

# Linear-Time Algorithms for Scattering Number and Hamilton-Connectivity of Interval Graphs <sup>\*</sup>

Hajo Broersma<sup>1</sup>, Jiří Fiala<sup>2\*\*</sup>, Petr A. Golovach<sup>3</sup>, Tomáš Kaiser<sup>4\*\*\*</sup>, Daniël Paulusma<sup>5†</sup>,  
and Andrzej Proskurowski<sup>6</sup>

<sup>1</sup> Faculty of EEMCS, University of Twente, Enschede, [h.j.broersma@utwente.nl](mailto:h.j.broersma@utwente.nl)

<sup>2</sup> Department of Applied Mathematics, Charles University, Prague, [fiala@kam.mff.cuni.cz](mailto:fiala@kam.mff.cuni.cz)

<sup>3</sup> Institute of Computer Science, University of Bergen, [petr.golovach@ii.uib.no](mailto:petr.golovach@ii.uib.no)

<sup>4</sup> Department of Mathematics, Institute for Theoretical Computer Science (CE-ITI)  
and the European Centre of Excellence NTIS (New Technologies for the Information Society),  
University of West Bohemia, Plzeň, [kaisert@kma.zcu.cz](mailto:kaisert@kma.zcu.cz)

<sup>5</sup> School of Engineering and Computing Sciences, Durham University, [daniel.paulusma@durham.ac.uk](mailto:daniel.paulusma@durham.ac.uk)

<sup>6</sup> Department of Computer Science, University of Oregon, Eugene, [andrzej@cs.uoregon.edu](mailto:andrzej@cs.uoregon.edu)

**Abstract.** We prove that for all  $k \leq -1$  an interval graph is  $-(k + 1)$ -Hamilton-connected if and only if its scattering number is at most  $k$ . This complements a previously known fact that an interval graph has a nonnegative scattering number if and only if it contains a Hamilton cycle, as well as a characterization of interval graphs with positive scattering numbers in terms of the minimum size of a path cover. We also give an  $O(n + m)$  time algorithm for computing the scattering number of an interval graph with  $n$  vertices and  $m$  edges, which improves the previously best-known  $O(n^3)$  time bound for solving this problem. As a consequence of our two results, the maximum  $k$  for which an interval graph is  $k$ -Hamilton-connected can be computed in  $O(n + m)$  time.

## 1 Introduction

The HAMILTON CYCLE problem is that of testing whether a given graph has a Hamilton cycle, i.e., a cycle passing through all the vertices. This problem is a notorious NP-complete problem, which remains NP-complete on many graph classes such as the classes of planar cubic 3-connected graphs [23], chordal bipartite graphs [37], and strongly chordal split graphs [37]. Bertossi and Bonucelli [6] proved that HAMILTON CYCLE is NP-complete for undirected path graphs, double interval graphs and rectangle graphs, all three of which are classes of intersection graphs that contain the class of interval graphs. A graph  $G$  is an *interval graph* if it is the intersection graph of a set of closed intervals on the real line, i.e., the vertices of  $G$  correspond to the intervals and two vertices are adjacent in  $G$  if and only if their intervals have at least one point in common. For interval graphs, Keil [29] showed in 1985 that HAMILTON CYCLE can be solved in  $O(n + m)$  time, thereby strengthening an earlier result of Bertossi [5] for *proper* interval graphs, which are interval graphs that have a closed interval representation, in which no interval is properly contained in another one. The approach of Keil was later extended by Damaschke [18] to an  $O(n^5)$ -time algorithm for circular-arc graphs. By using an algorithm that computes the so-called bump number of a graph, Deogun and Steiner [21] proved that the HAMILTON CYCLE is solvable in polynomial time on cocomparability graphs.

<sup>\*</sup> The paper was supported by Royal Society Joint Project Grant JP090172. An extended abstract of it appeared in the Proceedings of WG 2013 [7].

<sup>\*\*</sup> Author supported by the GraDR-EuroGIGA project GIG/11/E023 and by the project Kontakt LH12095.

<sup>\*\*\*</sup> CE-ITI is supported by project P202/12/G061 of the Czech Science Foundation.

<sup>†</sup> Author supported by EPSRC (EP/G043434/1).

Recently, Corneil, Dalton and Habib [17] proved and further extended this result by using an LexBFS search algorithm.

In this paper we examine whether the linear-time result of Keil [29] can be extended on interval graphs to hold for other connectivity properties, which are NP-complete or coNP-complete to verify in general. This line of research is well embedded in the literature. Before surveying existing work for interval graphs and presenting our new results, we first give necessary terminology, and in addition, we formally introduce the decision problems considered (we also state their computational complexity for general graphs).

## 1.1 Connectivity Properties and Corresponding Decision Problems

We only consider undirected finite graphs with no self-loops and no multiple edges. We refer to the textbook of Bondy and Murty [8] for any undefined graph terminology. Throughout the paper we let  $n$  and  $m$  denote the number of vertices and edges, respectively, of the input graph.

Let  $G = (V, E)$  be a graph. If  $G$  has a *Hamilton cycle*, i.e., a cycle containing all the vertices of  $G$ , then  $G$  is *hamiltonian*. Recall that the corresponding NP-complete decision problem is called HAMILTON CYCLE. If  $G$  contains a *Hamilton path*, i.e., a path containing all the vertices of  $G$ , then  $G$  is *traceable*. In this case, the corresponding decision problem is called the HAMILTON PATH problem, which is also well known to be NP-complete (cf. [22]).

The problems 1-HAMILTON PATH and 2-HAMILTON PATH are those of testing whether a given graph has a Hamilton path that starts in some given vertex  $u$  or that is between two given vertices  $u$  and  $v$ , respectively. The LONGEST PATH problem is to compute the maximum length of a path in a given graph. All three problems are NP-hard by a straightforward reduction from HAMILTON PATH, the former two are indeed NP-complete.

Let  $G = (V, E)$  be a graph. If for each two distinct vertices  $s, t \in V$  there exists a Hamilton path with end-vertices  $s$  and  $t$ , then  $G$  is *Hamilton-connected*. If  $G - S$  is Hamilton-connected for every set  $S \subset V$  with  $|S| \leq k$  for some integer  $k \geq 0$ , then  $G$  is  *$k$ -Hamilton-connected*. Note that a graph is Hamilton-connected if and only if it is 0-Hamilton-connected. The HAMILTON CONNECTIVITY problem is that of computing the maximum value of  $k$  for which a given graph is  $k$ -Hamilton-connected. Dean [19] showed that already deciding whether  $k = 0$  is NP-complete. Kužel, Ryjáček and Vrána [31] proved this for  $k = 1$ . A straightforward generalization of the latter result yields the same for any integer  $k \geq 1$ . As an aside, the HAMILTON CONNECTIVITY problem has recently been studied by Kužel, Ryjáček and Vrána [31], who showed that NP-completeness of the case  $k = 1$  for line graphs would disprove the conjecture of Thomassen that every 4-connected line graph is hamiltonian, unless  $P = NP$ .

A *path cover* of a graph  $G$  is a set of mutually vertex-disjoint paths  $P_1, \dots, P_k$  with  $V(P_1) \cup \dots \cup V(P_k) = V(G)$ . The size of a smallest path cover is denoted by  $\pi(G)$ . The PATH COVER problem is to compute this number, whereas the 1-PATH COVER problem is to compute the size of a smallest path cover that contains a path in which some given vertex  $u$  is an end-vertex. Because a Hamilton path of a graph is a path cover of size 1, PATH COVER and 1-PATH COVER are NP-hard via a reduction from HAMILTON PATH and 1-HAMILTON PATH, respectively.

We denote the number of connected components of a graph  $G = (V, E)$  by  $c(G)$ . A subset  $S \subset V$  is a *vertex cut* of  $G$  if  $c(G - S) \geq 2$ , and  $G$  is called  *$k$ -connected* if the size of a smallest vertex cut of  $G$  is at least  $k$ . We say that  $G$  is  *$t$ -tough* if  $|S| \geq t \cdot c(G - S)$  for every vertex

cut  $S$  of  $G$ . The *toughness*  $\tau(G)$  of a graph  $G = (V, E)$  was defined by Chvátal [16] as

$$\tau(G) = \min \left\{ \frac{|S|}{c(G-S)} : S \subset V \text{ and } c(G-S) \geq 2 \right\},$$

where  $\tau(G) = \infty$  if  $G$  is a complete graph. Note that  $\tau(G) \geq 1$  if  $G$  is hamiltonian; the reverse statement does not hold in general (see [8]). The TOUGHNESS problem is to compute  $\tau(G)$  for a graph  $G$ . Bauer, Hakimi and Schmeichel [4] showed that already deciding whether  $\tau(G) = 1$  is coNP-complete.

The *scattering number* of a graph  $G = (V, E)$  was defined by Jung [28] as

$$\text{sc}(G) = \max\{c(G-S) - |S| : S \subset V \text{ and } c(G-S) \geq 2\},$$

where  $\text{sc}(G) = -\infty$  if  $G$  is a complete graph. We call a set  $S$  on which  $\text{sc}(G)$  is attained a *scattering set*. Note that  $\text{sc}(G) \leq 0$  if  $G$  is hamiltonian. Shih, Chern and Hsu [38] showed that  $\text{sc}(G) \leq \pi(G)$  for all graphs  $G$ . Hence,  $\text{sc}(G) \leq 1$  if  $G$  is traceable. The SCATTERING NUMBER problem is to compute  $\text{sc}(G)$  for a graph  $G$ . The observation that  $\text{sc}(G) = 0$  if and only if  $\tau(G) = 1$  combined with the aforementioned result of Bauer, Hakimi and Schmeichel [4] implies that already deciding whether  $\text{sc}(G) = 0$  is coNP-complete.

## 1.2 Known Results for Interval Graphs

We first briefly discuss the results on testing hamiltonicity properties for proper interval graphs. Besides giving a linear-time algorithm for solving HAMILTON CYCLE on proper interval graphs, Bertossi [5] also showed that a proper interval graph is traceable if and only if it is connected. His work was extended by Chen, Chang and Chang [13] who showed that a proper interval graph is hamiltonian if and only if it is 2-connected, and that a proper interval graph is Hamilton-connected if and only if it is 3-connected. In addition, Chen and Chang [12] showed that a proper interval graph has scattering number at most  $2 - k$  if and only if it is  $k$ -connected.

Below we survey the results on testing hamiltonicity properties for interval graphs that appeared after Keil [29] solved the HAMILTON CYCLE problem on interval graphs.

*Testing for Hamilton cycles and Hamilton paths.* The  $O(n + m)$  time algorithm of Keil [29] makes use of an interval representation. One can find such a representation by executing the  $O(n + m)$  time interval recognition algorithm of Booth and Lueker [9]. If an interval representation is already given, Manacher, Mankus and Smith [36] showed that HAMILTON CYCLE and HAMILTON PATH can be solved in  $O(n \log n)$  time. In the same paper, they ask whether the time bound for these two problems can be improved to  $O(n)$  time if a so-called sorted interval representation is given. Chang, Peng and Liaw [11] answered this question in the affirmative. They showed that this even holds for PATH COVER.

*When no Hamilton path exists.* In this case, LONGEST PATH and PATH COVER are natural problems to consider. Ioannidou, Mertziou and Nikolopoulos [26] gave an  $O(n^4)$  algorithm for solving LONGEST PATH on interval graphs. Arikati and Pandu Rangan [1] and also Damaschke [18] showed that PATH COVER can be solved in  $O(n + m)$  time on interval graphs.

Damaschke [18] posed the complexity status of 1-HAMILTON PATH and 2-HAMILTON PATH on interval graphs as open questions. The latter question is still open, but Asdre and Nikolopoulos [3] answered the former question by presenting an  $O(n^3)$  time algorithm that solves 1-PATH COVER, and hence 1-HAMILTON PATH, on interval graphs. Li and Wu [32]

announced an  $O(n + m)$  time algorithm for 1-PATH COVER on interval graphs. Deogun, Kratsch and Steiner [20] showed that for all  $k \geq 1$  any cocomparability graph has a path cover of size at most  $k$  if and only if its scattering number is at most  $k$ .<sup>7</sup> Because every interval graph is cocomparability (see e.g. [10]), this result holds for interval graphs as well. Deogun, Kratsch and Steiner also proved that a cocomparability graph  $G$  is hamiltonian if and only if  $\text{sc}(G) \leq 0$ . Recall that the latter condition is equivalent to  $\tau(G) \geq 1$ . As such, this result restricted to interval graphs is known (see e.g. [15]) to be implicit already in Keil's algorithm [29]. Hung and Chang [25] gave an  $O(n + m)$  time algorithm that finds a scattering set of an interval graph  $G$  if  $\text{sc}(G) \geq 0$ .

### 1.3 Our Results

*When a Hamilton path does exist.* In this case, HAMILTON CONNECTIVITY is a natural problem to consider. Isaak [27] used a closely related variant of toughness called  $k$ -path toughness to characterize interval graphs that contain the  $k$ th power of a Hamiltonian path. However, the aforementioned results of Deogun, Kratsch and Steiner [20] suggest that trying to characterize  $k$ -Hamilton-connectivity in terms of the scattering number of an interval graph may be more appropriate than doing this in terms of its toughness. We confirm this by showing that for all  $k \geq 0$  an interval graph is  $k$ -Hamilton-connected if and only if its scattering number is at most  $-(k + 1)$ . Together with the results of Deogun, Kratsch and Steiner [20], this leads to the following theorem.

**Theorem 1.** *Let  $G$  be an interval graph. Then  $\text{sc}(G) \leq k$  if and only if*

- (i)  *$G$  has a path cover of size at most  $k$  when  $k \geq 1$*
- (ii)  *$G$  has a Hamilton cycle when  $k = 0$*
- (iii)  *$G$  is  $-(k + 1)$ -Hamilton-connected when  $k \leq -1$ .*

Moreover, we give an  $O(n + m)$  time algorithm for solving SCATTERING NUMBER on interval graphs that also produces a scattering set. This improves the  $O(n^3)$  time bound of a previous algorithm due to Kratsch, Kloks and Müller [30]. Combining our result with Theorem 1 yields that HAMILTON CONNECTIVITY can be solved in  $O(n + m)$  time on interval graphs.

For proper interval graphs we can express  $k$ -Hamilton-connectivity also in the following way. Recall that a proper interval graph has scattering number at most  $2 - k$  if and only if it is  $k$ -connected [12]. Combining this result with Theorem 1 yields that for all  $k \geq 0$ , a proper interval graph is  $k$ -Hamilton-connected if and only if it is  $(k + 3)$ -connected.

### 1.4 Our Proof Method

In order to explain our approach we first need to introduce some additional terminology. A set of  $p$  internally vertex-disjoint paths  $P_1, \dots, P_p$ , all of which have the same end-vertices  $u$  and  $v$  of a graph  $G$ , is called a *stave* or  $p$ -*stave* of  $G$ , which is *spanning* if  $V(P_1) \cup \dots \cup V(P_p) = V(G)$ . A spanning  $p$ -stave between two vertices  $u$  and  $v$  is also called a spanning  $(p; u, v)$ -path-system [14], a  $p^*$ -container between  $u$  and  $v$  [24,34] or a spanning  $p$ -trail [32]. We call a spanning  $p$ -stave between two vertices  $u$  and  $v$  of a graph an *optimal spanning stave* between  $u$  and  $v$  if there does not exist a spanning  $(p + 1)$ -stave between  $u$  and  $v$ .

<sup>7</sup> This has also been shown by Lehel in an unpublished manuscript [35].

By Menger's Theorem (Theorem 9.1 in [8]), a graph  $G$  is  $p$ -connected if and only if there exists a  $p$ -stave between any pair of vertices of  $G$ . It is also well-known that the existence of a  $p$ -stave between two given vertices can be decided in polynomial time (cf. [8]). However, given an integer  $p \geq 1$  and two vertices  $u$  and  $v$  of a general input graph  $G$ , deciding whether there exists a spanning  $p$ -stave between  $u$  and  $v$  is an NP-complete problem: for  $p = 1$  the problem is equivalent to the NP-complete problem 2-HAMILTON PATH; for  $p = 2$  the problem is equivalent to the NP-complete problem HAMILTON CYCLE; for  $p \geq 3$  NP-completeness follows by induction and by considering the graph obtained after adding one vertex adjacent to  $u$  and  $v$ .

Damaschke's algorithm [18] for solving PATH COVER on interval graphs, which is based on the approach of Keil [29], actually solves the following problem in  $O(n + m)$  time: given an interval graph  $G$  and an integer  $p$ , does  $G$  have a spanning  $p$ -stave between the vertex  $u_1$  corresponding to the leftmost interval of an interval model of  $G$  and the vertex  $u_n$  corresponding to the rightmost one? Here, the *leftmost* interval is the interval with the smallest right end-point, and the *rightmost* interval is the interval with the largest left end-point. Note that we may assume without loss of generality that these intervals are unique.

In Section 2 we present our  $O(n + m)$  time algorithm that solves SCATTERING NUMBER on interval graphs. Our approach is as follows. We extend Damaschke's algorithm to an  $O(n + m)$  time algorithm that takes as input only an interval graph  $G$  and finds an optimal spanning stave of  $G$  between  $u_1$  and  $u_n$ , unless it detects that there does not exist a spanning stave between  $u_1$  and  $u_n$ . In the latter case  $G$  is not hamiltonian, and hence,  $\text{sc}(G) \geq 1$  as shown by Deogun, Kratsch and Steiner [20]. Therefore, the  $O(n + m)$  time algorithm of Hung and Chang [25] for computing a scattering set may be applied. In the case that there is an optimal spanning stave between  $u_1$  and  $u_n$ , we show how this enables us to compute a scattering set  $S$  of  $G$  in  $O(n + m)$  time. In fact we show a stronger relationship, namely that  $|S| \geq 2 - p^*$  if  $p^*$  is the size of an optimal spanning stave between  $u_1$  and  $u_n$ . We consequently obtain a structural result, proved also in Section 2, which states that  $G$  contains a spanning  $p$ -stave between  $u_1$  and  $u_n$  if and only if  $\text{sc}(G) \leq 2 - p$ .

In Section 3 we prove our contribution to Theorem 1 (iii), that is, the case when  $k \leq -1$ . In particular, for proving the subcase  $k = -1$ , we show that an interval graph  $G$  is Hamilton-connected if it contains a spanning 3-stave between the vertex corresponding to the leftmost interval of an interval model and the vertex corresponding to the rightmost one. We then combine this claim with the structural result obtained in Section 2.

## 2 Spanning Staves and the Scattering Number

In order to present our algorithm we start by giving the necessary terminology and notations.

**Additional Terminology.** A set  $D \subseteq V$  *dominates* a graph  $G = (V, E)$  if each vertex of  $G$  belongs to  $D$  or has a neighbor in  $D$ . We will usually denote a path in a graph by its sequence of distinct vertices such that consecutive vertices are adjacent. If  $P = u_1 \dots u_n$  is a path, then we denote its *reverse* by  $P^{-1} = u_n \dots u_1$ . We may *concatenate* two paths  $P$  and  $P'$  whenever they are vertex-disjoint except for the last vertex of  $P$  coinciding with the first vertex of  $P'$ . The resulting path is then denoted by  $P \circ P'$ .

A *clique path* of an interval graph  $G$  with vertices  $u_1, \dots, u_n$  is a sequence  $C_1, \dots, C_s$  of all maximal cliques of  $G$ , such that each edge of  $G$  is present in some clique  $C_i$  and each vertex of  $G$  appears in consecutive cliques only. It is well known that a graph is interval if and only

if it has a clique path. We use the  $O(n + m)$  time recognition algorithm of interval graphs due to Booth and Lueker [9], which produces a clique path  $C_1, \dots, C_s$  when the input graph is interval. Hence, in the remainder of this section, we let  $G$  denote an interval graph with a clique path  $C_1, \dots, C_s$ . The latter yields a specific interval model for  $G$  that we will also use throughout the remainder of this paper: a vertex  $u_i$  of  $G$  is represented by the interval  $I_{u_i} = [\ell_i, r_i]$ , where  $\ell_i = \min\{j : u_i \in C_j\}$  and  $r_i = \max\{j : u_i \in C_j\}$ , which are referred to as the *start point* and the *end point* of  $u_i$ , respectively. By definition,  $C_1$  and  $C_s$  are maximal cliques. Hence both  $C_1$  and  $C_s$  contain at least one vertex that does not occur in any other clique. We assume that  $u_1$  is such a vertex in  $C_1$ , and hence it corresponds to the leftmost interval defined earlier. Analogously,  $u_n$ , corresponding to the rightmost one, is such a vertex in  $C_s$ . Note that  $I_{u_1} = [1, 1]$  and  $I_{u_n} = [s, s]$  are single points.

Keil [29] made the useful observation that any Hamilton path in an interval graph can be reordered into a path with a special property that allows to build a greedy-like algorithm. A path in an interval graph is *monotone* if every edge  $(u, v)$  can be assigned a point from  $I_u \cap I_v$  such that these points ordered in the appearance of the edges in the path form a nondecreasing sequence (it is called “straight” in previous works [29, 18]).

**Lemma 1 ([29]).** *If the interval graph  $G$  contains a Hamilton path, then it contains a monotone Hamilton path from  $u_1$  to  $u_n$ .*

We use Lemma 1 to rearrange certain path systems in  $G$  into a single path as follows. Let  $P$  be a path between  $u_1$  and  $u_n$  and let  $\mathcal{Q} = (Q_1, \dots, Q_k)$  be a collection of paths, each of which contains  $u_1$  or  $u_n$  as an end-vertex. Furthermore,  $P$  and all the paths of  $\mathcal{Q}$  are assumed to be vertex-disjoint except for possible intersections at  $u_1$  or  $u_n$ . Consider the path  $Q_1$ . By symmetry, it may be assumed to contain  $u_1$ . We apply Lemma 1 to the subgraph induced by the path  $P \circ (Q_1 - u_n)$  and obtain a path  $P'$  between  $u_1$  and  $u_n$  containing all the vertices of  $P \cup Q_1$ . Proceeding in a similar way for the paths  $Q_2, \dots, Q_k$ , we obtain a path between  $u_1$  and  $u_n$  on the same vertex set as  $P \cup \bigcup_{j=1}^k Q_j$ . We denote the resulting path by  $\text{merge}(P, Q_1, \dots, Q_k)$  or simply by  $\text{merge}(P, \mathcal{Q})$ .

Algorithm 1 is our  $O(n + m)$  time algorithm for finding an optimal spanning stave between  $u_1$  and  $u_n$  if it exists. Similarly to the algorithm of Damaschke [18], it gradually builds up a set  $\mathcal{P}$  of internally disjoint monotone paths starting at  $u_1$  and passing through vertices of  $C_t \setminus C_{t+1}$  before moving to  $C_t \cap C_{t+1}$  for  $t = 1, \dots, s - 1$ . In contrast to Damaschke’s algorithm, where the number of paths is fixed, our algorithm starts with the maximum possible number of paths, i.e. the degree of the leftmost vertex  $u_1$ . Intuitively, if some path cannot be further extended, it is abandoned and the set of possible paths is reduced. At the final phase we clear all abandoned paths by merging them with any path that reaches  $u_n$ .

It is convenient to consider all these paths ordered from  $u_1$  to their (temporary) end-vertices that we call *terminals*, and to use the terms *predecessor*, *successor*, and *descendant* of a fixed vertex  $v$  in one of the paths with the usual meaning of a vertex immediately before, immediately after, and somewhere after  $v$  in one of these paths, respectively. For a path  $P$  with end-vertex  $u$  and a vertex  $v \notin V(P)$ , we say that  $P$  has been *extended by attaching*  $v$ , if  $uv$  becomes the last edge of the resulting path, while all edges of  $P$  are preserved in the resulting path too. By extending a path by attaching a set of vertices we mean attaching vertices of the set one by one, in an arbitrary order.

We note that the path system  $\mathcal{P}$  provided by Algorithm 1 is a valid stave. A routine check confirms that the following loop invariant holds at line 6: *the last vertices of paths from  $\mathcal{P}$*

**Input:** A clique-path  $C_1, \dots, C_s$  in an interval graph  $G$ .

**Output:** An optimal spanning stave  $\mathcal{P}$  between  $u_1$  and  $u_n$ , if it exists.

```

1 begin
2   let  $p = \deg(u_1)$ ;
3   let  $R_i = u_1$  for all  $i = 1, \dots, p$ ;
4   let  $\mathcal{P} = \{R_1, \dots, R_p\}$ ;
5   let  $\mathcal{Q} = \emptyset$ ;
6   for  $t := 1$  to  $s - 1$  do
7     choose a  $P \in \mathcal{P}$  whose terminal has the smallest end point among all terminals;
8     if  $C_t \setminus (C_{t+1} \cup \bigcup(\mathcal{P} \cup \mathcal{Q})) \neq \emptyset$  then extend  $P$  by attaching vertices of  $C_t \setminus (C_{t+1} \cup \bigcup(\mathcal{P} \cup \mathcal{Q}))$  ;
9     for every path  $R \in \mathcal{P}$  do
10      if the terminal of  $R$  is not in  $C_{t+1}$  then
11        try to extend  $R$  by attaching a new vertex  $u$  from  $(C_t \cap C_{t+1}) \setminus \bigcup(\mathcal{P} \cup \mathcal{Q})$  with the
12        smallest end point;
13        if such  $u$  does not exist then
14          remove  $R$  from  $\mathcal{P}$ ;
15          insert  $R$  into  $\mathcal{Q}$ ;
16          decrement  $p$ ;
17          if  $p = 0$  then report that  $G$  has no spanning 1-stave between  $u_1$  and  $u_n$  and quit
18        end
19      end
20    end
21    choose any  $P \in \mathcal{P}$ ;
22    extend  $P$  by attaching vertices of  $C_s \setminus \bigcup(\mathcal{P} \cup \mathcal{Q})$  ;
23    let  $P = \text{merge}(P, \mathcal{Q})$ ;
24    for every path  $R \in \mathcal{P} \setminus P$  do extend  $R$  by attaching  $u_n$ ;
25    report the optimal spanning  $p$ -stave  $\mathcal{P}$ .
26 end

```

**Algorithm 1:** Finding an optimal spanning stave.

all belong to the clique  $C_t$ . This is guaranteed by the computations at lines 10–18. At line 20 it also holds that all vertices of  $C_t \setminus C_{t+1}$  appear in the current  $\mathcal{P} \cup \mathcal{Q}$ , as they have been included at line 8. When the loop terminates, the remaining vertices are incorporated at line 22. Thus the resulting path system  $\mathcal{P}$  is a spanning stave.

In Theorem 2 we show that no spanning stave may consist of more than  $2 - \text{sc}(G)$  paths. On the other hand, we will also show that the  $p$ -stave found by Algorithm 1 can be supplied with a scattering set witnessing that  $p \geq 2 - \text{sc}(G)$ . In other words this is an optimal scattering set whose existence also proves the optimality of the spanning stave. For this goal, we first develop some auxiliary terminology related to our algorithm.

If vertex  $u_i$  has been processed by the algorithm and attached to a path at lines 8 or 11 of Algorithm 1, we say that  $u_i$  has been *activated* at time  $a_i$ , and we assign  $a_i$  the current value of the variable  $t$ . Thus, we think of time steps  $t = 1, \dots, t = s$  during the execution of the algorithm. When at the same or at a later stage a vertex  $u_j$  has been attached as a successor of  $u_i$  to a path, we say that  $u_i$  has been *deactivated* at time  $d_i$ , and assign  $d_i = a_j$ . Hence, as soon as  $a_i$  and  $d_i$  have been assigned values, we have  $\ell_i \leq a_i \leq d_i \leq r_i$ . Furthermore, any of the implied inequalities holds whenever both of its sides are defined. Note that any of these inequalities may be an equality; in particular, a vertex can be activated and deactivated at the same time.

If the involved parameters have been assigned values, we consider the open (time) intervals  $(\ell_i, a_i)$ ,  $(a_i, d_i)$  and  $(d_i, r_i)$ , and we say that  $u_i$  is *free* during  $(\ell_i, a_i)$  if this interval is nonempty,

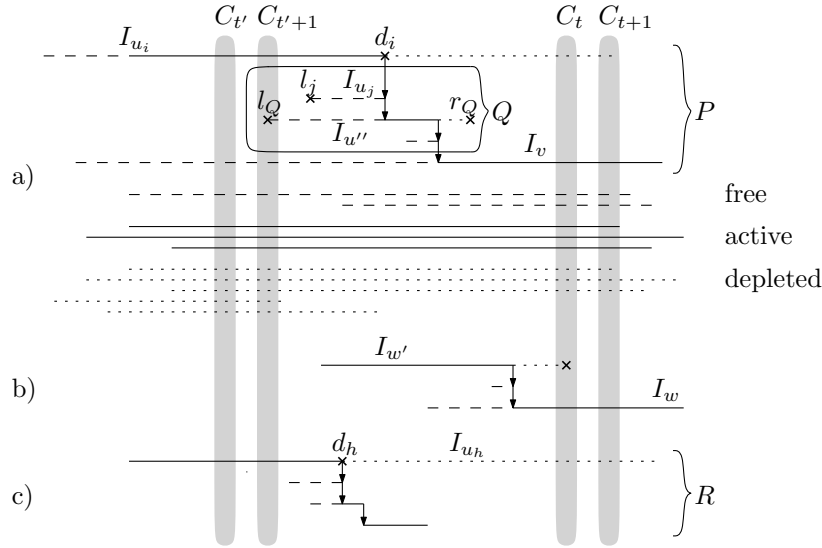
active during  $(a_i, d_i)$  if this interval is nonempty, and *depleted* during  $(d_i, r_i)$  if this interval is nonempty. In particular, note that the vertices that are attached to a path at line 8 (if any) are from  $C_t \setminus C_{t+1}$ , so they satisfy  $r_i = t$  and  $a_i = t$ . Such vertices will not be active or depleted during any (nonempty) time interval, but they are free during the time interval  $(\ell_i, r_i)$  if this interval is nonempty.

For  $1 \leq j \leq k \leq s$ , we define  $C_{j,k} = \bigcup_{i=j}^k C_i$ .

The following lemma is crucial.

**Lemma 2.** Suppose that Algorithm 1 terminates at line 16 or finishes an iteration of the loop at lines 6–20. Let the current value of the variable  $t$  be also denoted by  $t$ . If there is at least one depleted vertex during the interval  $(t, t+1)$ , then there exists an integer  $t' < t$  with the following properties (see Fig. 1a for an illustration):

- (i)  $C_{t'+1,t} \setminus (C_{t'} \cup C_{t+1}) \neq \emptyset$ ,
- (ii) a unique vertex  $u_i \in C_{t'} \cap C_{t+1}$  is active during  $(t', t'+1)$  and is depleted during  $(t, t+1)$ ,
- (iii) all vertices that are active during  $(t, t+1)$  are also active during  $(t', t'+1)$ , with the only possible exception of the last descendant of  $u_i$  (which we denote by  $v$ ) that can be free during  $(t', t'+1)$ ,
- (iv) all vertices that are depleted during  $(t, t+1)$  and distinct from  $u_i$  are also depleted during  $(t', t'+1)$ ,
- (v) all vertices that are active during  $(t', t'+1)$  are also active during  $(t, t+1)$ , with the only exception of  $u_i$ , and
- (vi) all vertices that are free during  $(t', t'+1)$  are also free during  $(t, t+1)$ , with the only possible exception of  $v$  if it is active during  $(t, t+1)$ .



**Fig. 1.** A path system as described in Lemma 2. The vertical arrows indicate successors in the paths and the time of activation and deactivation.



*Proof.* Assume that there is at least one depleted vertex during the interval  $(t, t+1)$ , and let  $u_i$  be a vertex with the latest deactivation time among those that are depleted during  $(t, t+1)$ . To prove that this vertex is unique, we note that all but at most one of the vertices deactivated during a given iteration of the loop on lines 6–20 (say, at time  $t$ ) have end point equal to  $t$  and hence cannot be depleted during a nonempty interval. The only possible exception is the terminal of the path  $P$  chosen at line 7 (and only if it is deactivated due to attaching a vertex to  $P$  at line 8).

We define  $Q$  to be the subpath of  $P$  formed by all descendants of  $u_i$ , except that if the last descendant  $v$  of  $u_i$  is active during  $(t, t+1)$ , we do not include  $v$  in  $Q$ . Observe that the successor of  $u_i$  has the same deactivation time as  $u_i$ , hence it is distinct from  $v$ , and therefore  $Q$  is nonempty. Let  $\ell_Q$  be the smallest start point among intervals corresponding to vertices of  $Q$ , and let  $r_Q$  be the largest such end point.

If  $P$  has a vertex that is active during  $(t, t+1)$ , this vertex is  $v$  and it is not a vertex of  $Q$ . Thus all vertices of  $Q$  are either depleted during  $(t, t+1)$  or their end point is less than or equal to  $t$ . By the choice of  $u_i$ , none of them belongs to  $C_{t+1}$ , and hence  $r_Q \leq t$ . We choose  $t' = \ell_Q - 1$ . Notice that for  $u_j \in V(Q)$ ,  $r_j \geq d_i$ . Thus if we let  $u_q$  be the vertex of  $u$  such that  $\ell_q = \ell_Q$ , then  $u_q$  is free during  $(t' + 1, d_i)$ .

Observe that all vertices of  $Q$  are in  $C_{t'+1, t} \setminus (C_{t'} \cup C_{t+1})$ . Hence, this set is not empty and property (i) is proved.

We prove (ii). Since the deactivation of  $u_i$  happened when its successor  $u_j$  was free, we have  $d_i \geq \ell_j > t'$ . Hence,  $u_i$  cannot be depleted during  $(t', t' + 1)$ . Observe that  $u_i \neq u_1$ , as  $u_1$  is not depleted during  $(t-1, t)$ . Therefore,  $u_i$  has a predecessor. Denote it by  $u'$ . If  $u'$  were adjacent to the vertex  $u_q$  of  $Q$ , then the algorithm would choose  $u_q$  as the successor of  $u'$ , since  $r_i > r_Q \geq r_q$ . Consequently, the start point of  $u'$  is less than or equal to  $t'$ , so  $u_i$  is active during  $(t', t' + 1)$ . The uniqueness of  $u_i$  will follow easily once we establish property (iv).

To show property (iii), assume that  $u_m$  is a vertex different from  $v$  that is active during  $(t, t+1)$  but has been activated after  $t'$ . Since  $u_1$  is not active during  $(t, t+1)$ ,  $u_m \neq u_1$  and  $u_m$  has a predecessor  $u'$ . We first suppose that  $u_m$  is active during  $(d_i - 1, d_i)$ . The vertex  $u'$  is deactivated at some time  $t''$  such that  $t' + 1 \leq t'' \leq d_i - 1$ . Hence, it is adjacent to the previously defined vertex  $u_q$  of  $Q$  that is free during  $(t' + 1, d_i)$ . Since  $r_q \leq r_Q < t + 1 \leq r_m$ , the successor of  $u'$  should be  $u_q$  rather than  $u_m$ , a contradiction.

It follows that  $u_m$  is not active during  $(d_i - 1, d_i)$ . The vertex  $u_m$  is included in some path  $R \in \mathcal{P}$ ,  $R \neq P$ . This path contains a vertex  $w'$  that is active during  $(d_i - 1, d_i)$  (see Fig. 1b), where  $u_m$  is a descendant of  $w'$ . Observe that  $w'$  is not active during  $(t, t+1)$  because  $u_m$  is. Suppose that the end point of  $w'$  is at least  $t+1$ . Then  $w'$  is depleted during  $(t, t+1)$ , so by the choice of  $u_i$ ,  $w'$  is deactivated before time  $d_i$  and cannot be active during  $(d_i - 1, d_i)$ , a contradiction.

Thus, the end point of  $w'$  is not larger than  $t$ . But then  $w'$  should have been chosen at line 7 of the algorithm instead of  $u_i$ .

For (iv), assume that some  $u_h \neq u_i$  is depleted during  $(t, t+1)$ , but  $d_h \geq t' + 1$ . By the choice of  $u_i$ , we have  $d_h < d_i$ . Without loss of generality, assume that  $u_h$  was chosen such that  $d_h$  is maximal. Let  $R$  be the path in  $\mathcal{P} \cup \mathcal{Q}$  containing  $u_h$ . Note that  $R \neq P$ . If  $R$  contains a vertex  $w$  that is active during  $(t, t+1)$ , then by (iii),  $w$  is active during  $(t', t' + 1)$  and we conclude that  $u_h$  cannot be included in  $R$ ; a contradiction.

It follows that no vertex of  $R$  is active during  $(t, t+1)$  (see Fig. 1c). Moreover, by the choice of  $u_h$ , the end points of all its descendants are less than or equal to  $t$ , because if there is a descendant  $u_j$  of  $u_h$  with  $r_j \geq t+1$ , then  $w$  is depleted during  $(t, t+1)$  and  $d_j > d_h$ ,

a contradiction. Recall that the vertex  $u_q$  is free during  $(t' + 1, d_i)$ . Since the path  $R$  cannot be terminated while a free vertex is available, it must contain a vertex that is active during  $(d_i - 1, d_i)$ . However, this vertex has a smaller end point than  $u_i$ , contradicting the correct execution of the algorithm at line 7.

To obtain (v), assume that  $w \neq u_i$  is active during  $(t', t' + 1)$  but not active during  $(t, t + 1)$ . The vertex  $w$  is included in some path  $R \in \mathcal{P} \cup \mathcal{Q}$ ,  $R \neq P$ . If one of the descendants of  $w$  is active during  $(t, t + 1)$ , then by (iii), this vertex is active during  $(t', t' + 1)$  contradicting the activeness of  $w$  at the same time. Similarly, if  $w$  or one of its descendants is depleted during  $(t, t + 1)$ , then by (iv), this vertex is depleted during  $(t', t' + 1)$  and  $w$  cannot be active. It follows that the end points of  $w$  and its descendants are less than or equal to  $t$ . If  $d_i = t' + 1$ , then  $R$  has a vertex that is active during  $(d_i - 1, d_i)$ . If  $d_i > t' + 1$ , then we use the observation that the vertex  $u_q$  is free during  $(t' + 1, d_i)$ , and again conclude that  $R$  has an active vertex during  $(d_i - 1, d_i)$ . Then this vertex should be selected by the algorithm in line 7 instead of  $u_i$ ; a contradiction.

It remains to prove (vi). Let  $w$  be a vertex that is free during  $(t', t' + 1)$  and not free during  $(t, t + 1)$ . Moreover, we assume that  $w \neq v$  if  $v$  is active during  $(t, t + 1)$ . Our algorithm does not terminate until time  $t$ . Therefore,  $w$  is included in some path  $R \in \mathcal{P} \cup \mathcal{Q}$ ,  $R \neq P$ . This path has a vertex that is active during  $(t', t' + 1)$ . By (v), this vertex remains active until  $t + 1$ , but it means that  $w$  is not included in  $R$ .  $\square$

Now we are ready to state and prove the main structural result.

**Theorem 2.** *A non-complete interval graph  $G$  contains a spanning  $p$ -stave between  $u_1$  and  $u_n$  if and only if  $\text{sc}(G) \leq 2 - p$ .*

*Proof.* Let us first assume that  $\mathcal{P} = (R_1 \dots, R_p)$  is a spanning  $p$ -stave between  $u_1$  and  $u_n$ . If  $G$  is complete, then the claim is trivial. Otherwise, let  $S \subset V(G)$  be a scattering set of  $G$ . We claim that  $u_1, u_n \notin S$ . Suppose the contrary. Since  $u_1$  is simplicial, i.e., its neighborhood induces a clique, we get that  $c(G - S) \leq c(G - (S - \{u_1\}))$  and therefore  $c(G - S) - |S| < c(G - (S - \{u_1\})) - |S - \{u_1\}|$ , a contradiction with the choice of  $S$ . The argument for  $u_n$  is symmetric.

Since each path in  $\mathcal{P}$  connects  $u_1$  and  $u_n$ , the union of intervals corresponding to the internal vertices of such a path is the interval  $[1, s]$ . In other words, the internal vertices of each path in  $\mathcal{P}$  dominate  $G$ . Hence the set  $S$ , which is a vertex cut by definition, contains an internal vertex from each path of  $\mathcal{P}$ . From each path  $R_i$  of  $\mathcal{P}$ , we choose a vertex  $s_i \in S$ . We let  $S' = \{s_1, \dots, s_p\}$ .

Consider the spanning subgraph  $G'$  of  $G$  induced by the edges of  $\mathcal{P}$ . Observe that  $G' - S'$  has two components. If we remove the remaining vertices of  $S \setminus S'$  one by one, then with each vertex we remove, the number of components of the remaining graph can increase by at most one, as  $u_1, u_n \notin S$ . Hence  $c(G - S) \leq c(G' - S) \leq 2 + |S| - p$ , which means that  $\text{sc}(G) \leq 2 - p$ , as  $S$  is a scattering set of  $G$ . This proves the forward implication of the statement.

For the other direction, let us assume that  $G$  does not have a spanning  $p$ -stave between  $u_1$  and  $u_n$ . If  $\deg(u_1) < p$ , then let  $S$  be the set of neighbors of  $u_1$ . Because  $G$  is not a complete graph,  $u_n \notin S$ , i.e.,  $S$  is a vertex cut and  $c(G - S) \geq 2$ . Then  $\text{sc}(G) \geq c(G - S) - |S| \geq 2 - |S| > 2 - p$ . Otherwise, if  $\deg(u_1) \geq p$ , then during the execution of Algorithm 1, at some stage the value set at line 15 becomes smaller than  $p$ . Suppose  $t_1$  is the value of the variable  $t$  at this moment. We will complete the proof by constructing a scattering set  $S$ , for which we show that  $c(G - S) - |S| > 2 - p$ .

We repeatedly use Lemma 2 and find a finite sequence  $t_1, t_2, \dots, t_k$ , such that  $t_{i+1} = (t_i)'$  as long as there are depleted vertices during  $(t_i, t_i + 1)$  for  $i < k$ . Note that this sequence is decreasing, as we analyze the execution of the algorithm backwards, that is, we first analyze what happened before  $t_1$ , then what happened before  $t_2$ , and so on. As the sequence is limited to positive integers, this process stops at some moment. In particular, we have no depleted vertices during  $(t_k, t_k + 1)$ . In the utmost case this might be the time interval  $(1, 2)$ , where no vertex is depleted.

We choose  $S = \bigcup_{i=1}^k (C_{t_i} \cap C_{t_{i+1}})$  and prove that  $G - S$  has at least  $|S| - p + 3$  components.

The subgraphs  $G[C_{1,t_k}] - S$  and  $G[C_{t_1+1,s}] - S$  contain  $u_1$  and  $u_n$ , respectively; in particular, they have at least one component each. By property (i) in Lemma 2,  $G[C_{t_{i+1}+1,t_i}] - S$  has at least one component for each  $i \in \{1, \dots, k-1\}$ . Since all these components are distinct components of  $G - S$ , the graph  $G - S$  has at least  $k + 1$  components.

By properties (ii), (v) and (vi) in Lemma 2,  $(C_{t_{i+1}} \cap C_{t_{i+1}+1}) \setminus (C_{t_i} \cap C_{t_{i+1}})$  contains only vertices that are depleted during  $(t_{i+1}, t_{i+1} + 1)$  for each  $i \in \{1, \dots, k-1\}$ . Further,  $C_{t_1} \cap C_{t_1+1}$  has no vertices that are free during  $(t, t + 1)$ , because at least one path is not extendable at time  $t_1$ . Also this set has at most  $p - 1$  vertices that are active during  $(t, t + 1)$ . Hence, the remaining vertices are depleted. By properties (ii) and (iv) in Lemma 2, for each  $i \in \{1, \dots, k-1\}$ , exactly one vertex that is depleted during  $(t_i, t_{i+1})$  has a different status during  $(t_{i+1}, t_{i+1} + 1)$  and is active. It follows that  $|S| \leq (p - 1) + (k - 1) = k + p - 2$  as required.  $\square$

We will now discuss how to compute a scattering set of  $G$  in  $O(n + m)$  time. We apply Algorithm 1. This takes  $O(n + m)$  time; the only operation whose time complexity has not been discussed is  $\text{merge}(P, Q)$  at line 23, and we refer to Damaschke's proof of Lemma 1 to verify that this line can be implemented in  $O(n + m)$  time. If Algorithm 1 outputs that there is no spanning stave between  $u_1$  and  $u_n$ , then  $G$  is not hamiltonian. Recall that in that case  $\text{sc}(G) \geq 1$  [20] and that we then may apply the  $O(n + m)$  time algorithm of Hung and Chang [25] for computing a scattering set. Otherwise, Algorithm 1 finds an optimal spanning stave. Our proof of Theorem 2 provides a construction of a scattering set of  $G$  that can be straightforwardly implemented in  $O(n + m)$  time.

**Corollary 1.** *A scattering set of an interval graph can be computed in  $O(n + m)$  time.*

### 3 Hamilton-connectivity

In this section we prove our contribution to Theorem 1, which is the following.

**Theorem 3.** *For all  $k \geq 0$ , an interval graph  $G$  is  $k$ -Hamilton-connected if and only if  $\text{sc}(G) \leq -(k + 1)$ .*

*Proof.* Let  $k \geq 0$  and  $G$  be an interval graph with leftmost and rightmost vertices  $u_1$  and  $u_n$  as defined before. The statement of Theorem 3 is readily seen to hold when  $G$  is a complete graph. Hence we may assume without loss of generality that  $G$  is not complete.

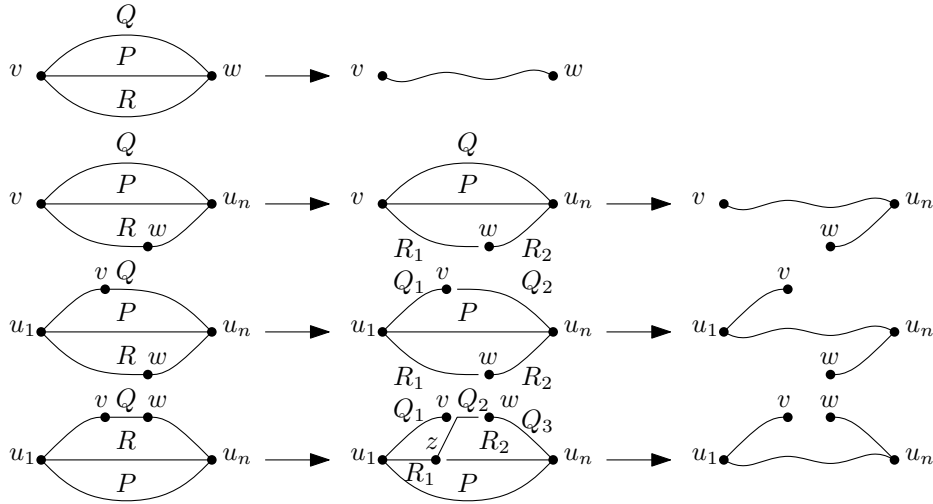
First suppose that  $G$  is  $k$ -Hamilton-connected. Then  $G$  has at least  $k + 3$  vertices. We claim that  $G - R$  is traceable for every subset  $R \subset V(G)$  with  $|R| \leq k + 2$ . In order to see this, suppose that  $R \subseteq V(G)$  with  $|R| \leq k + 2$ . We may assume without loss of generality that  $|R| = k + 2$ . Let  $s$  and  $t$  be two vertices of  $R$ . By definition,  $G^* = G - (R \setminus \{s, t\})$  has a

Hamilton path with end-vertices  $s$  and  $t$ . Hence  $G - R = G^* - \{s, t\}$  is traceable. Below we apply this claim twice.

Because  $G$  is not complete,  $G$  has a scattering set  $S$ . By definition,  $S$  is a vertex cut. Hence  $S = \{s_1, \dots, s_\ell\}$  for some  $\ell \geq k + 3$ , as otherwise  $G - S$  would be traceable, and thus connected, due to our claim. Let  $T = \{s_1, \dots, s_{k+2}\}$  and let  $U = \{s_{k+3}, \dots, s_\ell\}$ . By our claim,  $G' = G - T$  is traceable implying that  $\text{sc}(G') \leq 1$  [38]. Because  $c(G' - U) = c(G - S) \geq 2$ , we find that  $U$  is a vertex cut of  $G'$ . We use these two facts to derive that

$$\begin{aligned}
1 &\geq \text{sc}(G') \\
&\geq c(G' - U) - |U| \\
&= c(G - T - U) - |T| - |U| + |T| \\
&= c(G - S) - |S| + |T| \\
&= \text{sc}(G) + |T| \\
&= \text{sc}(G) + k + 2,
\end{aligned}$$

implying that  $\text{sc}(G) \leq 1 - (k + 2) = -(k + 1)$ , as required.



**Fig. 2.** The essential cases in the proof of Theorem 3 for  $k = 0$ .

Now suppose that  $\text{sc}(G) \leq -(k+1)$ . First let  $k = 0$ . By Theorem 2, there exists a spanning 3-stave  $\mathcal{P} = (P, Q, R)$  between  $u_1$  and  $u_n$ . Let  $v, w$  be an arbitrary pair of vertices of  $G$ . We distinguish four cases in order to find a Hamilton path between  $v$  and  $w$ ; see Fig. 2 for an illustration.

**Case 1:**  $v = u_1$  and  $w = u_n$ . In this case,  $\text{merge}(P, Q, R)$  is the desired Hamilton path.

**Case 2:**  $v = u_1$  and  $w \neq u_n$ . Assume without loss of generality that  $w \in R$ . We split  $R$  before  $w$  into the subpaths  $R_1$  and  $R_2$ , i.e.,  $w$  becomes the first vertex of  $R_2$  and it does not belong to  $R_1$ . Then  $\text{merge}(P, Q, R_1) \circ R_2^{-1}$  is the desired path. The case with  $v \neq u_1$  and  $w = u_n$  is symmetric.

**Case 3:**  $v \neq u_1$  and  $w \neq u_n$  belong to different paths, say  $v \in Q$  and  $w \in R$ . We split  $Q$  after  $v$  into  $Q_1$  and  $Q_2$ , and we also split  $R$  before  $w$ , as above. Then  $Q_1^{-1} \circ \text{merge}(P, Q_2, R_1) \circ R_2^{-1}$  is the desired path.

**Case 4:**  $v \neq u_1$  and  $w \neq u_n$  belong to the same path, say  $Q$ . Without loss of generality, assume that both  $v \neq u_1$  and  $w \neq u_n$  appear in this order on  $Q$ . We split  $Q$  after  $v$  and before  $w$  into three subpaths  $Q_1, Q_2, Q_3$ . If  $v$  and  $w$  are consecutive on  $Q$ , i.e., when  $Q_2$  is empty, then  $Q_1^{-1} \circ \text{merge}(P, R) \circ Q_3^{-1}$  is the desired path. Otherwise, let  $z$  be any vertex on  $R$  that is a neighbor of the first vertex of  $Q_2$ . Such  $z$  exists since the path  $R$  dominates  $G$ . We split  $R$  after  $z$  into  $R_1$  and  $R_2$ . By the choice of  $z$ ,  $R_1$  and  $Q_2$  can be combined through  $z$  into a valid path  $R'$  containing exactly the same vertices as  $R_1$  and  $Q_2$  and starting at  $u_1$ . Then we choose  $Q_1^{-1} \circ \text{merge}(P, R', R_2) \circ Q_3^{-1}$ .

Now let  $k \geq 1$ . Let  $S$  be a set of vertices with  $|S| \leq k$ . We need to show that  $G - S$  is Hamilton-connected. Let  $T$  be a scattering set of  $G - S$  and let  $S^* = S \cup T$ . Because  $T$  is a scattering set of  $G - S$ , we find that  $S^*$  is a vertex cut of  $G$ . We use this to derive that

$$\begin{aligned} \text{sc}(G - S) &= c(G - S - T) - |T| \\ &= c(G - S^*) - |S^*| + |S^*| - |T| \\ &\leq \text{sc}(G) + k - 0 \\ &\leq -1. \end{aligned}$$

Then, by returning to the case  $k = 0$  with  $G - S$  instead of  $G$ , we find that  $G - S$  is Hamilton-connected, as required. This completes the proof of Theorem 3.  $\square$

## 4 Future Work

We conclude our paper by posing a number of open problems. We start with recalling two open problems posed in the literature, the first of which is the aforementioned question of Damaschke [18]:

1. *What is the complexity of 2-HAMILTON PATH for interval graphs?*

Our results imply that we may restrict ourselves to interval graphs with scattering number equal to 0 or 1. This can be seen as follows. Let  $G$  be an interval graph that together with two of its vertices  $u$  and  $v$  forms an instance of 2-HAMILTON PATH. We apply Corollary 1 to compute  $\text{sc}(G)$  in  $O(n + m)$  time. If  $\text{sc}(G) < 0$ , then  $G$  is Hamilton-connected by Theorem 1. Then, by definition, there exists a Hamilton path between  $u$  and  $v$ . If  $\text{sc}(G) > 1$ , then  $G$  is not traceable, also due to Theorem 1. Hence, there exists no Hamilton path between  $u$  and  $v$ .

The second open problem the literature is due to Asdre and Nikolopoulos [3], who consider the  $\ell$ -PATH COVER problem. This problem generalizes the 1-PATH COVER problem and is that of determining the size of a smallest path cover of a graph  $G$  subject to the additional condition that every vertex of a given set  $T$  of size  $\ell$  is an end-vertex of a path in the path cover. Note that this problem generalizes 2-HAMILTON PATH. Asdre and Nikolopoulos ask the following question:

2. *What is the complexity of  $\ell$ -PATH COVER for interval graphs?*

In another paper [2], Asdre and Nikolopoulos proved that  $\ell$ -PATH COVER, and hence 2-HAMILTON PATH, can be solved in  $O(n + m)$  time on proper interval graphs.

The SPANNING STAVE problem is that of computing the minimum value of  $p$  for which a given graph has a spanning  $p$ -stave. Because a Hamilton path of a graph is a spanning 1-stave and HAMILTON PATH is NP-complete, this problem is NP-hard in general.

3. *What is the complexity of SPANNING STAVE for interval graphs?*

The following example shows that we cannot generalize Lemma 1 and apply Algorithm 1 as an attempt to solve this problem. Take the graph  $K_4 - e$ , which has four vertices  $a, b, c, d$  and five edges, say  $(a, b), (a, c), (b, c), (b, d), (c, d)$ . This graph is interval. However, we only have a spanning 2-stave between  $a$  and  $d$  (as their degrees are 2) but there is a spanning 3-stave between  $b$  and  $c$ , namely paths  $\{b, a, c; b, c; b, d, c\}$ .

Chen et al. [14] define the *spanning connectivity* of a Hamilton-connected graph  $G$  as the largest integer  $q$  such that  $G$  has a spanning  $p$ -stave between any two vertices of  $G$  for all integers  $1 \leq p \leq q$ . So, for instance, the complete graph on  $n$  vertices has spanning connectivity  $n - 1$ , and a graph has spanning connectivity at least 1 if and only if it is Hamilton-connected. By the latter statement, the corresponding optimization problem SPANNING CONNECTIVITY is NP-hard. We posed as an open problem to determine the complexity of this problem for proper interval graphs and interval graphs [7]. In response, very recently, Li and Wu [33] announced an  $O(n + m)$  time algorithm for solving SPANNING CONNECTIVITY on interval graphs.

Kratsch, Kloks and Müller [30] gave an  $O(n^3)$  time algorithm for solving TOUGHNESS on interval graphs. We showed that SCATTERING NUMBER can be solved in linear time on interval graphs.

4. *Can TOUGHNESS be solved in linear time on interval graphs?*

Finally, we ask whether our algorithmic results can be generalized to superclasses of interval graphs, such as circular-arc graphs or cocomparability graphs. The complexity status of HAMILTON CONNECTIVITY is still open for both circular-arc graphs and cocomparability graphs, although HAMILTON CYCLE can be solved in  $O(n^2 \log n)$  time on circular-arc graphs [38] and in  $O(n^3)$  time on cocomparability graphs [21]. It is known that SCATTERING NUMBER can be solved in  $O(n^4)$  time on circular-arc graphs and in polynomial time on cocomparability graphs of bounded dimension [30].

5. *Can HAMILTON CONNECTIVITY be solved in polynomial time for circular-arc graphs or cocomparability graphs?*

6. *Can SCATTERING NUMBER be solved in linear time for circular-arc graphs or cocomparability graphs?*

It seems that our approach by relating the Hamilton connectivity of a graph to its scattering number via the existence of a  $k$ -stave has been tailored just for the class of interval graphs. The method fails for any graph class that contains all complete bipartite graphs  $K_{n,n}$ , so for example, for the class of cocomparability graphs and many of its subclasses, such as the classes of permutation graphs and convex graphs. For graphs from these classes, it is no longer clear which two vertices must be chosen as the “leftmost” and “rightmost” vertex, respectively. It can be seen from the previous example, which considered the graph  $K_4 - e$ , that this choice is important even for interval graphs. For all  $n \geq 2$ , the complete bipartite graph  $K_{n,n}$  is not Hamilton-connected. However, there exists a spanning  $n$ -stave between any pair of adjacent vertices, but only a spanning 2-stave between the remaining pairs.

*Acknowledgement.* We thank referees for valuable references to related work and for useful comments that helped us to improve the readability of our paper.

## References

1. S.R. Arikati and C. Pandu Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inf. Proc. Let.* 35 (1990) 149–153.
2. K. Asdre and S.D. Nikolopoulos, A polynomial solution to the k-fixed-endpoint path cover problem on proper interval graphs, *Theor. Comp. Sci.* 411 (2010) 967–975.
3. K. Asdre and S.D. Nikolopoulos, The 1-fixed-endpoint path cover problem is polynomial on interval graphs, *Algorithmica* 58 (2010) 679–710.
4. D. Bauer, S.L. Hakimi, and E. Schmeichel, Recognizing tough graphs is NP-hard, *Disc. Appl. Math.* 28 (1990) 191–195.
5. A.A. Bertossi, Finding Hamiltonian circuits in proper interval graphs, *Inf. Proc. Let.* 17 (1983) 97–101.
6. A.A. Bertossi and M.A. Bonucelli, Hamilton circuits in interval graph generalizations, *Inf. Proc. Let.* 23 (1986) 195–200.
7. H.J. Broersma, J. Fiala, P.A. Golovach, T. Kaiser, D. Paulusma and A. Proskurowski, Linear-time algorithms for scattering number and Hamilton-connectivity of interval graphs, *Proc. WG 2013, Lecture Notes in Computer Science* 8165 (2013) 127–138.
8. J.A. Bondy and U.S.R. Murty, *Graph Theory*, vol. 244 of Graduate Texts in Mathematics, Springer Verlag, 2008.
9. K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms, *J. Com. Sys. Sci.* 13 (1976) 335–379.
10. A. Brandstädt, V.B. Le and J. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
11. M.-S. Chang, S.-L. Peng, and J.-L. Liaw, Deferred-query: An efficient approach for some problems on interval graphs, *Networks* 34 (1999) 1–10.
12. C. Chen and C.-C. Chang, Connected proper interval graphs and the guard problem in spiral polygons, *Combinatorics and Computer Science, LNCS* 1120 (1996) 39–47.
13. C. Chen, C.-C. Chang, and G.J. Chang, Proper interval graphs and the guard problem, *Disc. Math.* 170 (1997) 223–230.
14. Y. Chen, Z.-H. Chen, H.-J. Lai, P. Li, and E. Wei, On spanning disjoint paths in line graphs, *Graphs and Combinatorics* 29 (2013) 1721–1731.
15. G. Chen, M.S. Jacobson, A.E. Kézdy and J. Lehel, Tough enough chordal graphs are Hamiltonian, *Networks* 31 (1998) 29–38.
16. V. Chvátal, Tough graphs and Hamiltonian circuits, *Disc. Math.* 5 (1973) 215–228.
17. D.G. Corneil, B. Dalton, M. Habib, LDFS-based certifying algorithm for the Minimum Path Cover problem on cocomparability graphs, *SIAM J. Comput.* 42 (2013) 792–807.
18. P. Damaschke, Paths in interval graphs and circular arc graphs, *Disc. Math.* 112 (1993) 49–64.
19. A.M. Dean, The computational complexity of deciding Hamiltonian-connectedness, *Congr. Num.* 93 (1993) 209–214.
20. J.S. Deogun, D. Kratsch and G. Steiner, 1-tough cocomparability graphs are Hamiltonian, *Disc. Math.*, 170 (1997) 99–106.
21. J.S. Deogun and G. Steiner, Polynomial algorithms for Hamiltonian cycle in cocomparability graphs, *SIAM J Comp.* 23 (1994) 520–552.
22. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co Ltd, 1979.
23. M.R. Garey, D.S. Johnson, and R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, *SIAM J Comp.* 5 (1976) 704–714.
24. D. Hsu, On container width and length in graphs, groups, and networks, *IEICE Trans. Fund*, E77-A (1994) 668–680.
25. R.-W. Hung and M.-S. Chang, Linear-time certifying algorithms for the path cover and Hamiltonian cycle problems on interval graphs, *Appl. Math. Lett.* 24 (2011) 648–652.
26. K. Ioannidou, G.B. Mertzios, and S.D. Nikolopoulos, The longest path problem has a polynomial solution on interval graphs, *Algorithmica* 61 (2011) 320–341.
27. G. Isaak, Powers of Hamiltonian paths in interval graphs, *J Graph Th.* 28 (1998) 31–38.

28. H.A. Jung, On a class of posets and the corresponding comparability graphs, *J Comb. Th. B* 24 (1978) 125–133.
29. J. M. Keil, Finding Hamiltonian circuits in interval graphs, *Inf. Proc. Let.* 20 (1985) 201–206.
30. D. Kratsch, T. Kloks, and H. Müller, Measuring the vulnerability for classes of intersection graphs, *Disc. Appl. Math.* 77 (1997) 259–270.
31. R. Kužel, Z. Ryjáček, and P. Vrána, Thomassen’s conjecture implies polynomiality of 1-Hamilton-connectedness in line graphs, *J Graph Th.* 69 (2012) 241–250.
32. P. Li and Y. Wu, A linear time algorithm for solving the 1-fixed-endpoint path cover problem on interval graphs, *Manuscript*.
33. P. Li and Y. Wu, Spanning connectedness and Hamiltonian thickness of graphs and interval graphs, *Manuscript*.
34. C.-K. Lin, H.-M. Huang, and L.-H. Hsu, On the spanning connectivity of graphs, *Discrete Mathematics* 307 (2007) 285–289.
35. J. Lehel, The path partition of cocomparability graphs, *manuscript*, 1991.
36. G.K. Manacher, T.A. Mankus, and C.J. Smith, An optimum  $\Theta(n \log n)$  algorithm for finding a canonical Hamiltonian path and a canonical Hamiltonian circuit in a set of intervals, *Inf. Proc. Let.* 35 (1990) 205–211.
37. H. Müller, Hamiltonian circuits in chordal bipartite graphs, *Disc. Math.* 156 (1996) 291–298.
38. W.K. Shih, T.C. Chern, and W.L. Hsu, An  $O(n^2 \log n)$  time algorithm for the Hamiltonian cycle problem on circular-arc graphs, *SIAM J. Comp.* 21 (1992) 1026–1046.