used in many graph algorithms mainly for the recognition of various graph classes.

DOI https://doi.org/10.18690/978-961-286-516-0.3
ISBN 978-961-286-516-0

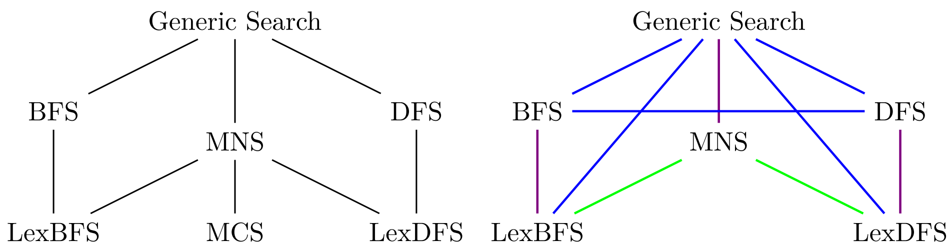

Figure 1: On the left: Hasse diagram showing how graph searches are refinements of one another. On the right is a summary of our results: Green pairs are equivalent on $\{P_4, C_4\}$-free graphs. Violet pairs are equivalent on {pan, diamond}-free graphs. Blue pairs are equivalent on {paw, diamond, P_4, C_4}-free graphs.

Definition 2.0.3 *An ordering σ of V is a LexBFS ordering if and only if the following holds: if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then there exists a vertex d such that $d <_\sigma a$ and $db \in E$ and $dc \notin E$.*

LexBFS is a restricted version of Breadth First Search, where the usual queue of vertices is replaced by a queue of unordered subsets of the vertices which is sometimes refined, but never reordered.

Lexicographic Depth First Search (LexDFS) was introduced in 2008 by Corneil and Krueger [5] and represents a special instance of a Depth First Search.

Definition 2.0.4 *An ordering σ of V is a LexDFS ordering if and only if the following holds: if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then there exists a vertex d such that $a <_\sigma d <_\sigma b$ and $db \in E$ and $dc \notin E$.*

Maximal Neighborhood Search (MNS), introduced in 2008 by Corneil and Krueger [5], is a common generalization of LexBFS, LexDFS, and MCS, and also of Maximal Label Search (see [2] for definition).

Definition 2.0.5 *An ordering σ of V is an MNS ordering if and only if the following statement holds: If $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then there exists a vertex d with $d <_\sigma b$ and $db \in E$ and $dc \notin E$.*

The MNS algorithm uses the set of integers as the label, and at every step of iteration chooses the vertex with maximal label under set inclusion.

Corneil [5] exposed an interesting structural aspect of graph searches: the particular search methods can be seen as restrictions, or special instances of some more general search methods. For six well-known graph search methods they present a depiction, similar to the one in Figure 1, showing how those methods are related under the set inclusion. For example, every LexBFS ordering is at the same time an instance of BFS and MNS ordering of the same graph. Similarly, every LexDFS ordering is at the same time also an instance of MNS, and of DFS (see Figure 1). The reverse, however, is not true, and there exist orderings that are BFS and MNS, but not LexBFS, or that are DFS and MNS but not LexDFS.

3 Problem description and results

Since the connections in Figure 1 represent relations of inclusion, it is natural to ask under which conditions on a graph G the particular inclusion holds also in the converse direction. More formally, we say that two search methods are *equivalent on a graph G* if the sets of vertex orderings produced by both of them are the same. We say that two graph search methods are *equivalent on a graph class $\mathcal{G}$* if they are equivalent on every member $G \in \mathcal{G}$. Perhaps surprisingly, three different graph families suffice to describe graph classes equivalent for the ten pairs of graph search methods that we consider. Those are described in Theorems 3.1 to 3.3 below, but first we need a few more definitions.

All the graphs considered in the paper are finite and connected. A *k-pan* is a graph consisting of a k-cycle, with a pendant vertex added to it. We say that a graph is *pan-free* if it does not contain a pan of any size as an induced subgraph. A 3-pan is also known as a *paw graph.*

Theorem 3.1 *Let G be a connected graph. Then the following is equivalent:*

A1. *Graph G is $\{P_4, C_4$, paw, diamond$\}$-free.*
A2. *Every graph search of G is a DFS ordering of G.*
A3. *Every graph search of G is a BFS ordering of G.*
A4. *Any vertex-order of G is a BFS, if and only if it is a DFS.*

Theorem 3.2 *Let G be a connected graph. Then the following is equivalent:*

B1. *Graph G is {pan, diamond}-free.*
B2. *Every DFS ordering of G is a LexDFS ordering of G.*
B3. *Every BFS ordering of G is a LexBFS ordering of G.*
B4. *Every graph search of G is an MNS ordering of G.*

Theorem 3.3 *Let G be a connected graph. Then the following is equivalent:*

C1. *Graph G is $\{P_4, C_4\}$-free.*
C2. *Every MNS ordering of G is a LexDFS ordering of G.*
C3. *Every MNS ordering of G is a LexBFS ordering of G.*

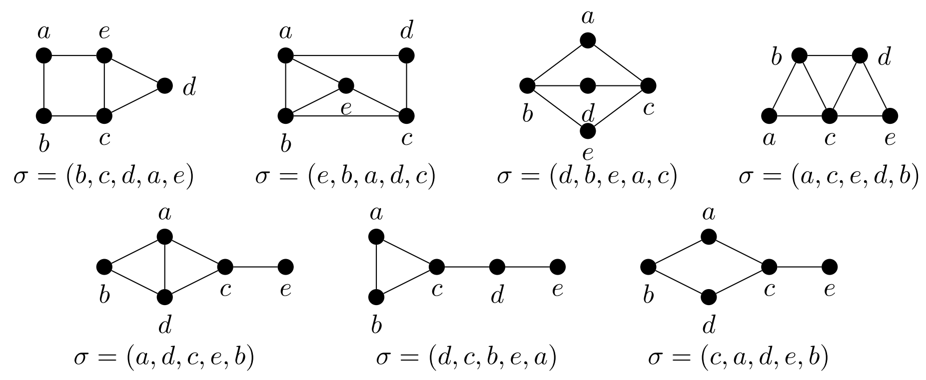

Figure 2: Graphs and corresponding orderings that are MNS and not MCS orderings.

From Theorems 3.1 and 3.2 we can immediately derive similar results for two additional pairs of graph search methods.

Corollary 3.3.1 *Let G be a connected graph. Then the following is equivalent:*

A1. Graph G is $\{P_4, C_4,$ paw, diamond$\}$-free.
A5. Every graph search of G is a LDFS ordering of G.
A6. Every graph search of G is a LBFS ordering of G.

4 Conclusion and further work

In this paper we consider the major graph search methods and study the graphs in which vertex-orders of one type coincide with vertex-orders of some other type. Interestingly, three different graph families suffice to describe graph classes equivalent for the ten pairs of graph search methods that we consider, which provides an additional aspect of similarities between the studied search methods.

Among the natural graph search methods not yet considered in this setting would be the *Maximum Cardinality Search* (MCS), introduced in 1984 (for definition see Tarjan and Yannakakis [12]). As shown on Figure 1, every MCS is a special case of an MNS vertex-order. While it is easy to verify that $\{P_4, C_4, \text{paw, diamond}\}$-free graphs do not distinguish between MNS and MCS vertex orders, Figure 2 provides examples of graphs which admit MNS, but not MNS vertex orders. Characterising graphs equivalent for MNS and MCS remains an open question.

Acknowledgements

The authors would like to thank prof. Martin Milanič for the initial suggestion of the problem, and to Ekki Köhler and his reseach group, for introducing the diverse world of graph searches to us.

References

[1] Beisegel, J. Characterising AT-free graphs with BFS. In *Graph-Theoretic Concepts in Computer Science* (2018), A. Brandstädt, E. Köhler, and K. Meer, Eds., pp. 15–26.

[2] Berry, A., Krueger, R., and Simonet, G. Maximal label search algorithms to compute perfect and minimal elimination orderings. *SIAM Journal on Discrete Mathematics 23*, 1 (2009), 428–446.

[3] Brandstädt, A., Dragan, F. F., and Nicolai, F. LexBFS-orderings and powers of chordal graphs. *Discrete Math. 171*, 1-3 (1997), 27–42.

[4] Corneil, D. G., Dusart, J., Habib, M., and Kohler, E. On the power of graph searching for cocomparability graphs. *SIAM Journal on Discrete Mathematics 30*, 1 (2016), 569–591.

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

[6] Even, S. *Graph algorithms.* Cambridge University Press, 2011.

[7] Golumbic, M. *Algorithmic Graph Theory and Perfect Graphs.* Annals of Discrete Mathematics, Volume 57. Elsevier, 2004, pp. 98–99.

[8] Köhler, E., and Mouatadid, L. Linear time lexdfs on cocomparability graphs. In *Scandinavian Workshop on Algorithm Theory* (2014), Springer, pp. 319–330.

[9] Moore, E. F. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959* (1959), pp. 285–292.

[10] Rose, D. J., Lueker, G. S., and Tarjan, R. E. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing 5*, 2 (1976), 266–283.

[11] Tarjan, R. E. Depth-first search and linear graph algorithms. *SIAM journal on computing 1*, 2 (1972), 146–160.

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