Theoretical Computer Science 689 (2017) 67–95

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Polynomial fixed-parameter algorithms: A case study for longest path on interval graphs [☆,☆☆]

Archontia C. Giannopoulou [a,1], George B. Mertzios [b,*], Rolf Niedermeier [a]

[a] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
[b] School of Engineering and Computing Sciences, Durham University, UK

ARTICLE INFO

*Article history:*
Received 12 April 2017
Received in revised form 8 May 2017
Accepted 21 May 2017
Available online 25 May 2017
Communicated by P.G. Spirakis

*Keywords:*
Polynomial-time algorithm
Longest path problem
Interval graphs
Proper interval vertex deletion set
Data reduction
Fixed-parameter algorithm

ABSTRACT

We study the design of fixed-parameter algorithms for problems already known to be solvable in polynomial time. The main motivation is to get more efficient algorithms for problems with unattractive polynomial running times. Here, we focus on a fundamental graph problem: Longest Path, that is, given an undirected graph, find a maximum-length path in $G$. Longest Path is NP-hard in general but known to be solvable in $O(n^4)$ time on $n$-vertex interval graphs. We show how to solve Longest Path on Interval Graphs, parameterized by vertex deletion number $k$ to proper interval graphs, in $O(k^9 n)$ time. Notably, Longest Path is trivially solvable in linear time on proper interval graphs, and the parameter value $k$ can be approximated up to a factor of 4 in linear time. From a more general perspective, we believe that using parameterized complexity analysis may enable a refined understanding of efficiency aspects for polynomial-time solvable problems similarly to what classical parameterized complexity analysis does for NP-hard problems.



## 1. Introduction

Parameterized complexity analysis [20,22,24,44] is a flourishing field dealing with the exact solvability of NP-hard problems. The key idea is to lift classical complexity analysis, rooted in the P versus NP phenomenon, from a one-dimensional to a two- (or even multi-)dimensional perspective, the key concept being "fixed-parameter tractability (FPT)". But why should this natural and successful approach be limited to intractable (i.e., NP-hard) problems? We are convinced that appropriately parameterizing polynomially solvable problems sheds new light on what makes a problem far from being solvable in *linear* time, in the same way as classical FPT algorithms help in illuminating what makes an NP-hard problem far from being solvable in *polynomial* time. In a nutshell, the credo and leitmotif of this paper is that "FPT inside P" is a very interesting, but still too little explored, line of research.

The known results fitting under this leitmotif are somewhat scattered around in the literature and do not systematically refer to or exploit the toolbox of parameterized algorithm design. This should change and "FPT inside P" should be placed

☆ A preliminary conference version of this work appears in the *Proceedings of the 10th International Symposium on Parameterized and Exact Computation (IPEC)*, Patras, Greece, September 2015, pages 102–113 [28].

☆☆ Partially supported by the EPSRC Grant EP/K022660/1.

* Corresponding author.
*E-mail addresses:* archontia.giannopoulou@gmail.com (A.C. Giannopoulou), george.mertzios@durham.ac.uk (G.B. Mertzios), rolf.niedermeier@tu-berlin.de (R. Niedermeier).

1 The main part of this paper was prepared while the author was affiliated at the School of Engineering and Computing Sciences, Durham University, UK.

http://dx.doi.org/10.1016/j.tcs.2017.05.017
0304-3975/

on a much wider footing, using parameterized algorithm design techniques such as data reduction and kernelization. As a simple illustrative example, consider the Maximum Matching problem. By following a "Buss-like" kernelization (as is standard knowledge in parameterized algorithmics [22,44]) and then applying a known polynomial-time matching algorithm, it is not difficult to derive an efficient algorithm that, given a graph $G$ with $n$ vertices, computes a matching of size at least $k$ in $O(kn+k^3)$ time. For the sake of completeness we present the details of this algorithm in Section 5.

More formally, and somewhat more generally, we propose the following scenario. Given a problem with instance size $n$ for which there exists an $O(n^c)$-time algorithm, our aim is to identify appropriate parameters $k$ and to derive algorithms with time complexity $f(k)\cdot n^{c'}$ such that $c'<c$, where $f(k)$ depends only on $k$. First we refine the class FPT by defining, for every polynomially-bounded function $p(n)$, the class FPT($p(n)$) containing the problems solvable in $f(k)\cdot p(n)$ time, where $f(k)$ is an arbitrary (possibly exponential) function of $k$. It is important to note that, in strong contrast to FPT algorithms for NP-hard problems, here the function $f(k)$ may also become *polynomial* in $k$. Motivated by this, we refine the complexity class P by introducing, for every polynomial function $p(n)$, the class *P-FPT*($p(n)$) *(Polynomial Fixed-Parameter Tractable)*, containing the problems solvable in $O(k^t\cdot p(n))$ time for some constant $t\geq 1$, i.e., the dependency of the complexity on the parameter $k$ is at most polynomial. In this paper we focus our attention on the (practically perhaps most attractive) subclass *PL-FPT (Polynomial-Linear Fixed-Parameter Tractable)*, where PL-FPT = P-FPT($n$). For example, the algorithm we sketched above for Maximum Matching, parameterized by solution size $k$, yields containment in the class PL-FPT.

In an attempt to systematically follow the leitmotif "FPT inside P", we put forward three desirable algorithmic properties:

1. The running time should have a polynomial dependency on the parameter.
2. The running time should be as close to linear as possible if the parameter value is constant, improving upon an existing "high-degree" polynomial-time (unparameterized) algorithm.
3. The parameter value, or a good approximation thereof, should be computable efficiently (preferably in linear time) for arbitrary parameter values.

In addition, as this research direction is still only little explored, we suggest to focus first on problems for which the best known upper bounds of the time complexity are polynomials of high degree, e.g., $O(n^4)$ or higher.

**Related work.** Here we discuss previous work on *graph problems* that fits under the leitmotif "FPT inside P"; however, there exists further related work also in other topics such as string matching [6], XPath query evaluation in XML databases [12], and Linear Program solving [40].

The complexity of some known polynomial-time algorithms can be easily "tuned" with respect to specific parameters, thus immediately reducing the complexity whenever these parameters are bounded. For instance, in $n$-vertex and $m$-edge graphs with nonnegative edge weights, Dijkstra's $O(m+n\log n)$-time algorithm for computing shortest paths can be adapted to an $O(m+n\log k)$-time algorithm, where $k$ is the number of distinct edge weights [39] (also refer to [46]). In addition, motivated by the quest for explaining the efficiency of several shortest path heuristics for road networks (where Dijkstra's algorithm is too slow for routing applications), the "highway dimension" was introduced [4] as a parameterization helping to do rigorous proofs about the quality of the heuristics. Altogether, the work on shortest path computations shows that, despite of known quasi-linear-time algorithms, adopting a parameterized view may be of significant (practical) interest.

Maximum flow computations constitute another important application area for "FPT inside P". An $O(k^3 n\log n)$-time maximum flow algorithm was presented [33] for graphs that can be made planar by deleting $k$ "crossing edges"; notably, here it is assumed that the embedding and the $k$ crossing edges are given along with the input. An $O(g^8 n\log^2 n\log^2 C)$-time maximum flow algorithm was developed [17], where $g$ is the genus of the graph and $C$ is the sum of all edge capacities; here it is also assumed that the embedding and the parameter $g$ are given in the input. Finally, we remark that multiterminal flow [31] and Wiener index computations [14] have exploited the treewidth parameter, assuming that the corresponding tree decomposition of the graph is given. However, in both publications [14,31] the dependency on the parameter $k$ is *exponential*.

We finally mention that, very recently, two further works delved deeper into "FPT inside P" algorithms for Maximum Matching [25,42].

**Our contribution.** In this paper, to illustrate the potential algorithmic challenges posed by the "FPT inside P" framework (which seem to go clearly beyond the known "FPT inside P" examples), we focus on Longest Path on Interval Graphs, which is known to be solvable in $O(n^4)$ time [35], and we derive a PL-FPT-algorithm (with the appropriate parameterization) that satisfies all three desirable algorithmic properties described above.

The Longest Path problem asks, given an undirected graph $G$, to compute a maximum-length path in $G$. On general graphs, the decision variant of Longest Path is NP-complete and many FPT algorithms have been designed for it, e.g., [5,10,18,26,38,50], contributing to the parameterized algorithm design toolkit techniques such as color-coding [5] (and further randomized techniques [18,38]) as well as algebraic approaches [50]. The currently best known deterministic FPT algorithm runs in $O(2.851^k n\log^2 n\log W)$ time on weighted graphs with maximum edge weight $W$, where $k$ is the number of vertices in the path [26], while the currently best known randomized FPT algorithm runs in $O(1.66^k n^{O(1)})$ time with constant, one-sided error [10]. Longest Path is known to be solvable in polynomial time only on very few non-trivial graph classes [35,41] (see also [49] for much smaller graph classes). This problem has also been studied on directed graphs; a polynomial-time algorithm was given by Gutin [30] for the class of orientations of multipartite tournaments, which was later extended by

Bang-Jensen and Gutin [7]. With respect to undirected graphs, a few years ago it was shown that Longest Path on Interval Graphs can be solved in polynomial time, providing an algorithm that runs in $O(n^4)$ time [35]; this algorithm has been extended with the same running time to the larger class of cocomparability graphs [41] using a lexicographic depth first search (LDFS) approach. Notably, a longest path in a *proper* interval graph can be computed by a *trivial* linear-time algorithm since every connected proper interval graph has a Hamiltonian path [8]. Consequently, as the classes of interval graphs and of proper interval graphs seem to behave quite differently, it is natural to parameterize Longest Path on Interval Graphs by the size $k$ of a *minimum proper interval (vertex) deletion set*, i.e., by the minimum number of vertices that need to be deleted to obtain a proper interval graph. That is, this parameterization exploits what is also known as "distance from triviality" [16,23,29,45] in the sense that the parameter $k$ measures how far a given input instance is from a trivially solvable special case. As it turns out, one can compute a 4-approximation of $k$ in $O(n+m)$ time for an interval graph with $n$ vertices and $m$ edges. Using this constant-factor approximation of $k$, we provide a polynomial fixed-parameter algorithm that runs in $O(k^9 n)$ time, thus proving that Longest Path on Interval Graphs is in the class PL-FPT when parameterized by the size of a minimum proper interval deletion set.

To develop our algorithm, we first introduce in Section 2 two data reduction rules on interval graphs. Each of these reductions shrinks the size of specific vertex subsets, called *reducible* and *weakly reducible* sets, respectively. Then, given any proper interval deletion set $D$ of an interval graph $G$, in Section 3 we appropriately decompose the graph $G \setminus D$ into two collections $\mathcal{S}_1$ and $\mathcal{S}_2$ of reducible and weakly reducible sets, respectively, on which we apply the reduction rules of Section 2. The resulting interval graph $\widehat{G}$ is *weighted* (with weights on its vertices) and has some special properties; we call $\widehat{G}$ a *special weighted interval graph* with *parameter* $\kappa$, where in this case $\kappa = O(k^3)$. Notably, although $\widehat{G}$ has reduced size, it still has $O(n)$ vertices. Then, in Section 4 we present a fixed-parameter algorithm (with parameter $\kappa$) computing in $O(\kappa^3 n)$ time the maximum weight of a path in a special weighted interval graph. We note here that such a maximum-weight path in a special weighted interval graph can be directly mapped back to a longest path in the original interval graph. Thus, our parameterized algorithm computes a longest path in the initial interval graph $G$ in $O(\kappa^3 n) = O(k^9 n)$ time.

Turning our attention away from Longest Path on Interval Graphs we present for the sake of completeness our "Buss-like" kernelization of the Maximum Matching problem in Section 5. Using this kernelization an efficient algorithm can be easily deduced which, given an arbitrary graph $G$ with $n$ vertices, computes a matching of size at least $k$ in $G$ in $O(kn + k^3)$ time. Finally, in the concluding Section 6 we discuss our contribution and provide a brief outlook for future research directions.

**Notation.** We consider finite, simple, and undirected graphs. Given a graph $G$, we denote by $V(G)$ and $E(G)$ the sets of its vertices and edges, respectively. A graph $G$ is *weighted* if it is given along with a weight function $w : V(G) \to \mathbb{N}$ on its vertices. An edge between two vertices $u$ and $v$ of a graph $G = (V, E)$ is denoted by $uv$, and in this case $u$ and $v$ are said to be *adjacent*. The *neighborhood* of a vertex $u \in V$ is the set $N(u) = \{v \in V \mid uv \in E\}$ of its adjacent vertices. The cardinality of $N(u)$ is the degree $\deg(u)$ of $u$. For every subset $S \subseteq V$ we denote by $G[S]$ the subgraph of $G$ induced by the vertex set $S$ and we define $G \setminus S = G[V \setminus S]$. A set $S \subseteq V$ induces an *independent set* (resp. a *clique*) in $G$ if $uv \notin E$ (resp. if $uv \in E$) for every pair of vertices $u, v \in S$. Furthermore, $S$ is a *vertex cover* if and only if $V \setminus S$ is an independent set. For any two graphs $G_1, G_2$, we write $G_1 \subseteq G_2$ if $G_1$ is an induced subgraph of $G_2$. A *matching* $M$ in a graph $G$ is a set of edges of $G$ without common vertices. All paths considered in this paper are simple. Whenever a path $P$ visits the vertices $v_1, v_2, \ldots, v_k$ in this order, we write $P = (v_1, v_2, \ldots, v_k)$. Furthermore, for two vertex-disjoint paths $P = (a, \ldots, b)$ and $Q = (c, \ldots, d)$ where $bc \in E$, we denote by $(P, Q)$ the path $(a, \ldots, b, c, \ldots, d)$.

A graph $G = (V, E)$ is an *interval* graph if each vertex $v \in V$ can be bijectively assigned to a closed interval $I_v$ on the real line, such that $uv \in E$ if and only if $I_u \cap I_v \neq \emptyset$, and then the collection of intervals $\mathcal{I} = \{I_v : v \in V\}$ is called an *interval representation* of $G$. The interval graph $G$ is a *proper interval graph* if it admits an interval representation $\mathcal{I}$ such that $I_u \nsubseteq I_v$ for every $u, v \in V$, and then $\mathcal{I}$ is called a *proper interval representation*. Given an interval graph $G = (V, E)$, a subset $D \subseteq V$ is a *proper interval deletion set* of $G$ if $G \setminus D$ is a proper interval graph. The *proper interval deletion number* of $G$ is the size of the smallest proper interval deletion set. Finally, for any positive integer $t$, we denote $[t] = \{1, 2, \ldots, t\}$.

## 2. Data reductions on interval graphs

In this section we present two data reductions on interval graphs. The first reduction (cf. Section 2.2) shrinks the size of a collection of vertex subsets of a certain kind, called *reducible* sets, and it produces a *weighted* interval graph. The second reduction (cf. Section 2.3) is applied to an arbitrary weighted interval graph; it shrinks the size of a collection of another kind of vertex subsets, called *weakly reducible* sets, and it produces a smaller weighted interval graph. Both reductions retain as an invariant the maximum path weight. The proof of this invariant is based on the crucial notion of a *normal* path in an interval graph (cf. Section 2.1). The following vertex ordering characterizes interval graphs [47]. Moreover, given an interval graph $G$ with $n$ vertices and $m$ edges, this vertex ordering of $G$ can be computed in $O(n+m)$ time [47].

**Lemma 1** *([47]). A graph $G$ is an interval graph if and only if there is an ordering $\sigma$ (called* right-endpoint ordering*) of $V(G)$ such that for all $u <_\sigma v <_\sigma z$, if $uz \in E(G)$ then also $vz \in E(G)$.*

In the remainder of the paper we assume that we are given an interval graph $G$ with $n$ vertices and $m$ edges as input, together with an interval representation $\mathcal{I}$ of $G$, where the endpoints of the intervals are given sorted increasingly. Without

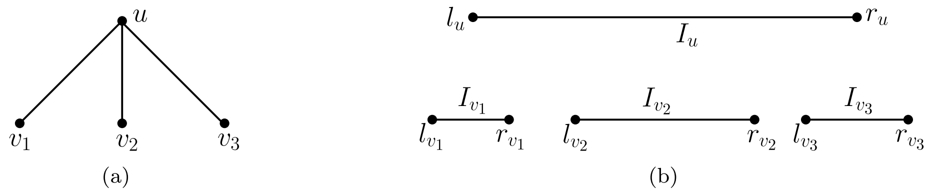


**Fig. 1.** (a) The forbidden induced subgraph (claw $K_{1,3}$) for an interval graph to be a proper interval graph and (b) an interval representation of the $K_{1,3}$.

loss of generality, we assume that the endpoints of all intervals are distinct. For every vertex $v \in V(G)$ we denote by $I_v = [l_v, r_v]$ the interval of $\mathcal{I}$ that corresponds to $v$, i.e., $l_v$ and $r_v$ are the left and the right endpoint of $I_v$, respectively. In particular, $G$ is assumed to be given along with the *right-endpoint ordering* $\sigma$ of its vertices $V(G)$, i.e., $u <_\sigma v$ if and only if $r_u < r_v$ in the interval representation $\mathcal{I}$ (see also Lemma 1). Given a set $S \subseteq V(G)$, we denote by $\mathcal{I}[S]$ the interval representation induced from $\mathcal{I}$ on the intervals of the vertices of $S$. We say that two vertices $u_1, u_2 \in S$ are *consecutive in* $S$ (with respect to the vertex ordering $\sigma$) if $u_1 <_\sigma u_2$ and for every vertex $u \in S \setminus \{u_1, u_2\}$ either $u <_\sigma u_1$ or $u_2 <_\sigma u$. Furthermore, for two sets $S_1, S_2 \subseteq V(G)$, we write $S_1 <_\sigma S_2$ whenever $u <_\sigma v$ for every $u \in S_1$ and $v \in S_2$. Finally, we denote by $\mathbf{span}(S)$ the interval $[\min\{l_v : v \in S\}, \max\{r_v : v \in S\}]$.

It is well known that an interval graph $G$ is a proper interval graph if and only if $G$ is $K_{1,3}$-free, i.e., if $G$ does not include the claw $K_{1,3}$ with four vertices (cf. Fig. 1) as an induced subgraph [48]. It is worth noting here that, to the best of our knowledge, it is unknown whether a minimum proper interval deletion set of an interval graph $G$ can be computed in polynomial time. However, since there is a unique forbidden induced subgraph $K_{1,3}$ on four vertices, we can apply Cai's generic algorithm [15] on an arbitrary given interval graph $G$ with $n$ vertices to compute a proper interval deletion set of $G$ of size at most $k$ in FPT time $4^k \cdot \mathrm{poly}(n)$. The main idea of Cai's bounded search tree algorithm in our case is that we repeat the following two steps until we either get a $K_{1,3}$-free graph or have used up the deletion of $k$ vertices from $G$: (i) detect an induced $K_{1,3}$, (ii) branch on deleting one of the four vertices of the detected $K_{1,3}$. At every iteration of this process we have four possibilities on the next vertex to delete. Thus, since we can delete up to $k$ vertices in total, the whole process finishes after at most $4^k \cdot \mathrm{poly}(n)$ steps.

As we prove in the next theorem, a 4-approximation of the minimum proper interval deletion number of an interval graph can be computed much more efficiently.

**Theorem 1.** *Let $G = (V, E)$ be an interval graph, where $|V| = n$ and $|E| = m$. Let $k$ be the size of the minimum proper interval deletion set of $G$. Then a proper interval deletion set $D$ of size at most $4k$ can be computed in $O(n + m)$ time.*

**Proof.** Let $\{u, v_1, v_2, v_3\}$ be a set of four vertices that induces a $K_{1,3}$ in $G$ such that $v_1, v_2, v_3 \in N(u)$, $v_1 <_\sigma v_2 <_\sigma v_3$, and $I_{v_2} \subseteq I_u$ in the interval representation $\mathcal{I}$ (cf. Fig. 1). Let $v_1'$ (resp. $v_3'$) be the neighbor of vertex $u$ with the leftmost right endpoint $r_{v_1'}$ (resp. with the rightmost left endpoint $l_{v_3'}$) in the representation $\mathcal{I}$, i.e., $r_{v_1'} = \min\{r_v : v \in N(u)\}$ and $l_{v_3'} = \max\{l_v : v \in N(u)\}$. Then note that the set $\{u, v_1', v_2, v_3'\}$ also induces a $K_{1,3}$ in $G$.

We now describe an $O(n + m)$-time algorithm that iteratively detects an induced $K_{1,3}$ and removes its vertices from the current graph. During its execution the algorithm maintains a set $D$ of "marked" vertices; a vertex is marked if it has been removed from the graph at a previous iteration. Initially, $D = \emptyset$, i.e., all vertices are unmarked. The algorithm processes once every vertex $u \in V$ in an arbitrary order. If $u \in D$ (i.e., if $u$ has been marked at a previous iteration), then the algorithm ignores $u$ and proceeds with the next vertex in $V$. If $u \notin D$ (i.e., if $u$ is unmarked), then the algorithm iterates for every vertex $v \in N(u) \setminus D$ and it computes the vertices $z_1(u), z_2(u) \in N(u) \setminus D$ such that $r_{z_1(u)} = \min\{r_v : v \in N(u) \setminus D\}$ and $l_{z_2(u)} = \max\{l_v : v \in N(u) \setminus D\}$. In the case where $N(u) \setminus D = \emptyset$, the algorithm defines $z_1(u) = z_2(u) = u$. Then the algorithm iterates once again for every vertex $v \in N(u) \setminus D$ and it checks whether the set $\{u, v, z_1(u), z_2(u)\}$ induces a $K_{1,3}$ in $G$. If it detects at least one vertex $v \in N(u) \setminus D$ such that $\{u, v, z_1(u), z_2(u)\}$ induces a $K_{1,3}$, then it marks all four vertices $\{u, v, z_1(u), z_2(u)\}$, i.e., it adds these vertices to the set $D$. Otherwise the algorithm proceeds with processing the next vertex of $V$. It is easy to check that every vertex $u \in V$ is processed by this algorithm in $O(\deg(u))$ time, and thus all vertices of $V$ are processed in $O(n + m)$ time in total.

The algorithm terminates after it has processed all vertices of $V$ and it returns the computed set $D$ of all quadruples of marked vertices. Note that there are $\frac{|D|}{4}$ such quadruples. This set $D$ is clearly a proper interval deletion set of $G$, since $G \setminus D$ does not contain an induced $K_{1,3}$, i.e., $k \leq |D|$. In addition, each of the detected quadruples of the set $D$ induces a $K_{1,3}$ in the initial interval graph $G$, and thus any minimum proper interval deletion set must contain at least one vertex from each of these quadruples, i.e., $k \geq \frac{|D|}{4}$. Summarizing $k \leq |D| \leq 4k$. $\square$

Note that, whenever four vertices induce a claw $K_{1,3}$ in an interval graph $G$, then in the interval representation $\mathcal{I}$ of $G$ at least one of these intervals is necessarily properly included in another one (e.g., $I_{v_2} \subseteq I_u$ in Fig. 1(b)). However the converse is not always true, as there may exist two vertices $u, v$ in $G$ such that $I_v \subseteq I_u$, although $u$ and $v$ do not belong to any induced claw $K_{1,3}$ in $G$.

**Definition 1.** Let $G = (V, E)$ be an interval graph. An interval representation $\mathcal{I}$ of $G$ is *semi-proper* when, for any $u, v \in V$:

- if $I_v \subseteq I_u$ in $\mathcal{I}$ then the vertices $u$ and $v$ belong to an induced claw $K_{1,3}$ in $G$, i.e. $\{u, v, a, b\}$ induces a claw $K_{1,3}$ in $G$ for some vertices $a, b$.

Every interval representation $\mathcal{I}$ of a graph $G$ can be efficiently transformed into a semi-proper representation $\mathcal{I}'$ of $G$, as we prove in the next theorem. In the remainder of the paper we always assume that this preprocessing step has been already applied to $\mathcal{I}$.

**Theorem 2** *(preprocessing). Given an interval representation $\mathcal{I}$, a semi-proper interval representation $\mathcal{I}'$ can be computed in $O(n+m)$ time.*

**Proof.** Similarly to the proof of Theorem 1, the algorithm first computes for every vertex $u \in V$ the vertices $z_1(u), z_2(u) \in N(u)$, such that $r_{z_1(u)} = \min\{r_v : v \in N(u)\}$ and $l_{z_2(u)} = \max\{l_v : v \in N(u)\}$. If $N(u) = \emptyset$, then the algorithm defines $z_1(u) = z_2(u) = u$.

The algorithm iterates over all $u \in V$. For each $u \in V$, the algorithm iterates over all $v \in N(u)$ such that $I_v \subseteq I_u$ in the current interval representation. Let these vertices be $\{v_1, v_2, \ldots, v_t\}$, where $l_{v_1} < l_{v_2} < \ldots < l_{v_t}$. The algorithm processes the vertices $\{v_1, v_2, \ldots, v_t\}$ in this order. For every $i \in \{1, 2, \ldots, t\}$, if $z_2(u) \in N(v_i)$, then the algorithm increases the right endpoint of $I_{v_i}$ to the point $r_u + \varepsilon_i$, for an appropriately small $\varepsilon_i > 0$. The algorithm chooses the values of $\varepsilon_i$ such that $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_t$. By performing these operations no new adjacencies are introduced, and thus the resulting interval representation remains a representation of the same interval graph $G$.

We note here that the algorithm can be efficiently implemented (i.e., in $O(n+m)$ time in total) without explicitly computing the values of these $\varepsilon_i$, as follows. Since the endpoints of the $n$ intervals of $\mathcal{I}$ are assumed to be given increasingly sorted, we initially scan them from left to right and map them bijectively to the integers $\{1, 2, \ldots, 2n\}$. Then, instead of increasing the right endpoint of $I_{v_i}$ to the point $r_u + \varepsilon_i$ as described above, where $i \in \{1, 2, \ldots, t\}$, we just store the vertices $\{v_1, v_2, \ldots, v_t\}$ (in this order) in a linked list after the endpoint $r_u$. At the end of the whole process (i.e., after dealing with all pairs of vertices $u, v$ such that $v \in N(u)$ and $I_v \subseteq I_u$ in the interval representation $\mathcal{I}$), we scan again all interval endpoints from left to right and re-map them bijectively to the integers $\{1, 2, \ldots, 2n\}$, where in this new mapping we place the endpoints $\{r_{v_1}, r_{v_2}, \ldots, r_{v_t}\}$ (in this order) immediately after $r_u$. This can be clearly done in $O(n+m)$ time.

Then the algorithm iterates (again) over all $v \in N(u)$ such that $I_v \subseteq I_u$ in the current interval representation. Let these vertices be $\{v_1, v_2, \ldots, v_{t'}\}$, where $r_{v_1} > r_{v_2} > \ldots > r_{v_{t'}}$. The algorithm processes the vertices $\{v_1, v_2, \ldots, v_{t'}\}$ in this order. For every $i \in \{1, 2, \ldots, t'\}$, if $z_2(u) \notin N(v_i)$ and $z_1(u) \in N(v_i)$, then the algorithm decreases the left endpoint of $I_{v_i}$ to the point $l_u - \varepsilon_i$, for an appropriately small $\varepsilon_i > 0$. The algorithm chooses the values of $\varepsilon_i$ such that $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_t$. Similarly to the above, no new adjacencies are introduced by performing these operations, and thus the resulting interval representation remains a representation of the same interval graph $G$. Furthermore, the algorithm can be efficiently implemented (i.e., in $O(n+m)$ time in total) without explicitly computing these values of $\varepsilon_i$, similarly to the description in the previous paragraph.

Denote by $\mathcal{I}'$ the resulting interval representation of $G$, which is obtained after performing all the above operations. Furthermore denote by $\sigma'$ the right-endpoint ordering of the intervals in $\mathcal{I}'$. Let $u, v \in V$. It can be easily checked that, if $I_v \subseteq I_u$ in $\mathcal{I}'$, then also $I_v \subseteq I_u$ in the initial representation $\mathcal{I}$. Furthermore, it follows directly by the above construction that, if $I_v \subseteq I_u$ in $\mathcal{I}'$, then $z_1(u), z_2(u) \notin N(v)$, where $z_1 <_{\sigma'} v <_{\sigma'} z_2$, and thus the vertices $\{u, v, z_1(u), z_2(u)\}$ induce a $K_{1,3}$ in $G$.

The computation of the vertices $z_1(u), z_2(u)$ for all vertices $u \in V$ can be done in $O(n+m)$ time. Furthermore, for every $u \in V$ we can visit all vertices $v \in N(u)$ in $O(\deg(u))$ time in the above algorithm, since the endpoints of the intervals are assumed to be given sorted in increasing order. For every such edge $uv \in E$, where $I_v \subseteq I_u$, we can check in $O(1)$ time whether $z_1(u) \in N(v)$ (resp. whether $z_2(u) \in N(v)$) by checking whether $r_v < l_{z_2(u)}$ (resp. by checking whether $r_{z_1(u)} < l_v$). Therefore, the total running time of the algorithm is $O(n+m)$. □

### *2.1. Normal paths*

All our results on interval graphs rely on the notion of a *normal* path [35] (also referred to as a *straight* path in [21,37]). This notion has also been extended to the greater class of cocomparability graphs [41]. Normal paths are useful in the analysis of our data reductions in this section, as well as in our algorithm in Section 4, as they impose certain *monotonicity* properties of the paths. Informally, the vertices in a normal path appear in a "left-to-right fashion" in the right-endpoint ordering $\sigma$. In the following, given a graph $G$ and a path $P = (v_1, v_2, \ldots, v_l)$ of $G$, we write $v_i <_P v_j$ if and only if $i < j$, i.e., whenever $v_i$ precedes $v_j$ in $P$.

**Definition 2.** Let $G = (V, E)$ be an interval graph and $\sigma$ be a right-endpoint ordering of $V$. The path $P = (v_1, v_2, \ldots, v_k)$ of $G$ is *normal* if:

- $v_1$ is the leftmost vertex among $\{v_1, v_2, \ldots, v_k\}$ in $\sigma$ and
- $v_i$ is the leftmost vertex of $N(v_{i-1}) \cap \{v_i, v_{i+1}, \ldots, v_k\}$ in $\sigma$, for every $i = 2, \ldots, k$.

**Lemma 2** *([35]). Let $G$ be an interval graph and $\mathcal{I}$ be an interval representation of $G$. For every path $P$ of $G$, there exists a normal path $P'$ of $G$ such that $V(P) = V(P')$.*

We now provide a few properties of normal paths on interval graphs that we will need later on.

**Observation 1.** Let $G$ be an interval graph and $P$ be a normal path of $G$. Let $y$ be the last vertex of $P$ and $z \in V(G) \setminus V(P)$ such that $yz \in E(G)$ and $v <_\sigma z$ for every vertex $v \in V(P)$. Then $(P, z)$ is a normal path of $G$.

**Observation 2.** Let $G$ be an interval graph, $P$ be a normal path of $G$, and $u, w \in V(P)$. If $u <_P w$ and $w <_\sigma u$, then $u$ is not the first vertex of $P$.

**Lemma 3.** *Let $G$ be an interval graph, $P$ be a normal path of $G$, and $u, w \in V(P)$. If $u <_P w$ and $w <_\sigma u$, then $wu \in E(G)$.*

**Proof.** The proof is done by contradiction. Let $u, w \in V(P)$, where $u <_P w$ and $w <_\sigma u$. Assume that $wu \notin E(G)$. Among all such pairs of vertices, we can assume without loss of generality that $\mathbf{dist}_P(u, w)$ is maximum, where $\mathbf{dist}_P(u, w)$ denotes the distance between the vertices $u$ and $w$ on the path $P$. From Observation 2, $u$ is not the first vertex of $P$, and therefore $u$ has a predecessor, say $z$, in $P$. Note that $\mathbf{dist}_P(z, w) = \mathbf{dist}_P(u, w) + 1$. Suppose that $wz \in E(G)$. Then, since $w <_\sigma u$ and $u <_P w$, it follows by the normality of $P$ that $u$ is not the next vertex of $z$ in the path $P$, which is a contradiction. Therefore $wz \notin E(G)$. Suppose now that $z <_\sigma w$. Then, since in this case $z <_\sigma w <_\sigma u$ and $zu \in E(G)$, it follows by Lemma 1 that $wu \in E(G)$, which is a contradiction to our assumption. Therefore $w <_\sigma z$. Recall that $z <_P u <_P w$, $zw \notin E(G)$, and $\mathbf{dist}_P(z, w) = \mathbf{dist}_P(u, w) + 1$. This is a contradiction to our assumption that $\mathbf{dist}_P(u, w)$ is maximum. Therefore $wu \in E(G)$. This completes the proof of the lemma. □

**Lemma 4.** *Let $G$ be an interval graph and $P = (P_1, u, w, P_2)$ be a normal path of $G$, $u, w \in V(G)$. If $w <_\sigma u$ then $I_w \subseteq I_u$.*

**Proof.** Let us assume, to the contrary, that $I_w \nsubseteq I_u$, that is, $l_w < l_u$. From Observation 2, $u$ is not the first vertex of $P$ and therefore $u$ has a predecessor, say $z$, in $P$. Since $P$ is normal, $w <_\sigma u$, and $u <_P w$, it follows that $w$ is not a neighbor of $z$. Notice then that $z <_P u <_P w$. Furthermore, as $z$ is a neighbor of $u$, $w <_\sigma z$. Summarizing, $z <_P w$, $w <_\sigma z$, and $wz \notin G$. From Lemma 3, this is a contradiction to the assumption that $P$ is normal. Therefore, $I_w \subseteq I_u$. □

**Lemma 5.** *Let $G$ be an interval graph and $\mathcal{I}$ be an interval representation of $G$. Let $S \subseteq V(G)$ such that $\mathcal{I}[S]$ is a proper interval representation of $G[S]$. Let $P$ be a normal path of $G$ and $u, v \in S \cap V(P)$. If $u <_\sigma v$ then $u <_P v$.*

**Proof.** Let $P$ be a normal path of $G$ where $u, v \in V(P)$, $u <_\sigma v$. If $uv \notin E(G)$ then from Lemma 3, we obtain that $u <_P v$. Thus, from now on, we assume that $uv \in E(G)$. Towards a contradiction we further assume that $v <_P u$, that is, $P = (P_1, v, P_2, u, P_3)$. From Observation 2, it follows that $v$ is not the first vertex of $P$ and thus $P_1 \neq \emptyset$, that is, $P = (P_1', y, v, P_2, u, P_3)$, for some $y \in V(G)$. Notice also that if $P_2 = \emptyset$, then $P = (P_1', y, v, u, P_3)$ and from Lemma 4, $I_u \subseteq I_v$, a contradiction to the assumption that $u, v \in S$ and $\mathcal{I}[S]$ is a proper interval representation of $G[S]$. Thus, $P_2 \neq \emptyset$. Therefore, $P = (P_1', y, v, z, P_2', u, P_3)$, for some $z \in V(G)$. Since $P$ is normal and $u <_\sigma v$ then $y \notin N(u)$. Notice that if we prove that $u <_\sigma y$, then we obtain a contradiction from Lemma 3 and the lemma follows. Thus, it is enough to prove that $u <_\sigma y$.

To prove that $u <_\sigma y$ we claim towards a contradiction that $y <_\sigma u$. Notice then, that as $y \notin N(u)$, it also holds that $r_y < l_u$. However, since $v$ and $u$ are proper intervals and $u <_\sigma v$, $l_u < l_v$. Thus, $r_y < l_v$, a contradiction to the assumption that $yv \in E(G)$ as $y$ is the predecessor of $v$ in $P$. Therefore, $u <_\sigma y$ and this completes the proof of the lemma. □

**Lemma 6.** *Let $G$ be an interval graph and $P = (P_1, u, u', P_2)$ be a normal path of $G$. For every vertex $v \in V(P_2)$, it holds that $u <_\sigma v$ or $u' <_\sigma v$.*

**Proof.** Towards a contradiction we assume that $v <_\sigma u$ and $v <_\sigma u'$. Then from Lemma 3, we obtain that $uv \in E(G)$. Thus, since $u', v \in N(u)$ and $v <_\sigma u'$, it follows by the normality of $P$ that $u'$ is not the next vertex of $u$ in $P$, which is a contradiction. Therefore $u <_\sigma v$ or $u' <_\sigma v$. □

**Lemma 7.** *Let $G$ be an interval graph and $P = (P_1, u, v, w, P_2)$ be a normal path of $G$. If $v <_\sigma u$ then $v <_\sigma w$.*

**Proof.** Let us assume that $v <_\sigma u$ and $w <_\sigma v$. From Lemma 4 it follows that $I_v \subseteq I_u$ and that $I_w \subseteq I_v$. Therefore, $I_w \subseteq I_u$ and thus $w \in N(u)$. This is a contradiction to the assumption that $P$ is normal as $v, w \in N(u)$, $w <_\sigma v$, $v <_P w$, and $v$ is the vertex that follows $u$ in $P$. □

**Lemma 8.** *Let $G$ be an interval graph and $P = (P_1, u, w, P_2, v, P_3)$ be a normal path of $G$ where $v \in N(u)$. If $u <_\sigma w$, then $u <_\sigma x <_\sigma v$, for every vertex $x \in V(P_2) \cup \{w\}$.*

**Proof.** As $w, v \in N(u)$, $w <_P v$, $w$ follows $u$ in $P$, and $P$ is normal we obtain that $u <_\sigma w <_\sigma v$. Thus, it remains to prove that $x <_\sigma v$ for every $x \in V(P_2)$. We prove first that $u <_\sigma x$ for every $x \in V(P_2)$. Assume to the contrary that $x <_\sigma u <_\sigma w$. If $x \in N(u)$ then we obtain a contradiction to the assumption that $P$ is normal. Thus, $x <_\sigma u$, $u <_P x$, and $ux \notin E(G)$. This is again a contradiction to the assumption that $P$ is normal (from Lemma 3). Therefore, $u <_\sigma x$ for every $x \in V(P_2)$ and, since $u <_\sigma w$, $u <_\sigma x$ for every $x \in V(P_2) \cup \{w\}$.

We assume towards a contradiction that there exists a vertex $x' \in V(P_2)$ such that $v <_\sigma x'$. Without loss of generality we also assume that $x'$ is the first such vertex of $P_2$, that is, $x <_\sigma v$ for every vertex $x \in V(P_2)$ with $x <_P x'$. We denote by $z$ the predecessor of $x'$ in $P_2$ or $w$ if $x'$ is the first vertex of $P_2$. Then, as the path is normal, $x' <_P v$, $v <_\sigma x'$, and $zv \notin E(G)$. Then, since $z <_\sigma v$, $r_z < l_v$. However, as $uv \in E(G)$ and $u <_\sigma v$, we obtain that $l_v < r_u$. Thus, $r_z < l_v < r_u$ and, therefore, $z <_\sigma u$, a contradiction. Thus, $x <_\sigma v$ for every vertex $x \in V(P_2) \cup \{w\}$. □

### *2.2. The first data reduction*

Here we present our first data reduction on interval graphs (see Reduction Rule 1). By applying this data reduction to a given interval graph $G$, we obtain a weighted interval graph $G^\#$ with weights on its vertices, such that the maximum weight of a path in $G^\#$ equals the greatest number of vertices of a path in $G$ (cf. Theorem 3). We first introduce the notion of a *reducible* set of vertices and some related properties, which are essential for our Reduction Rule 1.

**Definition 3.** Let $G$ be a (weighted) interval graph and $\mathcal{I}$ be an interval representation of $G$. A set $S \subseteq V(G)$ is *reducible* if it satisfies the following conditions:

1. $\mathcal{I}[S]$ induces a connected proper interval representation of $G[S]$ and
2. for every $v \in V(G)$ such that $I_v \subseteq \mathbf{span}(S)$ it holds $v \in S$.

The intuition behind reducible sets is as follows. For every reducible set $S$, a longest path $P$ contains either all vertices of $S$ or none of them (cf. Lemma 9). Furthermore, in a certain longest path $P$ which contains the whole set $S$, the vertices of $S$ appear *consecutively* in $P$ (cf. Lemma 10). Thus we can reduce the number of vertices in a longest path $P$ (without changing its total weight) by replacing all vertices of $S$ with a single vertex having weight $|S|$, see Reduction Rule 1.

The next two observations will be useful for various technical lemmas in the remainder of the paper. Observation 3 follows by the two conditions of Definition 3 for the reducible sets $S$ in a weighted interval graph $G$. Furthermore, Observation 4 can be easily verified by considering any proper interval representation.

**Observation 3.** Let $G$ be a (weighted) interval graph, $\mathcal{I}$ be an interval representation of $G$, and $S \subseteq V(G)$ be a reducible set. Then, for every $u \in S$ and every $v \in V(G) \setminus \{u\}$, it holds $I_v \nsubseteq I_u$.

**Observation 4.** Let $G$ be a proper interval graph and $\mathcal{I}$ be a proper interval representation of $G$. For every $u, v \in V(G)$:

- If $u <_\sigma v$, then $l_u < l_v$.
- If $u$ and $v$ are consecutive vertices in the ordering $\sigma$ and $G$ is connected, then $uv \in E(G)$.

**Lemma 9.** *Let $G$ be a weighted interval graph with weight function $w : V(G) \to \mathbb{N}$ and let $S$ be a reducible set in $G$. Let also $P$ be a path of maximum weight in $G$. Then either $S \subseteq V(P)$ or $S \cap V(P) = \emptyset$.*

**Proof.** Let $P$ be a path of $G$ of maximum weight and $S$ be a reducible set of $G$. Without loss of generality we may assume by Lemma 2 that $P$ is a normal path. Assume that $S \cap V(P) \neq \emptyset$ and $S \nsubseteq V(P)$. Then there exist two consecutive vertices $u_1, u_2 \in S$ in the vertex ordering $\sigma$ (where $u_1 <_\sigma u_2$) such that either $u_1 \in V(P)$ and $u_2 \notin V(P)$, or $u_1 \notin V(P)$ and $u_2 \in V(P)$. In both cases we will show that we can augment the path $P$ by adding vertex $u_2$ or $u_1$, respectively, which contradicts our maximality assumption on $P$. Since, by Definition 3, $\mathcal{I}[S]$ induces a connected proper interval representation of $G[S]$, it follows by Observation 4 that $u_1u_2 \in E(G)$.

First suppose that $u_1 \in V(P)$ and $u_2 \notin V(P)$. Let $P = (P_1, u_1, P_2)$. Notice first that, if $P_2 = \emptyset$, then the path $P' = (P_1, u_1, u_2) = (P, u_2)$ is a path of $G$ with greater weight than $P$, which is a contradiction to the maximality assumption on $P$. Thus, $P_2 \neq \emptyset$. Let $w \in V(P_2)$ be the first vertex of $P_2$, i.e., $P = (P_1, u_1, w, P_2')$. We show that $u_1 <_\sigma w$. Assume to the contrary that $w <_\sigma u_1$. Then $I_w \subseteq I_{u_1}$ by Lemma 4. This is a contradiction, since $u_1 \in S$ and $S$ is a reducible set (cf. Definition 3). Therefore $u_1 <_\sigma w$. Then either $u_1 <_\sigma u_2 <_\sigma w$ or $u_1 <_\sigma w <_\sigma u_2$. Now we show that $u_2w \in E(G)$. If $u_1 <_\sigma u_2 <_\sigma w$, then Lemma 1 implies that $u_2w \in E(G)$, since $u_1w \in E(G)$. If $u_1 <_\sigma w <_\sigma u_2$ then again Lemma 1 implies that $u_2w \in E(G)$ since $u_1u_2 \in E(G)$. Thus, since $u_2w \in E(G)$, it follows that there exists the path $P' = (P_1, u_1, u_2, w, P_{2'})$ which has greater weight than $P$, which is a contradiction to the maximality assumption on $P$.

Now suppose that $u_1 \notin V(P)$ and $u_2 \in V(P)$. Let then $P = (P_1, u_2, P_2)$. Notice that, if $P_1 = \emptyset$, then the path $P' = (u_1, u_2, P_2) = (u_1, P)$ is a path of $G$ with greater weight than $P$, which is a contradiction. Thus $P_1 \neq \emptyset$. Let $z \in V(P_1)$ be the last vertex of $P_1$, i.e., $P = (P_1', z, u_2, P_2)$. We show that $u_1z \in E(G)$. First let $u_2 <_\sigma z$. Then $I_{u_2} \subseteq I_z$ by Lemma 4, and thus $N(u_2) \subseteq N(z)$. Therefore, since $u_1u_2 \in E(G)$, it follows that $u_1z \in E(G)$ in the case where $u_2 <_\sigma z$. Let now $z <_\sigma u_2$. Suppose that $u_1z \notin E(G)$. Note that $l_{u_2} < r_{u_1}$, since $u_1 <_\sigma u_2$ and $u_1u_2 \in E(G)$. Furthermore, since $S$ is a reducible set, $\mathcal{I}[S]$ induces a proper interval representation of $G[S]$ by Definition 3. Then, since $u_1 <_\sigma u_2$ and $u_1, u_2 \in S$ are consecutive in $\sigma$, Observation 4 implies that $l_{u_1} < l_{u_2}$. That is, $l_{u_1} < l_{u_2} < r_{u_1}$. Hence, since $u_2z \in E(G)$ and $u_1z \notin E(G)$, it follows that $l_{u_1} < l_{u_2} < r_{u_1} < l_z$. Finally $r_z < r_{u_2}$, since $z <_\sigma u_2$ by assumption, and thus $I_z \subseteq I_{u_2}$. This is a contradiction since $u_1 \in S$ and $S$ is a reducible set (cf. Definition 3). Therefore, $zu_1 \in E(G)$ in the case where $z <_\sigma u_2$. That is, we always have $zu_1 \in E(G)$. Therefore there exists the path $P' = (P_1, u_1, u_2, w, P_{2'})$ which has greater weight than $P$, which is a contradiction to the maximality assumption on $P$. □

**Lemma 10.** *Let $G$ be a weighted interval graph with weight function $w : V(G) \to \mathbb{N}$, and $S$ be a reducible set in $G$. Let also $P$ be a path of maximum weight in $G$ and let $S \subseteq V(P)$. Then there exists a path $P'$ of $G$ such that $V(P') = V(P)$ and the vertices of $S$ appear consecutively in $P'$.*

**Proof.** Let $P$ be a path of maximum weight of $G$ such that $S \subseteq V(P)$. Denote $S = \{u_1, u_2, \ldots, u_{|S|}\}$, where $u_1 <_\sigma u_2 <_\sigma \cdots <_\sigma u_{|S|}$. Without loss of generality we may assume by Lemma 2 that $P$ is normal. Since $\mathcal{I}[S]$ induces a proper interval representation of $G[S]$ (cf. Definition 3) Lemma 5 implies that also $u_1 <_P u_2 <_P \cdots <_P u_{|S|}$, i.e., the vertices of $S$ appear in the same order both in the vertex ordering $\sigma$ and in the path $P$. Furthermore, Observation 4 implies that $u_iu_{i+1} \in E(G)$ for every $i \in [|S|-1]$. Let $P = (P_0, u_1, P_1, u_2, \ldots, P_{|S|-1}, u_{|S|}, P_{|S|})$, where

$$V(P_i) \cap S = \emptyset, \text{ for every } i \in [|S|-1]. \tag{1}$$

For every $i \in [|S|-1]$, we denote

$$z_i = \begin{cases} \text{the first vertex of } P_i & \text{if } P_i \neq \emptyset \\ u_{i+1} & \text{otherwise} \end{cases}.$$

Let $i \in [|S|-1]$ and suppose that $z_i <_\sigma u_i$. If $z_i = u_{i+1}$ then $u_i <_\sigma z_i = u_{i+1}$, which is a contradiction. Therefore $z_i \neq u_{i+1}$. That is, $z_i$ is the first vertex of $P_i$, i.e., the successor of $u_i$ in $P$. Then, since we assumed that $z_i <_\sigma u_i$, it follows by Lemma 4 that $I_{z_i} \subseteq I_{u_i}$. Thus $z_i \in S$, since $u_i \in S$ and $S$ is a reducible set (cf. Definition 3). This is a contradiction to Eq. (1). Therefore $u_i <_\sigma z_i$, for every $i \in [|S|-1]$.

Recall that $\mathcal{I}[S]$ induces a proper interval representation of $G[S]$ and that $u_1 <_\sigma u_2 <_\sigma \cdots <_\sigma u_{|S|}$. Thus $l_{u_1} = \min\{l_u : u \in S\}$ and $r_{u_{|S|}} = \max\{r_u \ u \in S\}$, i.e., $\mathbf{span}(S) = [l_{u_1}, r_{u_{|S|}}]$. Now let $i \in [|S|-1]$. Since $u_iu_{i+1} \in E(G)$ and $u_i <_\sigma z_i$, Lemma 8 implies that $u_i <_\sigma z <_\sigma u_{i+1}$ for every $z \in V(P_i)$. Therefore $u_1 <_\sigma z <_\sigma u_{|S|}$, for every $z \in V(P_i)$, where $i \in [|S|-1]$. Suppose that there exists a vertex $z \in V(P_i)$ such that $l_{u_1} < l_z$. Then, since $z <_\sigma u_{|S|}$, it follows that $I_z \subseteq \mathbf{span}(S)$. Thus $z \in S$, since $S$ is a reducible set by assumption (cf. Definition 3). This is a contradiction to Eq. (1). Thus $l_z < l_{u_1}$ for every $z \in V(P_i)$, where $i \in [|S|-1]$. Since also $u_1 <_\sigma z$ as we proved above, it follows that $I_{u_1} \subseteq I_z$ for every $z \in V(P_i)$, where $i \in [|S|-1]$. Thus $N(u_1) \subseteq N(z)$ for every $z \in V(P_i)$, where $i \in [|S|-1]$. Therefore $P' = (P_0, P_1, \ldots, P_{|S|-1}, u_1, u_2, \ldots, u_{|S|}, P_{|S|})$ is a path of $G$, where $V(P') = V(P)$ and the vertices of $S$ appear consecutively in $P'$. This completes the proof of the lemma. □

We now present two auxiliary technical lemmas that will be used to prove the correctness of Reduction Rule 1 in Theorem 3.

**Lemma 11.** *Let $G$ be an interval graph and $\mathcal{I}$ be an interval representation of $G$. Let also $S, S' \subseteq V(G)$ be two reducible sets of $G$ such that $S \cap S' = \emptyset$. Let $G'$ be the graph obtained from $G$ by replacing $\mathcal{I}[S]$ by $\mathbf{span}(S)$. Then $S'$ remains a reducible set of $G'$.*

**Proof.** Let $u$ be the vertex of $V(G') \setminus V(G)$, i.e., $I_u = \mathbf{span}(S)$. Denote by $\mathcal{I}'$ the interval representation obtained from $\mathcal{I}$ after replacing $\mathcal{I}[S]$ by $\mathbf{span}(S)$. First note that $\mathcal{I}'[S'] = \mathcal{I}[S']$, and thus $\mathcal{I}'[S']$ induces a connected proper interval representation as $S'$ is a reducible set by assumption. This proves Condition 1 of Definition 3.

Let now $v \in V(G')$ such that $I_v \subseteq \mathbf{span}(S')$. Assume that $v \notin S'$. If $v \neq u$, then $v \in V(G)$ and thus $v \in S'$, since $S'$ is a reducible set of $G$. If $v = u$, then $\mathbf{span}(S) = I_u = I_v \subseteq \mathbf{span}(S')$. That is, for every $u_0 \in S$ we have $I_{u_0} \subseteq \mathbf{span}(S) \subseteq \mathbf{span}(S')$, and thus also $u_0 \in S'$, since $S'$ is a reducible set. This is a contradiction, since $S \cap S' = \emptyset$. Therefore, for every $v \in V(G')$ such that $I_v \subseteq \mathbf{span}(S')$, we have that $v \in S'$. This proves Condition 2 of Definition 3 and completes the proof of the lemma. □

**Lemma 12.** *Let $\ell$ be a positive integer. Let $G$ be a weighted interval graph, $w : V(G) \to \mathbb{N}$, $\mathcal{I}$ be an interval representation of $G$, and $S$ be a reducible set in $G$. Let also $G'$ be the graph obtained from $G$ by replacing $\mathcal{I}[S]$ by an interval $I_u = \mathbf{span}(S)$ where $w(u) = \sum_{v \in S} w(v)$. Then the maximum weight of a path in $G$ is $\ell$ if and only if the maximum weight of a path in $G'$ is $\ell$.*

**Proof.** First assume that the maximum weight of a path $P$ in $G$ is $\ell$. Without loss of generality, from Lemma 2, we may also assume that $P$ is normal. Furthermore, either $S \subseteq V(P)$ or $S \cap V(P) = \emptyset$ by Lemma 9. Notice that if $S \cap V(P) = \emptyset$, then $P$ is also a path of $G'$. Suppose that $S \subseteq V(P)$. Then from Lemma 10, we can obtain a path $\widehat{P}$ such that $V(\widehat{P}) = V(P)$

and the vertices of $S$ appear consecutively in $\widehat{P}$. Notice then that by replacing the subpath of $\widehat{P}$ consisting of the vertices of $S$ by the single vertex $u$, we obtain a path $P'$ of $G'$ such that $V(P') \setminus V(P) = \{u\}$ and $V(P') \cap V(P) = V(P) \setminus \{S\}$. Since $w(u) = \sum_{v \in S} w(v)$, we obtain that $\sum_{v \in V(P')} w(v) = \sum_{v \in V(P)} w(v)$. Thus, $G'$ has a path $P'$ of weight at least $\ell$.

Now assume that the maximum weight of a path $P'$ in $G'$ is $\ell$. If $u \notin V(P')$, then $V(P')$ is also a path of $G$. Suppose that $u \in V(P')$. Let $P' = (P_1, v, u, v', P_2)$. Our aim is to show that $u_{|S|}v' \in E(G)$ and $vu_1 \in E(G)$. Suppose that $v' <_\sigma u$. Then Lemma 4 implies that $I_{v'} \subseteq I_u = \mathbf{span}(S)$, and thus $v' \in S = V(G) \setminus V(G')$. This is a contradiction, since $v' \in V(G')$. Thus $u <_\sigma v'$, i.e., $r_u = r_{u_{|S|}} < r_{v'}$. Since $uv' \in E(G)$, it follows that $l_{v'} < r_u = r_{u_{|S|}}$. Therefore $r_{u_{|S|}} \in I_{v'}$, and thus $v'u_{|S|} \in E(G)$. It remains to show that $vu_1 \in E(G)$. First let $u <_\sigma v$. Then $I_u \subseteq I_v$ by Lemma 4. Furthermore, since $I_{u_1} \subseteq I_u = \mathbf{span}(S)$, it follows that $I_{u_1} \subseteq I_v$, and thus $vu_1 \in E(G)$. Let now $v <_\sigma u$, i.e., $r_v < r_u = r_{u_{|S|}}$. Then, since $vu \in E(G)$, it follows that $l_{u_1} = l_u < r_v < r_u$. If $I_v \subseteq I_u = \mathbf{span}(S)$, then $v' \in S = V(G) \setminus V(G')$, which is a contradiction since $v \in V(G')$. Thus $I_v \nsubseteq I_u$. Therefore, since $l_u < r_v < r_u$, it follows that $l_v < l_u = l_{u_1} < r_v$, i.e., $l_{u_1} \in I_v$. Thus, $u_1 v \in E(G)$. This implies that we may obtain a path $P$ of $G$ by replacing the vertex $u$ in $P'$ by the path $(u_1, u_2, \ldots, u_{|S|})$. As before, since $w(u) = \sum_{v \in S} w(v)$, the weight of $P$ is equal to the weight of $P'$. Hence, $G$ has a path of weight at least $\ell$. $\square$

In the next definition we reduce the interval graph $G$ to the *weighted* interval graph $G^\#$ which has fewer vertices than $G$. Then, as we prove in Theorem 3, the longest paths of $G$ correspond to the maximum-weight paths of $G^\#$.

**Reduction Rule 1** *(first data reduction). Let $G = (V, E)$ be an interval graph, $\mathcal{I}$ be an interval representation of $G$, and $D$ be a proper interval deletion set of $G$. Let $\mathcal{S}$ be a set of vertex disjoint reducible sets of $G$, where $S \cap D = \emptyset$, for every $S \in \mathcal{S}$. The* weighted interval graph *$G^\# = (V^\#, E^\#)$ is induced by the* weighted interval representation *$\mathcal{I}^\#$, which is derived from $\mathcal{I}$ as follows:*

- *for every $S \in \mathcal{S}$, replace in $\mathcal{I}$ the intervals $\{I_v : v \in S\}$ with the single interval $I_S = \mathbf{span}(S)$ which has weight $|S|$; all other intervals receive weight* 1.

In the next theorem we prove the correctness of Reduction Rule 1.

**Theorem 3.** *Let $\ell$ be a positive integer. Let $G$ be an interval graph and $G^\#$ be the weighted interval graph derived by Reduction Rule 1. Then the longest path in $G$ has $\ell$ vertices if and only if the maximum weight of a path in $G^\#$ is $\ell$.*

**Proof.** The construction of the graph $G^\#$ from $G$ in Reduction Rule 1 can be done sequentially, replacing each time the intervals of a set $S \in \mathcal{S}$ with one interval of the appropriate weight. After making this replacement for such a set $S \in \mathcal{S}$, the maximum weight of a path in the resulting graph is by Lemma 12 equal to the maximum weight of a path in the graph before the replacement of $S$. Furthermore, the vertex set of every other set $S' \in \mathcal{S} \setminus S$ remains reducible in the resulting graph by Lemma 11. Therefore we can iteratively replace all sets of $\mathcal{S}$, resulting eventually in the weighted graph $G^\#$, in which the maximum weight of a path is equal to the maximum number of vertices in a path in the original graph $G$. $\square$

In the next lemma we prove that the weighted interval graph $G^\#$, which is obtained from an interval graph $G$ by applying Reduction Rule 1 to it, has some useful properties. These properties will be exploited in Section 3 (cf. Corollary 2 in Section 3) as they are crucial for deriving a *special weighted interval graph* (cf. Definition 5 in Section 3).

**Lemma 13.** *Let $G$ be an interval graph, $D$ be a proper interval deletion set of $G$, and $\mathcal{S}$ be a set of vertex-disjoint reducible sets of $G$, where $S \cap D = \emptyset$, for every $S \in \mathcal{S}$. Suppose that $\mathbf{span}(S) \cap \mathbf{span}(S') = \emptyset$ for every two distinct sets $S, S' \in \mathcal{S}$. Furthermore, let $G^\#$ be the weighted interval graph obtained from Reduction Rule 1. Then $D$ is a proper interval deletion set of $G^\#$. Furthermore $V(G^\#) \setminus D$ can be partitioned into two sets $A$ and $U^\#$, where $A = V(G^\#) \setminus V(G)$ and:*

1. *$A$ is an independent set of $G^\#$, and*
2. *for every $v \in A$ and every $u \in V(G^\#) \setminus \{v\}$, we have $I_u \nsubseteq I_v$.*

**Proof.** Define $A = V(G^\#) \setminus V(G)$ and $U^\# = V(G^\#) \setminus (D \cup A)$. Note that the sets $A$ and $U^\#$ form a partition of the set $V(G^\#) \setminus D$. First we prove that $A$ is an independent set. Note by Reduction Rule 1 that every vertex $a \in A$ corresponds to a different set $S_a \in \mathcal{S}$, which corresponds to the interval $I_a = \mathbf{span}(S)$ in the interval representation $\mathcal{I}^\#$ of the graph $G^\#$. Therefore, since $\mathbf{span}(S) \cap \mathbf{span}(S') = \emptyset$ for every two distinct sets $S, S' \in \mathcal{S}$, it follows that $I_a \cap I_{a'} = \emptyset$ for every two distinct vertices $a, a' \in A$. That is, $A$ induces an independent set in $G^\#$.

Second we prove that for every $v \in A$ and every $u \in V(G^\#) \setminus \{v\}$, we have $I_u \nsubseteq I_v$. Suppose otherwise that there exist $v \in A$ and $u \in V(G^\#) \setminus \{v\}$ such that $I_u \subseteq I_v$. Then, since $A$ is an independent set, it follows that $u \notin A$, i.e., $u \in V(G)$. Recall by the construction that $I_v = \mathbf{span}(S)$, for some reducible set $S$ in $G$. Thus, since $I_u \subseteq I_v = \mathbf{span}(S)$, it follows by Definition 3 that $u \in S$. This is a contradiction since by construction $V(G^\#) \cap S = \emptyset$. Therefore, for every $v \in A$ and every $u \in V(G^\#) \setminus \{v\}$, we have $I_u \nsubseteq I_v$.

Now we prove that $D$ is a proper interval deletion set of $G^\#$. Suppose otherwise that $G^\# \setminus D$ is not a proper interval graph. Then $G^\# \setminus D$ contains an induced $K_{1,3}$ [48]. If this $K_{1,3}$ contains no vertex of $A$, then this $K_{1,3}$ is also contained as an induced subgraph of $G \setminus D$. This is a contradiction, since $D$ is a proper interval deletion set of $G$ by assumption. Thus this $K_{1,3}$ contains at least one vertex of $A$. Let $v$ denote the vertex of degree 3 in this $K_{1,3}$. Suppose that $v \in A$. Then, for at least one vertex $u$ of the other three vertices of this $K_{1,3}$ we have that $I_u \subseteq I_v$, which is a contradiction as we proved above. Suppose that $v \notin A$, i.e., $v \in U^{\#}$. Denote the leaves of the $K_{1,3}$ by $v_1, v_2, v_3$. Let $i \in [3]$. If $v_i \in U^{\#}$, then $v_i \in V(G)$. Otherwise, if $v_i \in A$, then there exists a reducible set $S_i$ in $G$ such that $I_{v_i} = \mathbf{span}(S_i)$. Furthermore, since $v \in U^{\#}$ and $I_v \cap \mathbf{span}(S_i) \neq \emptyset$, there exists at least one vertex $u_i \in S_i$ where $u_i v \in E(G)$. For every $i \in [3]$, define

$$z_i = \begin{cases} u_i & \text{if } v_i \in A, \\ v_i & \text{otherwise.} \end{cases}$$

Observe that $v, z_1, z_2, z_3$ belong to the original graph $G$, i.e., $v, z_1, z_2, z_3 \in V(G) \setminus D$. Since $I_{v_i} \cap I_{v_j} = \emptyset$ for every $1 \leq i < j \leq 3$ and $I_{z_i} \subseteq I_{v_i}$, $i \in [3]$ it follows that $I_{z_i} \cap I_{z_j} = \emptyset$, for every $1 \leq i < j \leq 3$. Therefore, $\{z_1, z_2, z_3\}$ is an independent set in $G \setminus D$. Moreover $vz_i \in E(G)$, for every $i \in [3]$. Thus, $\{v, z_1, z_2, z_3\}$ induces a $K_{1,3}$ in $G \setminus D$. This is a contradiction, since $D$ is a proper interval deletion set of $G$ by assumption. Therefore $D$ is a proper interval deletion set of $G^{\#}$. $\square$

### 2.3. The second data reduction

Here we present our second data reduction, which is applied to an arbitrary weighted interval graph $G$ with weights on its vertices (cf. Reduction Rule 2). As we prove in Theorem 4, the maximum weight of a path in the resulting weighted interval graph $\widehat{G}$ is the same as the maximum weight of a path in $G$. We first introduce the notion of a *weakly reducible* set of vertices and some related properties, which are needed for our Reduction Rule 2.

**Definition 4.** Let $G$ be a (weighted) interval graph and $\mathcal{I}$ be an interval representation of $G$. A set $S \subseteq V(G)$ is *weakly reducible* if it satisfies the following conditions:

1. $\mathcal{I}[S]$ induces a connected proper interval representation of $G[S]$ and
2. for every $v \in V(G)$ and every $u \in S$, if $I_v \subseteq I_u$ then $S \subseteq N(v)$.

Note here that Condition 2 of Definition 4 also applies to the case where $v = u$. Therefore $S \subseteq N(u)$ for every $u \in S$, i.e., $S$ induces a clique, as the next observation states.

**Observation 5.** Let $G$ be a (weighted) interval graph, $\mathcal{I}$ be an interval representation of $G$ and $S$ be a weakly reducible set in $G$. Then $G[S]$ is a clique. Furthermore $[\max\{l_v : v \in S\}, \min\{r_v : v \in S\}] \subseteq I_u$, for every $u \in S$.

The intuition behind weakly reducible sets is as follows. For every weakly reducible set $S$, a longest path $P$ contains either all vertices of $S$ or none of them (cf. Lemma 14). Furthermore let $D$ be a given proper interval deletion set of $G$. Then, in a certain path $P$ of maximum weight which contains the whole set $S$, the appearance of the vertices of $S$ in $P$ is *interrupted at most* $|D| + 3$ times by vertices outside $S$. That is, such a path $P$ has at most $\min\{|S|, |D| + 4\}$ vertex-maximal subpaths with vertices from $S$ (cf. Lemma 16). Thus we can reduce the number of vertices in a maximum-weight normal path $P$ (without changing its total weight) by replacing all vertices of $S$ with $\min\{|S|, |D| + 4\}$ vertices, cf. the Reduction Rule 2; each of these new vertices has the same weight and their total weight sums up to $|S|$.

**Lemma 14.** *Let $G$ be a weighted interval graph with weight function $w : V(G) \to \mathbb{N}$ and let $S$ be a weakly reducible set in $G$. Let also $P$ be a path of maximum weight in $G$. Then either $S \subseteq V(P)$ or $S \cap V(P) = \emptyset$.*

**Proof.** Let $P$ be a path of $G$ of maximum weight and $S$ be a weakly reducible set in $G$. Without loss of generality we also assume that $P$ is a normal path (Lemma 2). Suppose towards a contradiction that $S \cap V(P) \neq \emptyset$ and $S \nsubseteq V(P)$. Then there exist two consecutive vertices $u_1, u_2 \in S$ in the vertex ordering $\sigma$ (where $u_1 <_\sigma u_2$) such that either $u_1 \in V(P)$ and $u_2 \notin V(P)$, or $u_1 \notin V(P)$ and $u_2 \in V(P)$. In both cases we will show that we can augment the path $P$ by adding vertex $u_2$ or $u_1$, respectively, which contradicts our maximality assumption on $P$. From Observation 5, $S$ induces a clique in $G$. Hence, $u_1 u_2 \in E(G)$.

First suppose that $u_1 \in V(P)$ and $u_2 \notin V(P)$. Let $P = (P_1, u_1, P_2)$. Notice first that, if $P_2 = \emptyset$, then the path $P' = (P_1, u_1, u_2) = (P, u_2)$ is a path of $G$ with greater weight than $P$, which is a contradiction to the maximality assumption on $P$. Thus, $P_2 \neq \emptyset$. Let $w \in V(P_2)$ be the first vertex of $P_2$, i.e., $P = (P_1, u_1, w, P_2')$. We prove that $u_2 w \in E(G)$. For this, notice first that either $w <_\sigma u_1$ or $u_1 <_\sigma w$. If $w <_\sigma u_1$, then $I_w \subseteq I_{u_1}$ by Lemma 4. Since $u_1, u_2$ are vertices of the weakly reducible set $S$ and $I_w \subseteq I_{u_1}$ then $u_2 w \in E(G)$ (Definition 4). If $u_1 <_\sigma w$, then either $u_1 <_\sigma u_2 <_\sigma w$ or $u_1 <_\sigma w <_\sigma u_2$. If $u_1 <_\sigma u_2 <_\sigma w$, then Lemma 1 implies that $u_2 w \in E(G)$ since $u_1 w \in E(G)$. If $u_1 <_\sigma w <_\sigma u_2$, then again Lemma 1 implies that $u_2 w \in E(G)$, since $u_1 u_2 \in E(G)$. This completes the argument that $u_2 w \in E(G)$. Since $u_2 w \in E(G)$, it follows that there exists the path $P' = (P_1, u_1, u_2, w, P_2')$ which has greater weight than $P$, which is a contradiction to the maximality assumption on $P$.

Now suppose that $u_1 \notin V(P)$ and $u_2 \in V(P)$. Let then $P = (P_1, u_2, P_2)$. Notice that, if $P_1 = \emptyset$, then the path $P' = (u_1, u_2, P_2) = (u_1, P)$ is a path of $G$ with greater weight than $P$, which is a contradiction. Thus $P_1 \neq \emptyset$. Let $z \in V(P_1)$ be the last vertex of $P_1$, i.e., $P = (P'_1, z, u_2, P_2)$. We show that $u_1 z \in E(G)$. First let $u_2 <_\sigma z$. Then $I_{u_2} \subseteq I_z$ by Lemma 4, and thus $N(u_2) \subseteq N(z)$. Therefore, since $u_1 u_2 \in E(G)$, it follows that $u_1 z \in E(G)$ in the case where $u_2 <_\sigma z$. Now let $z <_\sigma u_2$. Suppose that $u_1 z \notin E(G)$. Note that $l_{u_2} < r_{u_1}$ since $u_1 <_\sigma u_2$ and $u_1 u_2 \in E(G)$. Furthermore, since $S$ is a weakly reducible set, $\mathcal{I}[S]$ induces a proper interval representation of $G[S]$ by Definition 4. Then, since $u_1 <_\sigma u_2$ and $u_1, u_2 \in S$ are consecutive in $\sigma$, Observation 4 implies that $l_{u_1} < l_{u_2}$. That is, $l_{u_1} < l_{u_2} < r_{u_1}$. Hence, since $u_2 z \in E(G)$ and $u_1 z \notin E(G)$, it follows that $l_{u_1} < l_{u_2} < r_{u_1} < l_z$. Finally $r_z < r_{u_2}$, since $z <_\sigma u_2$ by assumption, and thus $I_z \subseteq I_{u_2}$. Then, since $u_1, u_2$ are vertices of the weakly reducible set $S$ and $I_z \subseteq I_{u_2}$, it follows by Definition 4 that $u_1 z \in E(G)$, a contradiction. Therefore $u_1 z \in E(G)$ in the case where $z <_\sigma u_2$. That is, always $z u_1 \in E(G)$. Hence, there exists the path $P' = (P_1, u_1, u_2, w, P_{2'})$ which has greater weight than $P$, which is a contradiction to the maximality assumption on $P$. □

We are now ready to present our second data reduction. As we prove in Theorem 4, this data reduction maintains the maximum weight of a path.

**Reduction Rule 2** *(second data reduction). Let $G$ be a weighted interval graph with weight function $w : V(G) \to \mathbb{N}$ and $\mathcal{I}$ be an interval representation of $G$. Let $D$ be a proper interval deletion set of $G$. Finally, let $\mathcal{S} = \{S_1, S_2, \dots, S_{|\mathcal{S}|}\}$ be a family of pairwise disjoint weakly reducible sets, where $S_i \cap D = \emptyset$ for every $i \in [|\mathcal{S}|]$. We recursively define the graphs $G_0, G_1, \dots, G_{|\mathcal{S}|}$ with the interval representations $\mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_{|\mathcal{S}|}$ as follows:*

- *$G_0 = G$ and $\mathcal{I}_0 = \mathcal{I}$,*
- *for $1 \le i \le |\mathcal{S}|$, $\mathcal{I}_i$ is obtained by replacing in $\mathcal{I}_{i-1}$ the intervals $\{I_v : v \in S_i\}$ with $\min\{|S_i|, |D| + 4\}$ copies of the interval $I_{S_i} = \mathbf{span}(S_i)$, each having equal weight $\frac{1}{\min\{|S_i|,|D|+4\}} \sum_{u \in S_i} w(u)$; all other intervals remain unchanged, and*
- *finally $\widehat{G} = G_{|\mathcal{S}|}$ and $\widehat{\mathcal{I}} = \mathcal{I}_{|\mathcal{S}|}$.*

Note that in the construction of the interval representation $\mathcal{I}_i$ by Reduction Rule 2, where $i \in [|\mathcal{S}|]$, we can always slightly perturb the endpoints of the $\min\{|S_i|, |D| + 4\}$ copies of the interval $I_{S_i} = \mathbf{span}(S_i)$ such that all endpoints remain distinct in $\mathcal{I}_i$, and such that these $\min\{|S_i|, |D| + 4\}$ newly introduced intervals induce a proper interval representation in $\mathcal{I}_i$.

Next we prove in Lemma 16 that, for every weakly reducible set $S$, every maximum-weight path $P$ which contains the whole set $S$ can be rewritten as a path $P'$, where the appearance of the vertices of $S$ in $P$ is *interrupted at most* $|D| + 3$ times by vertices outside $S$. That is, such a path $P'$ has at most $\min\{|S|, |D| + 4\}$ vertex-maximal subpaths with vertices from $S$. Before we prove Lemma 16, we first provide the following auxiliary lemma.

**Lemma 15.** *Let $G$ be an interval graph and $\mathcal{I}$ be an interval representation of $G$. Let also $S, S' \subseteq V(G)$ such that $S$ is a weakly reducible set of $G$ and $S \cap S' = \emptyset$. Let $G'$ be the graph obtained from $G$ by replacing $\mathcal{I}[S']$ by $\mathbf{span}(S')$. Then $S$ remains a weakly reducible set of $G'$.*

**Proof.** Let $u$ be the vertex of $V(G') \setminus V(G)$, i.e., $I_u = \mathbf{span}(S')$. Denote by $\mathcal{I}'$ the interval representation obtained from $\mathcal{I}$ after replacing $\mathcal{I}[S']$ by $\mathbf{span}(S')$. First note that $\mathcal{I}'[S] = \mathcal{I}[S]$, and thus $\mathcal{I}'[S]$ induces a connected proper interval representation as $S$ is a weakly reducible set by assumption. This proves Condition 1 of Definition 4.

Let now $x \in V(G')$ and $v \in S$. Assume that $I_x \subseteq I_v$. If $x \neq u$, then $x \in V(G)$, and thus $I_x \subseteq I_v$ in $\mathcal{I}$. Therefore, since $S$ is a weakly reducible set of $G$ by assumption, it follows that $S \subseteq N(x)$. If $x = u$, then $I_x = I_u \subseteq I_v$ in $\mathcal{I}'$. Thus, since $I_u = \mathbf{span}(S')$, it follows that for every vertex $x' \in S'$, we have $I_{x'} \subseteq I_u \subseteq I_v$. Thus, since $S$ is a weakly reducible set of $G$, it follows that $S \subseteq N(x')$, i.e., $I_{x'} \cap I_w \neq \emptyset$ for every $w \in S$. Therefore, since $I_{x'} \subseteq I_u$, it follows that also $I_u \cap I_w \neq \emptyset$ for every $w \in S$, i.e., $S \subseteq N(u) = N(x)$. This proves Condition 2 of Definition 4. □

The next observation follows directly by the definition of the graphs $G_0, G_1, \dots, G_{|\mathcal{S}|}$ (see the Reduction Rule 2) and by Lemma 15.

**Observation 6.** For every $i \in [|\mathcal{S}|]$, the sets $S_i, \dots, S_{|\mathcal{S}|}$ are weakly reducible sets of $G_{i-1}$.

**Lemma 16.** *Let $G$ be a weighted interval graph with weight function $w : V(G) \to \mathbb{N}$ and $\mathcal{I}$ be an interval representation of $G$. Let $D$ be a proper interval deletion set of $G$ and $\mathcal{S} = \{S_1, S_2, \dots, S_{|\mathcal{S}|}\}$ be a family of pairwise disjoint weakly reducible sets of $G$ such that $S_i \cap D = \emptyset$, $i \in [|\mathcal{S}|]$. Also, let $i \in [|\mathcal{S}|]$ and $G_{i-1}$ be one of the graphs obtained by Reduction Rule 2. Finally, let $P$ be a path of maximum weight in $G_{i-1}$ and let $S_i \subseteq V(P)$. Then there exists a path $P'$ in $G_{i-1}$ on the same vertices as $P$, which has at most $\min\{|S_i|, |D| + 4\}$ vertex-maximal subpaths consisting only of vertices from $S_i$.*

**Proof.** Let $P$ be a path of maximum weight in $G_{i-1}$ such that $S_i \subseteq V(P)$. Denote $S_i = \{u_1, u_2, \dots, u_{|S_i|}\}$, where $u_1 <_\sigma u_2 <_\sigma \cdots <_\sigma u_{|S_i|}$. Without loss of generality we may assume by Lemma 2 that $P$ is normal. Observation 6 yields that

$S_i$ is a weakly reducible set of $G_{i-1}$. Thus, since $\mathcal{I}[S_i]$ induces a proper interval representation of $G_{i-1}[S_i]$ (cf. Definition 4), Lemma 5 implies that $u_1 <_P u_2 <_P \cdots <_P u_{|S_i|}$, i.e., the vertices of $S_i$ appear in the same order both in the vertex ordering $\sigma$ and in the path $P$. Furthermore, Observation 5 implies that $u_w u_{w+1} \in E(G_{i-1})$, for every $w \in [|S_i|-1]$. Let $P = (P_0, u_1, P_1, u_2, \ldots, P_{|S_i|-1}, u_{|S_i|}, P_{|S_i|})$. For every $w \in [|S_i|-1]$, we denote

$$z_w = \begin{cases} \text{the first vertex of } P_w & \text{if } P_w \neq \emptyset \\ u_{w+1} & \text{otherwise} \end{cases}.$$

Let

$$Z_1 = \{z_w : z_w <_\sigma u_w, w \in [|S_i|-1]\},$$
$$Z_2 = \{z_w : u_w <_\sigma z_w, w \in [|S_i|-1]\}.$$

Note that, as $u_w <_\sigma u_{w+1}$, for every $w \in [|S_i|-1]$ it follows that $Z_1 \cap S_i = \emptyset$.

In the first part of the proof we show that $|Z_1| \leq |D|+2$. First we show that for every $z \in Z_1$, $S_i \subseteq N(z)$. Let $w \in [|S_i|-1]$, such that $z_w \in Z_1$, i.e., $z_w <_\sigma u_w$. Then from Lemma 4, $I_{z_w} \subseteq I_{u_w}$. Therefore, since $u_w$ is a vertex of the weakly reducible set of $S_i$ in $G_{i-1}$, it follows by Definition 4 that $S_i \subseteq N(z_w)$ in the graph $G_{i-1}$. Thus $S_i \subseteq N(z)$ for every $z \in Z_1$.

We now prove that $Z_1$ is an independent set. Towards a contradiction we assume that there exist $z_w, z_j \in Z_1$ with $z_w <_P z_j$ and $z_w z_j \in E(G_{i-1})$. Then, since $u_w z_w, u_w z_j \in E(G_{i-1})$ and $z_w$ is the successor of $u_w$ in the normal path $P$, it follows that $z_w <_\sigma z_j$, i.e., $z_w$ and $z_j$ appear in the same order in the vertex ordering $\sigma$ and in the path $P$. Let $v$ be the successor of $z_w$ in $P$. Then since $z_w <_\sigma u_w$, Lemma 7 implies that $z_w <_\sigma v$. Furthermore, as $z_w <_\sigma z_j$, from Lemma 8 we obtain that $z_w <_\sigma u <_\sigma z_j$, for every $u \in V(P)$ such that $z_w <_P u <_P z_j$. Since $u_j$ is the predecessor of $z_j$ in $P$, it follows that $z_w <_P u_j <_P z_j$. Therefore $u_j <_\sigma z_j$, a contradiction as $z_j \in Z_1$. Therefore $Z_1$ is an independent set.

Assume now (towards a contradiction) that $|Z_1| \geq |D|+3$. Then $|Z_1 \setminus D| \geq |Z_1| - |D| \geq 3$. Hence, there exist at least 3 vertices $z', z'', z''' \in Z_1 \setminus D$. Let $u \in S_i$. Note that $u \notin D$, since $S_i \cap D = \emptyset$. Then $u, z', z'', z''' \in V(G_{i-1} \setminus D)$. Moreover, recall that $S_i \subseteq N(z)$, for every $z \in Z_1$. Thus $uz', uz'', uz''' \in E(G_{i-1} \setminus D)$. Recall also that $Z_1$ is an independent set and thus the vertices $z', z'', z'''$ form an independent set in $G_{i-1} \setminus D$. Thus, $\{u, z', z'', z'''\}$ induce a $K_{1,3}$ in $G_{i-1} \setminus D$. Notice that by construction of $G_{i-1}$ (cf. Reduction Rule 2) each one of the vertices $z', z'', z'''$ is either a vertex of $G$, or it appears in $G_{i-1}$ as the replacement of some weakly reducible set $S$. If $z' \notin V(G)$, then denote by $S'$ the weakly reducible set corresponding to vertex $z'$. Otherwise, if $z' \in V(G)$, we define $S' = \{z'\}$. Similarly, if $z'' \notin V(G)$ (resp. if $z''' \notin V(G)$) then denote by $S''$ (resp. by $S'''$) the weakly reducible set corresponding to vertex $z''$ (resp. $z'''$). Otherwise, if $z'' \in V(G)$ (resp. if $z''' \in V(G)$), we define $S'' = \{z''\}$ (resp. $S''' = \{z'''\}$). Since $z'u \in E(G_{i-1})$, observe that there always exists at least one vertex $v' \in S'$ such that $I_{v'} \cap I_u \neq \emptyset$, and thus $uv' \in E(G)$. Similarly, since $z''u, z'''u \in E(G_{i-1})$, there always exist vertices $v'' \in S''$ and $v''' \in S'''$ such that $I_{v''} \cap I_u \neq \emptyset$ and $I_{v'''} \cap I_u \neq \emptyset$, and thus $uv'', uv''' \in E(G)$. Note that $v', v'', v''' \in V(G) \setminus D$. Furthermore, by the definition of $v', v'', v'''$, it follows that $I_{v'} \subseteq I_{z'}$, $I_{v''} \subseteq I_{z''}$, and $I_{v'''} \subseteq I_{z'''}$. Therefore, since $z', z'', z'''$ form an independent set, the vertices $v', v'', v'''$ also form an independent set. Furthermore, as $uv', uv'', uv''' \in E(G)$, it follows that $\{u, v', v'', v'''\}$ induce a $K_{1,3}$ in $G \setminus D$, which is a contradiction to the assumption that $D$ is a proper interval deletion set of $G$. Thus $|Z_1| \leq |D|+2$. This completes the first part of our proof.

Let now $z_{i_0}$ be the vertex of $Z_2$ that appears first in $P$. In the second part of the proof we show that the set $\{x \in V(G_{i-1}) : u_{i_0} <_P x <_P u_{|S_i|}\}$ induces a clique in $G_{i-1}$. Note first that, since $u_{i_0} <_\sigma z_{i_0} \leq_\sigma u_{|S_i|}$ and $u_{i_0} u_{|S_i|} \in E(G_{i-1})$, Lemma 8 implies that $u_{i_0} <_\sigma x <_\sigma u_{|S_i|}$, for every $x \in V(G_{i-1})$ such that $u_{i_0} <_P x <_P u_{|S_i|}$. Therefore, since $u_{i_0} u_{|S_i|} \in E(G_{i-1})$, it follows by Lemma 1 that $x u_{|S_i|} \in E(G_{i-1})$ for every $x \in V(G_{i-1})$ such that $u_{i_0} <_\sigma x <_\sigma u_{|S_i|}$.

We now prove that $u_{i_0} x \in E(G_{i-1})$, for every $x \in V(G_{i-1})$ such that $u_{i_0} <_\sigma x <_\sigma u_{|S_i|}$. Suppose otherwise that there exists such an $x$ where $u_{i_0} x \notin E(G_{i-1})$. Note that $l_{u_{|S_i|}} < r_{u_{i_0}}$, since $u_{i_0} <_\sigma u_{|S_i|}$ and $u_{i_0} u_{|S_i|} \in E(G_{i-1})$. Furthermore, since $S_i$ is a weakly reducible set, $\mathcal{I}[S_i]$ induces a proper interval representation of $G_{i-1}[S_i]$ by Definition 4. Then, since $u_{i_0} <_\sigma u_{|S_i|}$, it follows by Observation 4 that $l_{u_{i_0}} < l_{u_{|S_i|}}$. That is, $l_{u_{i_0}} < l_{u_{|S_i|}} < r_{u_{i_0}}$. Hence, since $u_{|S_i|} x \in E(G_{i-1})$ and $u_{i_0} x \notin E(G_{i-1})$ by our assumption, it follows that $r_{u_{i_0}} < l_x$, i.e., $l_{u_{i_0}} < l_{u_{|S_i|}} < r_{u_{i_0}} < l_x$. Finally $r_x < r_{u_{|S_i|}}$ from the choice of $x$, and thus $I_x \subseteq I_{u_{|S_i|}}$. Therefore, since $S_i$ is a weakly reducible set, $ux \in E(G_{i-1})$ for every $u \in S_i$ by Definition 4. This is a contradiction to our assumption that $u_{i_0} x \notin E(G_{i-1})$. Therefore $x u_{i_0} \in E(G_{i-1})$ for every $x \in V(G_{i-1})$ such that $u_{i_0} <_P x <_P u_{|S_i|}$.

We finally show that the set $\{x \in V(G_{i-1}) : u_{i_0} <_P x <_P u_{|S_i|}\}$ induces a clique in $G_{i-1}$. Since $u_{i_0} x \in E(G_{i-1})$ for every vertex $x$ in this set, we obtain that $l_x < r_{u_{i_0}} < r_x$. Hence, the set $\{x \in V(G_{i-1}) : u_{i_0} <_P x <_P u_{|S_i|}\}$ induces a clique in $G_{i-1}$, as all such intervals $I_x$ contain the point $r_{u_{i_0}}$. This completes the second part of our proof.

In the third part of our proof we show that there exists a path $P'$ which has at most $|D|+4$ vertex-maximal subpaths that consist only of vertices from $S_i$. Define $P' = (P_0, u_1, P_1, u_2, P_2, \ldots, P_{i_0-1}, u_{i_0}, P_{i_0}, P_{i_0+1}, \ldots, P_{|S_i|-1}, u_{i_0+1}, \ldots, u_{|S_i|}, P_{|S_i|})$. Notice that $V(P') = V(P)$. From the second part of our proof the set $\bigcup_{j=i_0}^{|S_i|-1} V(P_j) \bigcup_{j=i_0+1}^{|S_i|-1} \{u_j\}$ induces a clique. Thus, $P'$ is a path of $G_{i-1}$ with $V(P') = V(P)$. From the choice of $i_0$, for every $w \in [i_0-1]$, if $P_w \neq \emptyset$, then $z_w \in Z_1$. Since $|Z_1| \leq |D|+2$, from the first part of our proof, there exist at most $|D|+2$ paths $P_w$, $w \in [i_0-1]$, that are not empty. Thus, the subpath $P'_1 = (P_0, u_1, P_1, u_2, P_2, \ldots, P_{i_0-1})$ contains at most $|D|+2$ vertex-maximal subpaths consisting only of vertices of $S_i$. Furthermore, the subpath $P'_2 = (u_{i_0}, P_{i_0}, P_{i_0+1}, \ldots, P_{|S_i|-1}, u_{i_0+1}, \ldots, u_{|S_i|}, P_{|S_i|})$ of $P'$ clearly contains two vertex-maximal subpaths consisting only of vertices of $S_i$, namely the subpaths $(u_{i_0})$ and $(u_{i_0}, \ldots, u_{|S_i|})$. Since, $P' = (P'_1, P'_2)$, it follows that

$P'$ has at most $|D|+4$ vertex-maximal subpaths that consist only of vertices from $S_i$. Moreover, $P'$ has clearly at most $|S_i|$ such vertex-maximal subpaths from vertices of $S_i$. This completes the proof of the lemma. $\square$

We now present the next auxiliary technical lemma that will be used to prove the correctness of Reduction Rule 2 in Theorem 4.

**Lemma 17.** *Let $\ell$ and $k$ be positive integers. Let $G$ be a weighted interval graph with weight function $w: V(G) \to \mathbb{N}$ and $\mathcal{I}$ be an interval representation of $G$. Also, let $S$ be a weakly reducible set of $G$. Finally, let $G'$ be the graph obtained from $G$ by replacing $\mathcal{I}[S]$ with $\min\{|S|, k+4\}$ copies of the interval $I_{v_j} = \mathbf{span}(S)$, each having weight $\frac{1}{\min\{|S|,k+4\}} \sum_{u \in S} w(u)$. If the maximum weight of a path in $G'$ is $\ell$, then the maximum weight of a path in $G$ is at least $\ell$.*

**Proof.** Denote $S = \{u_1, u_2, \ldots, u_{|S|}\}$, where $u_1 <_\sigma u_2 <_\sigma \cdots <_\sigma u_{|S|}$. That is, $r_{u_1} < r_{u_2} < \ldots < r_{u_{|S|}}$. Furthermore, since $\mathcal{I}[S]$ induces a proper interval representation, it follows by Observation 4 that also $l_{u_1} < l_{u_2} < \ldots < l_{u_{|S|}}$. That is, $l_{u_{|S|}} = \max\{l_u : u \in S\}$ and $r_{u_1} = \min\{r_u : u \in S\}$.

Assume that the maximum weight of a path $P'$ in $G'$ is $\ell$. Define $V_0 = V(G') \setminus V(G)$ and denote $V_0 = \{v_j : j \in [\min\{|S|, k+4\}]\}$. If $V(P') \cap V_0 = \emptyset$, then $P'$ is also a path in $G$. Assume now that $V(P') \cap V_0 \neq \emptyset$. Then we may assume without loss of generality that $V_0 \subseteq V(P')$. Indeed, otherwise we can augment $P'$ to a path of $G'$ with greater weight by adding the missing copies of $\mathbf{span}(S)$ right after the last copy of $\mathbf{span}(S)$ in $P'$, which is a contradiction to the maximality assumption on $P'$. Furthermore, by Lemma 2 we may assume without loss of generality that $P'$ is normal.

Let $P' = \left(P'_0, v_1, P'_1, v_2, \ldots, v_{|V_0|-1}, P'_{|V_0|-1}, v_{|V_0|}, P'_{|V_0|}\right)$. Denote by $Z$ the set of all predecessors and successors of the vertices of $V_0$ in the path $P'$. Consider a vertex $z \in Z \cap \left(V(P'_1) \cup V(P'_2) \cup \ldots \cup V(P'_{|V_0|-1})\right)$. Since $z$ is a predecessor or a successor of a vertex $v \in V_0$ in $P'$, where $I_v = \mathbf{span}(S)$, it follows that $N(z) \cap S \neq \emptyset$. Assume that $S \nsubseteq N(z)$ in the initial graph $G$. Recall that $[l_{u_{|S|}}, r_{u_1}] = [\max\{l_v : v \in S\}, \min\{r_v : v \in S\}]$. Therefore, since we assumed that $S \nsubseteq N(z)$, Observation 5 implies that $I_z \cap [l_{u_{|S|}}, r_{u_1}] = \emptyset$, and thus either $r_z < l_{u_{|S|}}$ or $r_{u_1} < l_z$.

Suppose first that $r_z < l_{u_{|S|}}$, i.e., $r_z < l_{u_{|S|}} < r_{u_1}$. Let $l_{u_1} < l_z$, i.e., $l_{u_1} < l_z < r_z < r_{u_1}$. Then $I_z \subseteq I_{u_1}$, and thus $S \subseteq N(z)$, since $S$ is a weakly reducible set (cf. Definition 4). This is a contradiction to our assumption that $S \nsubseteq N(z)$. Let $l_z < l_{u_1}$, i.e., $l_z < l_{u_1} < r_z < r_{u_1}$. Then $\mathcal{I}[\{z, \mathbf{span}(S)\}]$ induces a proper interval representation. Therefore, since the path $P'$ of $G'$ is normal, Lemma 5 implies that vertex $z$ appears in $P'$ before the first vertex $v_1$ of $V_0$, i.e., $z \in P'_0$. This is a contradiction to our assumption that $z \in Z \cap \left(V(P'_1) \cup V(P'_2) \cup \ldots \cup V(P'_{|V_0|-1})\right)$.

Suppose now that $r_{u_1} < l_z$, i.e., $l_{u_{|S|}} < r_{u_1} < l_z$. Let $r_z < r_{u_{|S|}}$, i.e., $l_{u_{|S|}} < l_z < r_z < r_{u_{|S|}}$. Then $I_z \subseteq I_{u_{|S|}}$, and thus $S \subseteq N(z)$, since $S$ is a weakly reducible set (cf. Definition 4). This is a contradiction to our assumption that $S \nsubseteq N(z)$. Let $r_{u_{|S|}} < r_z$, i.e., $l_{u_{|S|}} < l_z < r_{u_{|S|}} < r_z$. Then $\mathcal{I}[\{z, \mathbf{span}(S)\}]$ induces a proper interval representation. Therefore, since the path $P'$ of $G'$ is normal, Lemma 5 implies that vertex $z$ appears in $P'$ after the last vertex $v_{|V_0|}$ of $V_0$, i.e., $z \in P'_{|V_0|}$. This is a contradiction to our assumption that $z \in Z \cap \left(V(P'_1) \cup V(P'_2) \cup \ldots \cup V(P'_{|V_0|-1})\right)$.

Summarizing, $S \subseteq N(z)$ in the initial graph $G$, for every vertex $z \in V(P'_1) \cup V(P'_2) \cup \ldots \cup V(P'_{|V_0|-1})$ which is a predecessor or a successor of a vertex of $V_0$ in $P'$. Therefore $P = \left(P'_0, u_1, P'_1, u_2, \ldots, u_{|V_0|-1}, P'_{|V_0|-1}, u_{|V_0|}, u_{|V_0|+1}, \ldots, u_{|S|}, P'_{|V_0|}\right)$ is a path in $G$, where $V(P) = V(P')$. Finally, since $\sum_{i \in [|S|]} w(u_i) = \sum_{j \in [\min\{|S|,k+4\}]} w(v_j)$, it follows that $\sum_{v \in V(P)} w(v) = \sum_{v \in V(P')} w(v)$. Thus $G$ has a path $P$ of weight at least $\ell$. $\square$

Now we are ready to prove the correctness of our Reduction Rule 2.

**Theorem 4.** *Let $\ell$ be a positive integer. Let $G$ be a weighted interval graph and $\widehat{G}$ be the weighted interval graph obtained by Reduction Rule 2. Then the maximum weight of a path in $G$ is $\ell$ if and only if the maximum weight of a path in $\widehat{G}$ is $\ell$.*

**Proof.** First assume that the maximum weight of a path in $\widehat{G} = G_{|\mathcal{S}|}$ is $\ell$. Then, by iteratively applying Lemma 17 it follows that the maximum weight of a path in $G$ is at least $\ell$.

Conversely, assume that the maximum weight of a path in $G$ is $\ell$. We show by induction on $i \in \{0, 1, \ldots, |\mathcal{S}|\}$ that the maximum weight of a path in $G_i$ is at least $\ell$. For $i = 0$ we have $G_0 = G$ and the argument follows by our assumption. This proves the induction basis.

For the induction step, let $i \geq 1$ and assume that the weight of a maximum path in $G_{i-1}$ is at least $\ell$. Let $P$ be a path of maximum weight in $G_{i-1}$, i.e., the weight of $P$ is at least $\ell$. Without loss of generality, from Lemma 2, we may also assume that $P$ is normal. Furthermore, Lemma 14 implies that either $S_i \subseteq V(P)$ or $S_i \cap V(P) = \emptyset$. Notice that if $S_i \cap V(P) = \emptyset$, then $P$ is also a path of $G_i$. Suppose that $S_i \subseteq V(P)$. Then from Lemma 16, we can obtain a path $\widehat{P}$ such that $V(\widehat{P}) = V(P)$ and such that $\widehat{P}$ has $q \leq \min\{|S_i|, |D|+4\}$ vertex-maximal subpaths consisting only of vertices of $S_i$. Let $\widehat{P}_1, \widehat{P}_2, \ldots, \widehat{P}_q$ be those subpaths. Consider now the path $P'$ that is obtained by replacing in the interval representation of $\widehat{P}$ each of the subpaths $\widehat{P}_1, \widehat{P}_2, \ldots, \widehat{P}_{q-1}$ with a copy of $\mathbf{span}(S_i)$, and by replacing the subpath $\widehat{P}_q$ with $\min\{|S_i|, |D|+4\} - q$ copies of $\mathbf{span}(S_i)$.

Note that $P'$ is a path in the graph $G_i$. Recall by the definition of $G_i$ (cf. Reduction Rule 2) that each of the $\min\{|S_i|, |D|+4\}$ copies of the interval $\mathbf{span}(S_i)$ has weight $w(v_j) = \frac{1}{\min\{|S_i|,|D|+4\}} \sum_{u \in S_i} w(u)$. Since the total weight of all these copies of $\mathbf{span}(S_i)$ is equal to $\sum_{u \in S_i} w(u)$, it follows that $\sum_{v \in V(P')} w(v) = \sum_{v \in V(P)} w(v)$. That is, the weight of the path $P'$ of $G_i$ is at least $\ell$. Therefore the maximum weight of a path in $G_i$ is at least $\ell$. This completes the induction step and the proof of the theorem. □

## 3. Special weighted interval graphs

In this section we sequentially apply the two data reductions of Sections 2.2 and 2.3 to a given interval graph $G$ with a proper interval deletion set $D$. To do so, we first define a specific family $\mathcal{S}_1$ of *reducible* sets in $G \setminus D$ and we apply Reduction Rule 1 to $G$ with respect to the family $\mathcal{S}_1$, resulting in the weighted interval graph $G^{\#}$. Then we define a specific family $\mathcal{S}_2$ of *weakly reducible* sets in $G^{\#} \setminus D$ and we apply Reduction Rule 2 to $G^{\#}$ with respect to the family $\mathcal{S}_2$, resulting in the weighted interval graph $\widehat{G}$. As it turns out, the vertex sets of $\mathcal{S}_1 \cup \mathcal{S}_2$ are a partition of the graph $G \setminus D$. The final graph $\widehat{G}$ is then given as input to our fixed-parameter algorithm of Section 4.

We now introduce the notion of a *special weighted interval graph* with *parameter* $\kappa$. As we will prove at the end of this section, the constructed graph $\widehat{G}$ is a special weighted interval graph with parameter $\kappa$, where $\kappa$ depends only on the size of $D$ (cf. Theorem 6). Furthermore $\widehat{G}$ can be computed in $O(k^2 n)$ time (cf. Theorem 7).

**Definition 5** *(special weighted interval graph with parameter $\kappa$).* Let $G = (V, E)$ be a weighted interval graph, $\mathcal{I} = \{I_v : v \in V\}$ be an interval representation of $G$, and $\kappa \in \mathbb{N}$, where the vertex set $V$ can be partitioned into two sets $A$ and $B$ such that:

1. $A$ is an independent set in $G$,
2. for every $v \in A$ and every $u \in V \setminus \{v\}$, we have $I_u \nsubseteq I_v$, and
3. $|B| \leq \kappa$.

Then $G$ (resp. $\mathcal{I}$) is a *special weighted interval graph* (resp. *special interval representation*) with *parameter* $\kappa$. The partition $V = A \cup B$ is a *special vertex partition* of $G$.

### 3.1. The graph $G^{\#}$

Let $D$ be a proper interval deletion set of $G$ with $k$ vertices $d_1, d_2, \ldots, d_k$, i.e., $G \setminus D$ is a proper interval graph. First we add two isolated dummy vertices $d_0, d_{k+1}$ to the set $D$ (together with the corresponding intervals $I_{d_0}, I_{d_{k+1}}$), such that $d_0 <_\sigma v <_\sigma d_{k+1}$ for every vertex $v \in V(G)$. Note that $\mathcal{I}' = \mathcal{I} \cup \{I_{d_0}, I_{d_{k+1}}\}$ is the interval representation of an interval graph $G'$, where $V(G') = V(G) \cup \{d_0, d_{k+1}\}$ and $E(G') = E(G)$. For simplicity of the presentation we refer in the following to $G'$ and $\mathcal{I}'$ by $G$ and $\mathcal{I}$, respectively. Furthermore we denote $D = \{d_0, d_1, \ldots, d_k, d_{k+1}\}$, where $d_0 <_\sigma d_1 <_\sigma \ldots <_\sigma d_k <_\sigma d_{k+1}$ and $d_0, d_{k+1}$ are the two dummy isolated vertices. Now we define the following four sets:

$$
\begin{aligned}
L &= \{l_v : \ v \in D\}, \\
R &= \{r_v : \ v \in D\}, \\
U &= V(G) \setminus D, \\
U^* &= \{u \in U : \ I_u \cap R = \emptyset\}.
\end{aligned}
$$

Furthermore, for every $i \in [k+1]$ we define the following set:

$$L_i = \{l \in L : \ r_{d_{i-1}} < l < r_{d_i}\} \cup \{r_{d_{i-1}}, r_{d_i}\}. \tag{2}$$

For every $i \in [k+1]$ we denote $L_i = \{l_{i,0}, l_{i,1}, \ldots, l_{i,p_i}\}$, where $l_{i,0} < l_{i,1} < \cdots < l_{i,p_i}$. Note that $l_{i,0} = r_{d_{i-1}}$ and $l_{i,p_i} = r_{d_i}$, and thus $|L_i| \geq 2$ for every $i \in [k+1]$. Now, for every $i \in [k+1]$ and $x \in [p_i]$ define

$$U^*_{i,x} = \{u \in U^* : \ l_{i,x-1} < r_u < l_{i,x}\}$$

and

$$U^{**}_{i,x} = \left\{u \in U^*_{i,x} : \ N(u) \cap \left(\bigcup\nolimits_{j \in [x-1]} U^*_{i,j}\right) = \emptyset\right\}.$$

Note that the set $U^*$ is partitioned by the sets $\{U^*_{i,x} : i \in [k+1], x \in [p_i]\}$.

**Observation 7.** Let $u \in U$ such that $I_u$ is strictly contained between two consecutive points of $R \cup L$. Then $u \in U^{**}_{i,x}$, for some $i \in [k+1]$ and $x \in [p_i]$.

**Lemma 18.**

$$\sum_{i=1}^{k+1} p_i = 2(k+1).$$

**Proof.** First, note by Eq. (2) that $|L_i| = 2 + |\{l \in L : r_{d_{i-1}} < l < r_{d_i}\}|$. Furthermore, since $L_i = \{l_{i,0}, l_{i,1}, \dots, l_{i,p_i}\}$, it follows that $p_i = |L_i| - 1$, i.e., $p_i = 1 + |\{l \in L : r_{d_{i-1}} < l < r_{d_i}\}|$. Therefore,

$$\sum_{i=1}^{k+1} p_i = (k+1) + \sum_{i=1}^{k+1} |\{l \in L : r_{d_{i-1}} < l < r_{d_i}\}| = 2(k+1). \qquad \square$$

**Lemma 19.** *Let $i \in [k+1]$ and $x \in [p_i]$. The interval representation $\mathcal{I}[U^*_{i,x}]$ of $G[U^*_{i,x}]$ is a proper interval representation.*

**Proof.** The proof is done by contradiction. Assume that, for some $i \in [k+1]$ and $x \in [p_i]$, $\mathcal{I}[U^*_{i,x}]$ is not a proper interval representation of $G[U^*_{i,x}]$. That is, there exist two vertices $u, v \in U^*_{i,x}$ such that $I_u \subseteq I_v$. From the preprocessing of Theorem 2 there exist two vertices $z, z' \in V(G)$ such that $z <_\sigma u <_\sigma z'$, where $z, z' \in N(v) \setminus N(u)$.

First suppose that $z, z' \notin D$. Then the vertices $\{z, z', v, u\}$ induce a $K_{1,3}$ in $G \setminus D$, which is a contradiction to the assumption that $D$ is a proper interval deletion set of $G$ [48].

Now suppose that $z \in D$. Since $zv \in E(G)$ and $r_z < r_u < r_v$, it follows that $l_v < r_z$. Thus $r_z \in I_v$. Therefore, since we assumed that $z \in D$, it follows that $I_v \cap R \neq \emptyset$ and thus $v \notin U^*$ (cf. the definition of the set $U^*$). This is a contradiction to the assumption that $v \in U^*_{i,x} \subseteq U^*$.

Finally suppose that $z' \in D$. Then, since $u <_\sigma z'$ and $z' \notin N(u)$, it follows that $r_u < l_{z'}$. If $r_v < l_{z'}$, then $z' \notin N(v)$, which is a contradiction. Thus $l_{z'} < r_v$, i.e., $r_u < l_{z'} < r_v$. Therefore, since we assumed that $z' \in D$, it follows that $u, v$ do not belong to the same set $U^*_{i,x}$ (cf. the definition of $U^*_{i,x}$), which is a contradiction to our assumption.

Thus, for every $i \in [k+1]$ and $x \in [p_i]$, $\mathcal{I}[U^*_{i,x}]$ is a proper interval representation of $G[U^*_{i,x}]$. $\square$

For every $i \in [k+1]$ and $x \in [p_i]$ we denote the connected components of $U^{**}_{i,x}$ by $C^1_{i,x}, C^2_{i,x}, \dots, C^{q(i,x)}_{i,x}$, such that $V(C^1_{i,x}) <_\sigma V(C^2_{i,x}) <_\sigma \cdots <_\sigma V(C^{q(i,x)}_{i,x})$. Note that any two distinct components $C^t_{i,x}$ and $C^{t'}_{i',x'}$ are disjoint, i.e., $V(C^t_{i,x}) \cap V(C^{t'}_{i',x'}) = \emptyset$. Furthermore we define the family $\mathcal{S}_1$ of vertex subsets of $V(G) \setminus D$ as follows:

$$\mathcal{S}_1 = \{V(C^t_{i,x}) : i \in [k+1],\ x \in [p_i],\ t \in [q(i,x)]\}. \tag{3}$$

**Lemma 20.** *Every set $S \in \mathcal{S}_1$ is reducible.*

**Proof.** Consider a set $S \in \mathcal{S}_1$. Then $S = V(C^t_{i,x})$, for some $i \in [k+1]$, $x \in [p_i]$, and $t \in [q(i,x)]$. We need to prove that the two conditions of Definition 3 are satisfied for the connected component $C^t_{i,x}$ of $U^{**}_{i,x}$. For Condition 1, recall that $V(C^t_{i,x}) \subseteq U^{**}_{i,x} \subseteq U^*_{i,x}$. Thus, since the interval representation $\mathcal{I}[U^*_{i,x}]$ of $G[U^*_{i,x}]$ is proper by Lemma 19, the interval representation $\mathcal{I}[V(C^t_{i,x})]$ of $G[V(C^t_{i,x})]$ is a connected proper interval representation. This proves Condition 1 of Definition 3.

Now let $v \in V(G)$ such that $I_v \subseteq \mathbf{span}(V(C^t_{i,x}))$. Recall by the definition of $V(C^t_{i,x})$ that $G[V(C^t_{i,x})]$ is connected. Therefore, since $I_u \cap R = \emptyset$ for every $u \in V(C^t_{i,x}) \subseteq U^*$, it follows that $\mathbf{span}(V(C^t_{i,x})) \cap R = \emptyset$, and thus also $I_v \cap R = \emptyset$. That is, $v \in U^*$ (cf. the definition of $U^*$). Let $V(C^t_{i,x}) = \{u_1, u_2, \dots, u_a\}$, where $u_1 <_\sigma u_2 <_\sigma \dots <_\sigma u_a$. Note that $r_v \le r_{u_a}$, since $I_v \subseteq \mathbf{span}(V(C^t_{i,x}))$ by assumption. Furthermore, since $\mathcal{I}[V(C^t_{i,x})]$ is a proper interval representation as we proved above, Observation 4 implies that $l_{u_1} < l_{u_2} < \dots < l_{u_a}$. Suppose that $v <_\sigma u_1$. Then, since $I_v \subseteq \mathbf{span}(V(C^t_{i,x}))$, it follows that $I_v \subseteq I_{u_1}$. This is a contradiction to Condition 1 of Definition 3 that we proved above. Thus $u_1 \le_\sigma v$. That is, $l_{i,x-1} < r_{u_1} \le r_v \le r_{u_a} < l_{i,x}$. Therefore $v \in U^*_{i,x}$ (cf. the definition of $U^*_{i,x}$). Moreover, since $I_v \subseteq \mathbf{span}(V(C^t_{i,x}))$ by assumption, it follows that $N(v) \subseteq \bigcup_{u \in V(C^t_{i,x})} N(u)$, and thus $N(v) \cap \bigcup_{j \in [x-1]} U^*_{i,j} = \emptyset$ (cf. the definition of $U^{**}_{i,x}$). It follows that $v \in U^{**}_{i,x}$, and thus $v \in V(C^t_{i,x})$. This proves Condition 2 of Definition 3. $\square$

**Lemma 21.** *If $S$ and $S'$ are two distinct elements of $\mathcal{S}_1$, then $\mathbf{span}(S) \cap \mathbf{span}(S') = \emptyset$.*

**Proof.** Towards a contradiction let $S$ and $S'$ be two distinct elements of $\mathcal{S}_1$. That is, $S = V(C^t_{i,x})$ and $S' = V(C^{t'}_{i',x'})$, for some $i, i' \in [k+1]$, $x \in [p_i]$, $x' \in [p_{i'}]$, and $t \in [q(i,x)]$, $t' \in [q(i',x')]$. Assume that $\mathbf{span}(S) \cap \mathbf{span}(S') \neq \emptyset$. Then there exist $u \in S$ and $u' \in S'$ such that $I_u \cap I_{u'} \neq \emptyset$.

First suppose that $i \neq i'$. Without loss of generality we assume that $i' < i$. Note that $l_{i,x-1} < r_u < l_{i,x}$, since $u \in V(C^t_{i,x}) \subseteq U^*_{i,x}$. Furthermore, since by the definition of the set $L_i$ we have $r_{d_{i-1}} \le l_{i,x-1}$ and $l_{i,x} \le r_{d_i}$, it follows that

$r_{d_{i-1}} < r_u < r_{d_i}$. Similarly it follows that $r_{d_{i'-1}} < r_{u'} < r_{d_{i'}}$. Therefore, $r_{u'} < r_{d_{i'}} \leq r_{d_{i-1}} < r_u$. Furthermore, since $I_u \cap I_{u'} \neq \emptyset$, it follows that $l_u < r_{u'}$. That is, $l_u < r_{u'} < r_{d_{i-1}} < r_u$, and thus $r_{d_{i-1}} \in I_u$. This is a contradiction, since $u \in U^*$ (cf. the definition of $U^*$). Therefore, $i' = i$.

Now suppose that $x' \neq x$. Without loss of generality we assume that $x' < x$. Note that $u \in V(C^t_{i,x}) \subseteq U^{**}_{i,x}$ and that $u' \in V(C^{t'}_{i,x'}) \subseteq U^*_{i,x'} \subseteq \bigcup_{j\in[x-1]} U^*_{i,j}$. Therefore, since $N(u) \cap \left(\bigcup_{j\in[x-1]} U^*_{i,j}\right) = \emptyset$ (cf. the definition of $U^{**}_{i,x}$), it follows that $u' \notin N(u)$. This is a contradiction, since $I_u \cap I_{u'} \neq \emptyset$. Therefore, $x' = x$.

Summarizing, the sets $S$ and $S'$ are two different connected components of $U^{**}_{i,x}$. Therefore, there are no vertices $u \in S$ and $u' \in S'$ such that $I_u \cap I_{u'} \neq \emptyset$, which is a contradiction to our assumption. It follows that $\mathbf{span}(S) \cap \mathbf{span}(S') \neq \emptyset$, for any two distinct sets $S, S' \in \mathcal{S}_1$. $\square$

Note that, for every $i, x, t$, the connected component $C^t_{i,x}$ of $U^{**}_{i,x}$ contains no vertices of $D$, since by definition $U^{**}_{i,x} \subseteq U = V(G) \setminus D$. Therefore, since all sets of $\mathcal{S}_1$ are reducible (by Lemma 20) and disjoint, we can apply Reduction Rule 1 to the graph $G$ with respect to the sets of $\mathcal{S}_1$, by replacing in the interval representation $\mathcal{I}$ the intervals $\{I_v : v \in S\}$ with the interval $I_S = \mathbf{span}(S)$ with weight $|S|$, for every $S \in \mathcal{S}_1$. Denote the resulting weighted graph by $G^\# = (V^\#, E^\#)$ and its interval representation by $\mathcal{I}^\#$. Furthermore denote by $\sigma^\#$ the right-endpoint ordering of $\mathcal{I}^\#$. Then, the next corollary follows immediately by Theorem 3.

**Corollary 1.** *The maximum number of vertices of a path in $G$ is equal to the maximum weight of a path in $G^\#$.*

Define $A = V(G^\#) \setminus V(G)$, i.e., each vertex $v \in A$ corresponds to an interval $I_v = \mathbf{span}(S)$ in the interval representation $\mathcal{I}^\#$, where $S \in \mathcal{S}_1$. Recall that for every $S \in \mathcal{S}_1$, we have that $S = V(C^t_{i,x})$, where $i \in [k+1]$, $x \in [p_i]$, and $t \in [q(i,x)]$. Thus, in the remainder of this section we denote $A = \{v^t_{i,x} :\ i \in [k+1],\ x \in [p_i],\ t \in [q(i,x)]\}$. Furthermore, for every $v^t_{i,x} \in A$ we denote for simplicity the corresponding interval in the representation $\mathcal{I}^\#$ by $I^t_{i,x} = I_{v^t_{i,x}}$. Recall that $V(C^1_{i,x}) <_\sigma V(C^2_{i,x}) <_\sigma \cdots <_\sigma V(C^{q(i,x)}_{i,x})$ in the graph $G$, and thus $v^1_{i,x} <_{\sigma^\#} v^2_{i,x} <_{\sigma^\#} \ldots <_{\sigma^\#} v^{q(i,x)}_{i,x}$ in the graph $G^\#$. Furthermore, since $\mathbf{span}(S) \cap \mathbf{span}(S') = \emptyset$ for any two distinct sets $S, S' \in \mathcal{S}_1$ by Lemma 21, the next corollary follows immediately by Lemma 13.

**Corollary 2.** *The set $D$ remains a proper interval deletion set of the weighted interval $G^\# = (V^\#, E^\#)$. Furthermore the set $V^\# \setminus D$ can be partitioned into the sets $A = V(G^\#) \setminus V(G)$ and $U^\# = V^\# \setminus (D \cup A)$, such that:*

1. *$A$ is an independent set of $G^\#$, and*
2. *for every $v^t_{i,x} \in A$ and every $u \in V^\# \setminus \{v^t_{i,x}\}$, we have $I_u \nsubseteq I^t_{i,x}$.*

In the next theorem we prove that the weighted interval graph $G^\#$ and the vertex subset $A$ (cf. Corollary 2) can be computed in $O(n)$ time.

**Theorem 5.** *Let $G = (V, E)$ be an interval graph, where $|V| = n$. Let $D$ be a proper interval deletion set of $G$, where $D = \{d_0, d_1, \ldots, d_k, d_{k+1}\}$. Then the graph $G^\# = (V^\#, E^\#)$ and the independent set $A \subseteq V^\#$ can be computed in $O(n)$ time.*

**Proof.** Denote by $\mathcal{I}$ the given interval representation of $G$. Recall that the endpoints of the intervals in $\mathcal{I}$ are given sorted increasingly, e.g., in a linked list $M$. The sets $L$ and $R$ can be computed in $O(k)$ time, since they store the left and the right endpoints of the intervals of the set $D$, where $|D| = k$. Furthermore, the set $U = V \setminus D$ can be clearly computed in $O(n)$ time.

The set $U^*$ and the sets $L_i$, $i \in [k+1]$, can be efficiently computed as follows. First we visit all endpoints of the intervals in $\mathcal{I}$ (in increasing order). For every endpoint $p \in \{l_u, r_u : u \in U\} \cup \{l_{d_1}, l_{d_2}, \ldots, l_{d_{k+1}}\}$ that we visit, such that $r_{d_i} < p < r_{d_{i+1}}$, where $r_{d_i}, r_{d_{i+1}} \in D$, we add to $p$ the label $label(p) = d_i$. Initially, we set $L_i = \{r_{d_{i-1}}, r_{d_i}\}$, for every $i \in [k+1]$. Then we iterate for every $p \in \{l_{d_1}, l_{d_2}, \ldots, l_{d_{k+1}}\}$. If $label(p) = d_i$ then we add $p$ to the set $L_i$. Note that, during this computation, we can store the elements of each set $L_i$ in increasing order, using a linked list. Furthermore, in the same time we add to every element $p$ of the set $L_i$ a pointer to the position of $p$ in the linked list $M$, which keeps the endpoints of the intervals in $\mathcal{I}$ in increasing order. To compute the set $U^*$ we iterate for every $u \in U$ and we compare $label(l_u)$ with $label(r_u)$. If $label(l_u) = label(r_u)$ then we set $u \in U^*$, otherwise we set $u \notin U^*$. Similarly, during this computation we can store the elements of the set $U^*$ in increasing order (according to the order of their right endpoints in the linked list $M$). Since the endpoints in $\mathcal{I}$ are assumed to be sorted in increasing order, the computation of $U^*$ and of all sets $L_i$, $i \in [k+1]$, can be done in total in $O(n)$ time.

Note that there are in total $\sum_{i=1}^{k+1} p_i = 2(k+1)$ different sets $U^*_{i,x}$, where $i \in [k+1]$ and $x \in [p_i]$ (cf. Lemma 18). All these sets $U^*_{i,x}$ can be efficiently computed as follows, using the sets $U^*$ and $L_1, L_2, \ldots, L_{k+1}$, which we have already computed, as follows. For every $i \in [k+1]$ we iterate over the points $\{l_{i,0}, l_{i,1}, \ldots, l_{i,p_i}\}$ of the set $L_i$ in increasing order. For every $x \in [p_i]$ we visit sequentially in the linked list $M$ (which keeps the endpoints of $\mathcal{I}$ in increasing order) from the point $l_{i,x-1}$ until the point $l_{i,x}$. For every point $p$ between $l_{i,x-1}$ and $l_{i,x}$, we check whether $p = r_u$ and $u \in U^*$, and if this is the case then we add vertex $u$ to the set $U^*_{i,x}$. This check on point $p$ can be done on $O(1)$ time, e.g., by checking whether $label(l_u) = label(r_u)$, as we described above. Thus, as we scan once through the linked list $M$, the computation of all sets $U^*_{i,x}$ can be done in $O(n)$ time in total. Note that, during this computation we can store the elements of each of the sets $U^*_{i,x}$ in increasing order (according to the order of their right endpoints in the linked list $M$).

The computation of the sets $U^{**}_{i,x}$, where $i \in [k+1]$ and $x \in [p_i]$, can be done efficiently as follows. First, we scan once through each of the $O(k)$ sets $U^*_{i,x} \neq \emptyset$ and, for every such set, we compute its rightmost right endpoint $p_{i,x}$ in the interval representation $\mathcal{I}$, i.e., $p_{i,x} = \max\{r_u : u \in U^*_{i,x}\}$. Furthermore define $p_{i,0} = r_{d_{d-1}}$, for every $i \in [k+1]$. The computation of all such points $p_{i,x}$ can be done in $O(n)$ time in total. Now, for every fixed $i \in [k+1]$, the computation of the sets $U^{**}_{i,1}, U^{**}_{i,2}, \ldots, U^{**}_{i,p_i}$ can be done as follows. Initially, set $p = p_{i,0}$. We iterate for every $x = 1, 2, \ldots, p_i$ (in this order). For every such $x$, we first define $p = \max\{p, p_{i,x-1}\}$. Then, for every vertex $u \in U^*_{i,x}$ we check whether $p < l_u$, and if it is the case then we add vertex $u$ to the set $U^{**}_{i,x}$. It can be easily checked that this process eventually correctly computes the sets $U^{**}_{i,x}$ (cf. the definition of $U^{**}_{i,x}$). Since every two sets $U^*_{i,x}, U^*_{i',x'}$ are disjoint, all the sets $U^{**}_{i,x}$, where $i \in [k+1]$ and $x \in [p_i]$, can be done by these computations in $O(n)$ time in total. Moreover, in the same time we can store the elements of each of the sets $U^{**}_{i,x}$ in increasing order (according to the order of their right endpoints in the linked list $M$).

Let now $i \in [k+1]$ and $x \in [p_i]$. The computation of the connected components $C^1_{i,x}, C^2_{i,x}, \ldots, C^{q(i,x)}_{i,x}$ of $U^{**}_{i,x}$ can be done efficiently as follows. We visit all vertices of $U^{**}_{i,x}$ in increasing order. For every such vertex $u$, we compare its left endpoint $l_u$ with the right endpoint $r_{u'}$ of the previous vertex $u'$ of $U^{**}_{i,x}$. If $l_u < r_{u'}$ then $u$ belongs to the same connected component $C^t_{i,x}$ where $u'$ belongs, otherwise $u$ belongs to the next connected component $C^{t+1}_{i,x}$. The correctness of this computation of $C^1_{i,x}, C^2_{i,x}, \ldots, C^{q(i,x)}_{i,x}$ follows from the fact that the interval representation $\mathcal{I}[U^*_{i,x}]$ of $G[U^*_{i,x}]$ is proper by Lemma 19 and by Observation 4. Since we visit every vertex of each $U^{**}_{i,x}$ once, where $i \in [k+1]$ and $x \in [p_i]$, the computation of all connected components $C^t_{i,x}$, where $i \in [k+1]$, $x \in [p_i]$, and $t \in [q(i,x)]$, can be done in $O(n)$ time in total. Note that, in the same time we can also compute the interval $\mathbf{span}(C^t_{i,x})$ for every such component $C^t_{i,x}$, by just keeping track of the left endpoint $l_u$ (resp. right endpoint $r_{u'}$) of the leftmost vertex $u$ (resp. of the rightmost vertex $u'$) in $C^t_{i,x}$.

Summarizing, we can compute in $O(n)$ time the family $\mathcal{S}_1$ that contains all vertex sets $S = V(C^t_{i,x})$, where $i \in [k+1]$, $x \in [p_i]$, and $t \in [q(i,x)]$, cf. Eq. (3). Moreover, in the same time we can also compute the intervals $I_S = \mathbf{span}(S)$, where $S \in \mathcal{S}_1$. Then the interval representation $\mathcal{I}^\#$ can be computed from $\mathcal{I}$ in $O(n)$ time, by replacing for every $S \in \mathcal{S}_1$ the intervals $\{I_v : v \in S\}$ with the interval $I_S = \mathbf{span}(S)$. Note that these intervals $\{\mathbf{span}(S) : S \in \mathcal{S}_1\}$ are exactly the intervals of the vertices in the independent set $A$ of $G^\#$. Therefore, the sets $A \subseteq V^\#$ and $U^\# = V^\# \setminus (D \cup A)$ can be computed in $O(n)$ time. □

### 3.2. The graph $\widehat{G}$

Consider the weighted interval graph $G^\# = (V^\#, E^\#)$ with the interval representation $\mathcal{I}^\#$ and the right-endpoint ordering $\sigma^\#$ that we constructed in Section 3.1. Recall that, by Corollary 2, $D = \{d_0, d_1, \ldots, d_k, d_{k+1}\} \subseteq V^\#$ is a proper interval deletion set of $G^\#$ and that the vertices of $V^\# \setminus D$ are partitioned into the independent set $A = \{v^t_{i,x} : \ i \in [k+1], \ x \in [p_i], \ t \in [q(i,x)]\}$ and the set $U^\# = V^\# \setminus (D \cup A)$. Recall that $v^1_{i,x} <_{\sigma^\#} v^2_{i,x} <_{\sigma^\#} \ldots <_{\sigma^\#} v^{q(i,x)}_{i,x}$. For every vertex $v^t_{i,x} \in A$, the interval of $v^t_{i,x}$ in the representation $\mathcal{I}^\#$ is denoted by $I^t_{i,x}$. We define the set $T$ of endpoints in the representation $\mathcal{I}^\#$ as

$$T = R \cup L \cup \bigcup_{i,x} \left\{ l_{I^1_{i,x}}, r_{I^1_{i,x}}, l_{I^2_{i,x}}, r_{I^2_{i,x}}, l_{I^{q(i,x)-1}_{i,x}}, r_{I^{q(i,x)-1}_{i,x}}, l_{I^{q(i,x)}_{i,x}}, r_{I^{q(i,x)}_{i,x}} \right\}.$$

Note that $|T| \leq |R| + |L| + 8\sum_{i=1}^{k+1} p_i$, and thus Lemma 18 implies that $|T| \leq 18k + 16$. We denote $T = \{t_1, t_2, \ldots, t_{|T|}\}$, where $t_1 < t_2 < \cdots < t_{|T|}$. For every $1 \leq j \leq i \leq |T|$ we define

$$U_{ji} = \{u \in U^\# : t_{j-1} < l_u < t_j \text{ and } t_{i-1} < r_u < t_i\}. \tag{4}$$

Note that $\{U_{ji} : 1 \leq j \leq i \leq |T|\}$ provides a partition of $U^\#$. As the next lemma shows, it suffices to consider in the following only the sets $U_{ji}$ such that $j \neq i$.

**Lemma 22.** *For every $i \in [k+1]$, $U_{ii} = \emptyset$.*

**Proof.** Let $u \in U_{ii}$, for some $i \in |T|$. Since $U_{ii} \subseteq U^\# = V^\# \setminus (D \cup A)$, note that vertex $u$ exists also in the original (unweighted) interval graph $G$. Furthermore, since $I_u$ is strictly contained between two consecutive points of $T$, it is also strictly contained between two consecutive points of $R \cup L \subseteq T$. Therefore, for some $i \in [k+1]$ and $x \in [p_i]$, $u \in U^{**}_{i,x}$ by Observation 7. However, all vertices of $\bigcup_{i,x} U^{**}_{i,x}$ in the initial interval graph $G$ have been replaced by the vertex set $A$ in the weighted interval graph $G^\#$. This is a contradiction, since $u \in U^\# = V^\# \setminus (D \cup A)$. Thus $U_{ii} = \emptyset$. □

We are now ready to define the family $\mathcal{S}_2$ of vertex subsets of $U^\#$ as follows:

$$\mathcal{S}_2 = \{U_{ji} : 1 \leq j < i \leq |T|\}. \tag{5}$$

**Lemma 23.** *Every set $S \in \mathcal{S}_2$ is weakly reducible in the graph $G^\# = (V^\#, E^\#)$.*

**Proof.** Consider a set $S \in \mathcal{S}_2$. Then $S = U_{ji}$, for some $1 \le j < i \le |T|$. In the first part of the proof we show by contradiction that $\mathcal{I}^\#[U_{ji}]$ is a proper interval representation of $G^\#[U_{ji}]$. Assume otherwise that there exist two vertices $v, u \in U_{ji}$ such that $I_v \subseteq I_u$. Since the intervals for the vertices of $U_{ji}$ are the same in both interval representations $\mathcal{I}$ and $\mathcal{I}^\#$, it follows that $I_v \subseteq I_u$ in the representation $\mathcal{I}$ of the initial (unweighted) interval graph $G$. Then, from the preprocessing of Theorem 2 there exist two vertices $z, z' \in V(G)$ such that $z <_\sigma v <_\sigma z'$, where $z, z' \in N(u) \setminus N(v)$ in the graph $G$.

First suppose that $z, z' \notin D$. Then the vertices $\{z, z', u, v\}$ induce a $K_{1,3}$ in $G \setminus D$, which is a contradiction to the assumption that $D$ is a proper interval deletion set of $G$ [48].

Now suppose that $z \in D$. Then $r_z \in R \subseteq T$. If $r_z < l_u$ then $zu \notin E(G)$, which is a contradiction. Thus $l_u < r_z$. Moreover, since $zv \notin E(G)$ and $z <_\sigma v$, it follows that $r_z < l_v$. That is, $l_u < r_z < l_v$, where $r_z \in T$. This is a contradiction to the assumption that both $v$ and $u$ belong to the same set $U_{ji}$.

Finally suppose that $z' \in D$. Then $l_{z'} \in L \subseteq T$. If $r_u < l_{z'}$ then $z'u \notin E(G)$, which is a contradiction. Thus $l_{z'} < r_u$. Moreover, since $z'v \notin E(G)$ and $v <_\sigma z'$, it follows that $r_v < l_{z'}$. That is, $r_v < l_{z'} < r_u$, where $l_{z'} \in T$. This is a contradiction to the assumption that both $v$ and $u$ belong to the same set $U_{ji}$. This proves Condition 1 of Definition 4.

In the second part of the proof we show by contradiction that for every $u \in U_{ji}$ and every $v \in V^\#$, if $I_v \subseteq I_u$, then $U_{ji} \subseteq N(v)$ in the graph $G^\#$. Let $u \in U_{ji}$ and $v \in V^\#$ such that $I_v \subseteq I_u$. Assume that there exists a vertex $u' \in U_{ji}$ such that $u'v \notin E^\#$. First suppose that $I_v \cap T \neq \emptyset$, i.e., $l_v \le t_0 \le r_v$ for some $t_0 \in T$. Let $v <_{\sigma^\#} u'$. Then, since $u'v \notin E^\#$ by assumption, it follows that $r_v < l_{u'}$. Furthermore $l_u < l_v$, since $I_v \subseteq I_u$. Therefore, $l_u < l_v \le t_0 \le r_v < l_{u'}$, i.e., $l_u < t_0 < l_{u'}$, where $t_0 \in T$. This is a contradiction to the assumption that both $u$ and $u'$ belong to the same set $U_{ji}$. Let $u' <_{\sigma^\#} v$. Then, since $u'v \notin E^\#$ by assumption, it follows that $r_{u'} < l_v$. Furthermore $r_v < r_u$, since $I_v \subseteq I_u$. Therefore, $r_{u'} < l_v \le t_0 \le r_v < r_u$, i.e., $r_{u'} < t_0 < r_u$, where $t_0 \in T$. This is a contradiction to the assumption that both $u$ and $u'$ belong to the same set $U_{ji}$.

Now suppose that $I_v \cap T = \emptyset$. If $v \in D$ then both its endpoints belong to $R \cup T \subseteq T$, and thus $I_v \cap T \neq \emptyset$, which is a contradiction. Thus $v \in V^\# \setminus D = A \cup U^\#$. Let $v \in U^\#$, i.e., $v \in U_{j'i'}$, for some $1 \le j' \le i' \le |T|$. Note that $j' \neq i'$ by Lemma 22, and thus $j' < i'$. Thus, it follows by equation (4) that $l_v < t_{j'} \le t_{i'-1} < r_v$, i.e., $I_v \cap T \neq \emptyset$, which is a contradiction. Therefore, $v \in U^\#$, and thus $v \in A$. If $v \in \bigcup_{i,x}\{v_{i,x}^1, v_{i,x}^2, v_{i,x}^{q(i,x)-1}, v_{i,x}^{q(i,x)}\}$, then both its endpoints belong to $T$ (by the definition of the set $T$), i.e., $I_v \cap T \neq \emptyset$, which is a contradiction.

Therefore $v = v_{i,x}^h$, for some $i \in [k+1]$, $x \in [p_i]$, and $3 \le h \le q(i,x) - 2$. Recall that $v_{i,x}^1 <_{\sigma^\#} v_{i,x}^2 <_{\sigma^\#} v_{i,x}^h <_{\sigma^\#} v_{i,x}^{q(i,x)-1} <_{\sigma^\#} v_{i,x}^{q(i,x)}$. Furthermore recall that the intervals $\{I_z : z \in U_{i,x}^{**}\}$ in the interval representation $\mathcal{I}$ of the initial graph $G$ have been replaced by the intervals $\{I_{i,x}^h : 1 \le h \le q(i,x)\}$ in the interval representation $\mathcal{I}^\#$ of the graph $G^\#$. Suppose that $l_u < l_{v_{i,x}^1}$. Then, since $I_v = I_{v_{i,x}^h} \subseteq I_u$ by assumption, it follows that $I_u$ properly contains in the representation $\mathcal{I}^\#$ all three intervals of the vertices $v_{i,x}^1$, $v_{i,x}^2$, and $v = v_{i,x}^h$. Therefore the interval $I_u$ properly contains in the initial representation $\mathcal{I}$ all triples of intervals $\{I_{z_1}, I_{z_2}, I_{z_h}\}$, where $z_1 \in C_{i,x}^1$, $z_2 \in C_{i,x}^2$, and $z_h \in C_{i,x}^h$. Thus, since $z_i, z_2, z_3$ induce an independent set (they belong to different connected components $C_{i,x}^1, C_{i,x}^2, C_{i,x}^h$ of $U_{i,x}^{**}$ in the initial graph $G$), it follows that the vertices $\{u, z_1, z_2, z_h\}$ induce a $K_{1,3}$ in $G \setminus D$. This is a contradiction to the assumption that $D$ is a proper interval deletion set of $G$ [48]. Thus $l_{v_{i,x}^1} < l_u$. Suppose that $r_{v_{i,x}^{q(i,x)}} < r_u$. Then it follows similarly that $I_u$ properly contains in the initial representation $\mathcal{I}$ all triples of intervals $\{I_{z_h}, I_{z_{q(i,x)-1}}, I_{z_{q(i,x)}}\}$, where $z_h \in C_{i,x}^h$, $z_{q(i,x)-1} \in C_{i,x}^{q(i,x)-1}$, and $z_{q(i,x)} \in C_{i,x}^{q(i,x)}$, which is again a contradiction. Therefore, $l_{v_{i,x}^1} < l_u < r_u < r_{v_{i,x}^{q(i,x)}}$. That is, the interval $I_u$ is properly contained in the interval $\mathbf{span}(U_{i,x}^{**})$, and thus $u \in U_{i,x}^{**}$ (cf. the definition of the sets $U_{i,x}^*$ and $U_{i,x}^{**}$ in Section 3.1). Therefore, vertex $u$ has been replaced in the weighted graph $G^\#$ by a vertex of $A$. This is a contradiction, since $u \in U_{ji} \subseteq U^\#$ by assumption.

Thus for every $u \in U_{ji}$ and every $v \in V^\#$, if $I_v \subseteq I_u$, then $U_{ji} \subseteq N(v)$ in the graph $G^\#$. This proves Condition 2 of Definition 4. $\square$

Note that for every $1 \le j < i \le |T|$, the set $U_{ji}$ contains no vertices of $D$, since by definition $U_{ji} \subseteq U^\# = V^\# \setminus (D \cup A)$. Therefore, since all sets of $\mathcal{S}_2$ are disjoint, we can apply Reduction Rule 2 to the graph $G^\#$ with respect to the sets of $\mathcal{S}_2$, by replacing in the interval representation $\mathcal{I}^\#$ the intervals $\{I_v : v \in S\}$ with $\min\{|S|, |D|+4\}$ copies of the interval $I_S = \mathbf{span}(S)$, for every $S \in \mathcal{S}_2$. Denote the resulting weighted graph by $\widehat{G} = (\widehat{V}, \widehat{E})$ and its interval representation by $\widehat{\mathcal{I}}$. Then the next corollary follows immediately by Theorem 4.

**Corollary 3.** *The maximum weight of a path in $G^\#$ is equal to the maximum weight of a path in $\widehat{G}$.*

In the next two theorems we provide the main results of this section. In particular, in Theorem 6 we prove that the constructed weighted interval graph $\widehat{G}$ is a special weighted interval graph with a parameter $\kappa$ that is upper bounded by $O(k^3)$ and in Theorem 7 we provide a time bound of $O(k^2 n)$ for computing $\widehat{G} = (\widehat{V}, \widehat{E})$ and a special vertex partition $\widehat{V} = A \cup B$.

**Theorem 6.** *The weighted interval graph $\widehat{G} = (\widehat{V}, \widehat{E})$ is a* special weighted interval graph *with* parameter $\kappa = O(k^3)$.

**Proof.** Define $A = V(G^\#) \setminus V(G)$, i.e., $A$ is the set of vertices that have been introduced in the weighted interval graph $G^\#$ by applying Reduction Rule 1 to the initial (unweighted) interval graph $G$ (cf. Section 3.1). Note that the vertices of $A$ also belong to the weighted graph $\widehat{G}$, since they are not affected by the application of Reduction Rule 2 to the graph $G^\#$. Furthermore, we define the vertex set $B = \widehat{V} \setminus A$, i.e., $\widehat{V}$ is partitioned into the sets $A$ and $B$.

We will prove that $A$ and $B$ satisfy the three conditions of Definition 5. Since the vertices of $A$ are not affected by the application of Reduction Rule 2, Corollary 2 implies that $A$ induces an independent set in $\widehat{G}$. This proves Condition 1 of Definition 5.

Let $v_{i,x}^t \in A$ and $u \in \widehat{V} \setminus \{v_{i,x}^t\}$. Assume that $I_u \subseteq I_{i,x}^t$ in the interval representation $\widehat{\mathcal{I}}$. If $u$ is also a vertex of the weighted graph $G^\#$, then Corollary 2 implies that $I_u \not\subseteq I_{i,x}^t$ in the interval representation $\mathcal{I}^\#$ (and thus also in the representation $\widehat{\mathcal{I}}$). This is a contradiction to the assumption that $I_u \subseteq I_{i,x}^t$ in $\widehat{\mathcal{I}}$. Otherwise, if $u$ is a vertex of $\widehat{G}$ but not a vertex of $G^\#$, then $I_u = \mathbf{span}(S)$, for some $S \in \mathcal{S}_2$. Therefore, for every vertex $u' \in S$, we have that $I_{u'} \subseteq \mathbf{span}(S) = I_u \subseteq I_{i,x}^t$ in the interval representation $\mathcal{I}^\#$ of the graph $G^\#$. This is a contradiction by Corollary 2. Therefore, for every $v_{i,x}^t \in A$ and every $u \in \widehat{V} \setminus \{v_{i,x}^t\}$, we have $I_u \subseteq I_{i,x}^t$ in the interval representation $\widehat{\mathcal{I}}$. This proves Condition 2 of Definition 5.

Recall by Corollary 2 that the set $V^\# \setminus D$ is partitioned into the sets $A$ and $U^\#$. Furthermore, recall that $\{U_{ji} : 1 \le j < i \le |T|\}$ provides a partition of $U^\#$, and that each of these vertex sets $U_{ji}$ is replaced in the graph $\widehat{G}$ by at most $\min\{|U_{ji}|, |D| + 4\}$ vertices. Thus the vertex set $B$ contains all vertices of $D$ and at most $|D| + 4$ vertices for each of the vertex subsets $\{U_{ji} : 1 \le j < i \le |T|\}$ of $G^\#$. Recall that $|T| \le 18k + 16$, and thus there exist at most $\binom{18k+16}{2}$ different sets $U_{ji}$. Furthermore, recall that $D = \{d_0, d_1, \dots, d_k, d_{k+1}\}$, i.e., $|D| = k + 2$. Therefore, $|B| \le |D| + \binom{18k+16}{2} \cdot (|D| + 4) = (k + 2) + \binom{18k+16}{2} \cdot (k + 6) = O(k^3)$. This proves Condition 3 of Definition 5 and completes the proof of the theorem. □

**Theorem 7.** *Let $G = (V, E)$ be an interval graph, where $|V| = n$. Let $D = \{d_0, d_1, \dots, d_k, d_{k+1}\}$ be a proper interval deletion set of $G$. Then the special weighted interval graph $\widehat{G} = (\widehat{V}, \widehat{E})$ and a special vertex partition $\widehat{V} = A \cup B$ can be computed in $O(k^2 n)$ time.*

**Proof.** First recall that the graph $G^\# = (V^\#, E^\#)$ and the independent set $A \subseteq V^\#$ can be computed in $O(n)$ time by Theorem 5. Furthermore, recall by the proof of Theorem 5 that, during the computation of the interval representation $\mathcal{I}^\#$ of the graph $G^\#$, we also compute the points of $R \cup L$ and the intervals $I_{i,x}^t = \mathbf{span}(C_{i,x}^t)$ for the connected components $C_{i,x}^t$ of $U_{i,x}^{**}$, where $i \in [k + 1]$, $p_i \in [p_i]$, and $t \in [q(i, x)]$. Thus, since all these computations can be done in $O(n)$ time, we can also compute the set $T$ of endpoints in the interval representation $\mathcal{I}^\#$ in $O(n)$ time in total.

Now, for every pair $\{j, i\}$ such that $1 \le j < i \le |T|$, we can compute the set $U_{ji}$ in $O(n)$ time by visiting each vertex $u \in U^\#$ once and by checking whether $t_{j-1} < l_u < t_j$ and $t_{i-1} < r_u < t_i$ (cf. the definition of the sets $U_{ji}$). Furthermore, we can compute in the same time the interval $\mathbf{span}(U_{ji})$ by keeping the leftmost left endpoint and the rightmost right endpoint of $U_{ji}$, respectively. Thus, since there are $\binom{|T|}{2} \le \binom{18k+16}{2} = O(k^2)$ such pairs of indices $\{j, i\}$, all sets $U_{j,i}$ and all intervals $\mathbf{span}(U_{ji})$ can be computed in $O(k^2 n)$ time in total.

Once we have computed all intervals $\mathbf{span}(U_{ji})$, we can iteratively remove from the representation $\mathcal{I}^\#$ the intervals of the vertices of $U_{ji}$ and replace them with $\min\{|U_{ji}|, |D| + 2\} \le |U_{ji}|$ copies of the interval $\mathbf{span}(U_{ji})$, resulting thus at the interval representation $\widehat{\mathcal{I}}$ of $\widehat{G}$. Since the number of vertices in all these sets $U_{ji}$ is at most $n$, all these replacements can be done in $O(n)$ time in total. Finally, since the set $A$ can be computed in $O(n)$ time by Theorem 5, the set $B = \widehat{V} \setminus A$ can be also computed in $O(n)$ time. Summarizing, the interval representation $\widehat{\mathcal{I}}$ of $\widehat{G}$ and the special vertex partition $\widehat{V} = A \cup B$ can be computed in $O(k^2 n) + O(n) = O(k^2 n)$ time in total. □

Note here that, although $\widehat{G} = (\widehat{V}, \widehat{E})$ is a special weighted interval graph with a parameter $\kappa$ that depends only on the size of $D$ by Theorem 6, $\widehat{G}$ may still have $O(n)$ vertices, as the independent set $A$ in its special vertex partition $\widehat{V} = A \cup B$ may be arbitrarily large.

## 4. Parameterized longest path on interval graphs

In this section, we first present Algorithm 1 (cf. Section 4.1) which computes in $O(\kappa^3 n)$ time the maximum weight of a path in a *special weighted interval graph* with parameter $\kappa$ (cf. Definition 5). Then, using Algorithm 1 and the results of Sections 2 and 3, we conclude in Section 4.2 with our fixed-parameter algorithm for Longest Path on Interval Graphs, where the parameter $k$ is the size of a minimum proper interval deletion set $D$. Since Algorithm 1 can be implemented to run in $O(\kappa^3 n)$ time and $\kappa = O(k^3)$ by Theorem 6, the algorithm of Section 4.2 runs in $O(k^9 n)$ time.

### *4.1. The algorithm for special weighted interval graphs*

Consider a *special weighted interval graph* $G = (V, E)$ with parameter $\kappa \in \mathbb{N}$, which is given along with a special interval representation $\mathcal{I}$ and a special vertex partition $V = A \cup B$. Recall by Definition 5 that $A$ is an independent set and that $|B| \le \kappa$. Let $w : V \to \mathbb{N}$ be the vertex weight function of $G$. Now we add to the set $B$ an isolated dummy vertex $v_0$ such that $v_0 <_\sigma v_1$ and $w(v_0) = 0$. Thus, after the addition of $v_0$ to $G$, we have $|B| \le \kappa + 1$. Note that $v_0$ is not contained in any maximum-weight path of this augmented graph. Thus, every maximum-weight path in the augmented graph is also a maximum weight path in $G$, and vice versa. In the following we denote this augmented graph by $G$. Furthermore, denote by $\mathcal{I}$ the augmented interval representation and by $\sigma = (v_0, v_1, v_2, \ldots, v_n)$ its right-endpoint ordering. For every vertex $v \in B$, we define

$$\xi_v = \begin{cases} l_u & \text{if } l_v \in I_u \text{ for some } u \in A, \\ l_v & \text{otherwise.} \end{cases}$$

**Lemma 24.** *For every vertex $v \in B$, $\xi_v$ is well-defined.*

**Proof.** It is enough to prove that if there exists a vertex $u \in A$ such that $l_v \in I_u$ then $u$ is unique. Let us assume to the contrary that there exist two distinct vertices $u$ and $u'$ in $A$ such that $l_v \in I_u$ and $l_v \in I_{u'}$. Then $I_u \cap I_{u'} \neq \emptyset$ and $uu' \in E(G)$. This is a contradiction, since $A$ is an independent set. □

Now we define the set $\Xi$ as

$$\Xi = \{\xi_v, l_v : v \in B\}. \tag{6}$$

Note that $|\Xi| \leq 2|B| \leq 2(\kappa + 1)$. Furthermore, let $u, v \in V$, where $u \in N(v)$ and $u <_\sigma v$. We define the vertex

$$\pi_{u,v} = \max_\sigma\{\{u\} \cup \{w \in B \cap N(u) : u <_\sigma w <_\sigma v\}\} \tag{7}$$

Note that, by definition, if $\pi_{u,v} \neq u$ then $\pi_{u,v} \in B$. Furthermore, due to the condition that $u \in N(v)$ in the definition of the vertex $\pi_{u,v}$, it follows that $u \in B$ or $v \in B$, since $A$ is an independent set. That is, vertex $\pi_{u,v}$ is defined for at most $2(\kappa + 1)(n + 1) = O(\kappa n)$ pairs of vertices $u, v$.

**Definition 6.** Let $\xi \in \Xi$ and $i \in [n]$ such that $\xi < r_{v_i}$. We define the induced subgraph $G_\xi(v_i) = G[\{v \in V : \xi \leq l_v < r_v \leq r_{v_i}\}]$ of $G$ which contains all vertices whose intervals (in the representation $\mathcal{I}$ of $G$) are entirely contained between the points $\xi$ and $r_{v_i}$.

Note by Definition 6 that, if $l_{v_i} < \xi$, then the vertex $v_i$ does not belong to the subgraph $G_\xi(v_i)$.

**Notation 1.** Let $\xi \in \Xi$ and $i \in [n]$ such that $\xi < r_{v_i}$. Furthermore, let $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. We denote by $P_\xi(v_i, y)$ a *maximum weight normal path* of $G_\xi(v_i)$, among those normal paths whose last vertex is $y$. For every path $P_\xi(v_i, y)$, we denote its weight $w(P_\xi(v_i, y))$ by $W_\xi(v_i, y)$.

Before we present Algorithm 1, we first present some auxiliary technical lemmas (cf. Lemmas 25–29) that will be useful in the proof of correctness and the running time analysis of the algorithm (cf. Theorems 8 and 9, respectively).

**Lemma 25.** *Let $\xi \in \Xi$ and $i \in [n]$, where $\xi < r_{v_i}$, and let $y \in V(G_\xi(v_i))$ and $y \in N(v_i)$. Let also $P$ be a normal path of $G_\xi(v_i)$ that has $y$ as its last vertex. If $v_i \notin V(P)$, then $P$ is a path of $G_\xi(\pi_{y,v_i})$.*

**Proof.** Let $y'$ denote the rightmost vertex of $P$ (in the ordering $\sigma$). Since $v_i \notin V(P)$, we have that $y' \neq v_i$. Note also that either $y = y'$ or $y <_\sigma y'$. We claim that $y' \leq_\sigma \pi_{y,v_i}$. Notice that the statement trivially holds if $y = y'$. Thus, it is enough to prove that $y' \leq_\sigma \pi_{y,v_i}$ when $y <_\sigma y'$. First, as $y <_\sigma y'$ and $y' <_P y$, Lemma 3 implies that $y'y \in E(G)$. Furthermore, since $P$ is normal, Lemma 5 implies that $\mathcal{I}[\{y, y'\}]$ does not induce a proper interval representation. Therefore, since $y <_\sigma y'$, it follows that $I_y \subseteq I_{y'}$, and thus $y' \in B$. Hence, since also $y' <_\sigma v_i$ and $y'y \in E(G)$, it follows that $y' \leq_\sigma \pi_{x,v_i}$. Therefore $P$ is a path of $G_\xi(\pi_{y,v_i})$. □

**Lemma 26.** *Let $\xi \in \Xi$ and $i \in [n]$, where $\xi < r_{v_i}$, and let $y \in V(G_\xi(v_i))$ and $y \in N(v_i)$. If $l_y < l_{v_i}$ or $v_i \notin V(G_\xi(v_i))$, then $W_\xi(v_i, y) = W_\xi(\pi_{y,v_i}, y)$.*

**Proof.** Let $l_y < l_{v_i}$ or $v_i \notin V(G_\xi(v_i))$. First we prove that $v_i \notin V(P_\xi(v_i, y))$. If $v_i \notin V(G_\xi(v_i))$, then clearly $v_i \notin V(P_\xi(v_i, y))$. Suppose now that $v_i \in V(G_\xi(v_i))$ and that $l_y < l_{v_i}$. For the sake of contradiction, assume that $v_i \in V(P_\xi(v_i, y))$. Since $l_y < l_{v_i}$ and $r_y < r_{v_i}$, note that $I_y \nsubseteq I_{v_i}$ and $I_{v_i} \nsubseteq I_y$. Thus $\mathcal{I}[\{y, v_i\}]$ induces a proper interval representation. Therefore, since $y <_\sigma v_i$ and $y, v_i \in V(P_\xi(v_i, y))$ by assumption, Lemma 5 implies that $y <_P v_i$. This is a contradiction to the assumption that $y$ is the last vertex of $P_\xi(v_i, y)$. Therefore $v_i \notin V(P_\xi(v_i, y))$.

Thus Lemma 25 implies that $V(P_\xi(v_i, y))$ is a path of $G_\xi(\pi_{y,v_i})$. Therefore, since $G_\xi(\pi_{y,v_i})$ is a subgraph of $G_\xi(v_i)$ (cf. Eq. (7)), it follows that $W_\xi(v_i, y) = W_\xi(\pi_{y,v_i}, y)$. □

**Lemma 27.** *Let $\xi \in \Xi$ and $i \in [n]$, where $\xi < r_{v_i}$, and let $v_i \in V(G_\xi(v_i))$. Then $P_\xi(v_i, v_i) = (P_1, v_i)$, where*

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < r_{v_i}\} \tag{8}$$

**Proof.** Let $P = (P_1, v_i)$ be a normal path of $G_\xi(v_i)$ such that $w(P) = W_\xi(v_i, v_i)$. Denote by $x$ the last vertex of $P_1$. Then, since $P$ is a normal path of $G_\xi(v_i)$, $P_1$ is a normal path of $G_\xi(v_i)$ that does not contain $v_i$. Lemma 25 implies that $P_1$ is a normal path of $G_\xi(\pi_{x,v_i})$ that has $x$ as its last vertex and hence

$$w(P_1) \le W_\xi(\pi_{x,v_i}, x).$$

We will now prove that $w(P_1) = W_\xi(\pi_{x,v_i}, x)$. For this, assume towards a contradiction that $w(P_1) < W_\xi(\pi_{x,v_i}, x)$. Recall that $P_\xi(\pi_{x,v_i}, x)$ is a normal path of $G_\xi(\pi_{x,v_i}) \subseteq G_\xi(v_i)$ that has $x$ as its last vertex. Since $xv_i \in E(G)$, this implies that $(P_\xi(\pi_{x,v_i}, x), v_i)$ is a path of $G_\xi(v_i)$ that has $v_i$ as its last vertex. Furthermore, since $v_i$ is the rightmost vertex of the path, it follows that $(P_\xi(\pi_{x,v_i}, x), v_i)$ is normal (Observation 1). Moreover,

$$w(P) = w(P_1) + w(v_i) < W_\xi(\pi_{x,v_i}, x) + w(v_i) = w((P_\xi(\pi_{x,v_i}, x), v_i)),$$

a contradiction to the assumption that $w(P) = W_\xi(v_i, v_i)$. Hence,

$$w(P_1) = W_\xi(\pi_{x,v_i}, x).$$

To conclude, $P_\xi(v_i, v_i) = (P_1, v_i)$, where

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} \le r_x \le r_{v_i}\}$$

and this completes the proof of the lemma. □

**Lemma 28.** *Let $\xi \in \Xi$ and $i \in [n]$, where $\xi < r_{v_i}$, and let $v_i, y \in V(G_\xi(v_i))$ and $y \in N(v_i)$. Let $\zeta \in \{l_y\} \cup \{\xi \in \Xi : l_{v_i} < \xi < l_y\}$ and $x \in V(G_\xi(v_i))$ be such that $l_{v_i} < r_x < \zeta$. Furthermore, let $P_1$ be a normal path of $G_\xi(\pi_{x,v_i})$ with $x$ as its last vertex and $P_2$ be a normal path of $G_\zeta(\pi_{y,v_i})$ with $y$ as its last vertex. Then $P = (P_1, v_i, P_2)$ is a normal path of $G_\xi(v_i)$ with $y$ as its last vertex.*

**Proof.** Since $V(P_2) \subseteq V(G_\zeta(\pi_{y,v_i})) = \{v \in V(G) : \zeta \le l_v \le r_v \le r_{\pi_{y,v_i}}\}$ it follows that $\zeta \le l_v$ for every vertex $v \in V(P_2)$. Therefore, since $r_x < \zeta$, it follows that

$$x <_\sigma v, \text{ for every } v \in V(P_2). \tag{9}$$

Therefore, since $P_1$ is normal, $x$ is the last vertex of $P_1$, and $x <_\sigma v$, $xv \notin E(G)$ for every $v \in V(P_2)$, it follows that $V(P_1) \cap V(P_2) = \emptyset$ (Lemma 3). Moreover, since $l_{v_i} < r_x$, $xv_i \in E(G)$ and since $\zeta \le l_v \le r_v \le r_{\pi_{y,v_i}} \le r_{v_i}$ it follows that $v_i v \in E(G)$ for every vertex in $V(P_2)$. Therefore, $(P_1, v_i, P_2)$ is a path that has $y$ as its last vertex (as $y$ is the last vertex of $P_2$). Moreover, since $V(P_1) \subseteq V(G_\xi(\pi_{x,v_i})) \subseteq V(G_\xi(v_i))$, $V(P_2) \subseteq V(G_\zeta(\pi_{y,v_i})) \subseteq V(G_\xi(\pi_{x,v_i}))$ and $v_i \in V(G_\xi(v_i))$, $P$ is a path of $G_\xi(v_i)$. It remains to show that $P$ is normal.

We first show that if $v_1$ is the first vertex of $P_1$, then $v_1 <_\sigma v$ for every vertex $v \in V(P) \setminus \{v_1\}$. Notice that $v_1 <_\sigma v$, for every vertex $v \in V(P_1) \setminus \{v\}$, since $P_1$ is a normal path and $v_1$ is its first vertex. Recall also that $x <_\sigma v$, for every vertex $v \in P_2 \cup \{v_i\}$ (equation (9)). As $v_1 <_\sigma x <_\sigma v$ for every vertex $v \in P_2 \cup \{v_i\}$, it indeed follows that $v_1 <_\sigma v$, for every vertex in $V(P) \setminus \{v_1\}$.

We now show that for every vertex $v \in V(P)$, with successor $v' \in V(P)$ and every vertex $u \in V(P)$ such that $v' <_P u$, and $vu \in E(G)$ it holds that $v' <_\sigma u$. Let us assume to the contrary that for some $v \in V(P)$, with successor $v' \in V(P)$ there exists a vertex $u \in V(P)$ such that $v' <_P u$, $vu \in E(G)$, and $u <_\sigma v'$. Notice that if $\{v, v', u\} \subseteq V(P_1)$ or $\{v, v', u\} \subseteq V(P_2)$, then we obtain a contradiction to the assumptions that $P_1$ and $P_2$ are normal paths. Similarly, if $v = v_i$ we obtain a contradiction to the fact that $P_2$ is a normal path since the successor of $v_i$ in $P$ is the first vertex of $P_2$. Moreover, as the only neighbor of $x$ in $V(P) \setminus V(P_1)$ is $v_i$ we obtain that $v \in V(P_1) \setminus \{x\}$ and $u \in V(P_2)$. Notice then that since $vu \in E(G)$, it holds that $r_v > l_u \ge \zeta > r_x$, and since $x <_\sigma v$ and $v <_P x$, as $P_1$ is normal from Lemma 3, $xv \in E(G)$. However, then $x <_\sigma u <_\sigma v'$, a contradiction to the assumption that $P_1$ is normal. Therefore, we conclude that for every vertex $v \in V(P)$, with successor $v' \in V(P)$ and every vertex $u \in V(P)$ such that $v' <_P u$, and $vu \in E(G)$ it holds that $v' <_\sigma u$. Thus, we completed the proof that $P$ is a normal path of $G_\xi(v_i)$ that has $y$ as its last vertex. □

**Lemma 29.** *Let $\xi \in \Xi$ and $i \in [n]$, where $\xi < r_{v_i}$, and let $v_i, y \in V(G_\xi(v_i))$ and $y \in N(v_i)$. Let $P_\xi(v_i, y) = (P_1, v_i, P_2)$. If $P_2 \neq (y)$, then there exists some $\zeta \in \Xi$, where $l_{v_i} < \zeta \le l_y$, such that*

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < \zeta\}, \tag{10}$$

$$w(P_2) = W_\zeta(\pi_{y,v_i}, y). \tag{11}$$

*Otherwise, if $P_2 = (y)$ then $l_{v_i} < l_y$ and*

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < l_y\}. \tag{12}$$

**Proof.** Denote $P = P_\xi(v_i, y)$. Notice first that, since $v_i$ is the rightmost vertex of $P$, the path $P_1$ is not empty by Observation 2. Denote by $x$ the last vertex of $P_1$. Then, since $P_1$ is the prefix of the normal path $P$, observe that $P_1$ is a normal path of $G_\xi(v_i)$ that has $x$ as its last vertex and does not contain $v_i$. Furthermore, $P_1$ is a path of $G_\xi(\pi_{x,v_i})$ by Lemma 25. Therefore $P_1$ is a normal path of $G_\xi(\pi_{x,v_i})$, and thus,

$$w(P_1) \leq W_\xi(\pi_{x,v_i}, x).$$

Let $v \in V(P_2)$. Since $v_i <_P v$ and $v <_\sigma v_i$, Lemma 3 implies that $v_i v \in E(G)$. Therefore, again since $v_i <_P v$ and $v <_\sigma v_i$, Lemma 5 implies that $\mathcal{I}[v, v_i]$ does not induce a proper interval representation, i.e., either $I_v \subseteq I_{v_i}$ or $I_{v_i} \subseteq I_v$. Thus, since $v <_\sigma v_i$, it follows that $I_v \subseteq I_{v_i}$. That is, $I_v \subseteq I_{v_i}$ for every $v \in V(P_2)$.

Therefore, since $y \in V(P_2)$, it follows that $I_y \subseteq I_{v_i}$, and thus in particular $l_{v_i} < l_y$. Now let

$$\zeta = \begin{cases} \min\{l_v \in \Xi : \ v \in V(P_2)\} & \text{if } P_2 \neq (y), \\ l_y & \text{otherwise.} \end{cases}$$

We show that, if $P_2 \neq (y)$, then $\{l_v \in \Xi : \ v \in V(P_2)\} \neq \emptyset$, and thus $\zeta$ is well-defined. Notice first that, if $y \in B$, then $l_y \in \Xi$. Let $y \in A$, then let $y'$ be the neighbor of $y$ in $P_2$ (note that $y'$ always exists since $P_2 \neq (y)$). Then, as $A$ is an independent set, it follows that $y' \in B$, and thus $l_{y'} \in B$. Hence, if $P_2 \neq (y)$, then in any case $\{l_v \in \Xi : \ v \in V(P_2)\} \neq \emptyset$.

Suppose that $\zeta < r_x$. Then, by definition of $\zeta$, there exists some $v \in V(P_2)$ such that $\zeta = l_v < r_x$. Let $r_x < r_v$, i.e., $l_v < r_x < r_v$. Then $xv \in E(G)$. Therefore, since $v <_\sigma v_i$ for every $v \in V(P_2)$, it follows by the normality of $P$ that $v_i$ is not the next vertex of $x$ in $P$, which is a contradiction. Let $r_v < r_x$, i.e., $v <_\sigma x$. Then, since $x <_P v$, Lemma 3 implies that $xv \in E(G)$, which is again a contradiction by the normality of $P$. Therefore $r_x < \zeta$. Now note that $l_{v_i} < r_x$, since $xv_i \in E(G)$. That is, $l_{v_i} < r_x < \zeta$. Therefore, since $\zeta \leq l_y$ by the definition of $\zeta$, it follows that

$$l_{v_i} < r_x < \zeta \leq l_y.$$

Let now $P_2 \neq (y)$. We prove that $\zeta \leq l_v$, for every $v \in V(P_2)$. Assume otherwise that there exists a vertex $v \in V(P_2)$ such that $l_v < \zeta$. Then, by the definition of $\zeta$, it follows that $l_v \notin \Xi$. Therefore $v \notin B$, and thus $v \in A$ (cf. the definition of the set $\Xi$). Since $P_2 \neq (y)$, it follows that $v$ has at least one neighbor $u$ in $P_2$. Then $u \in B$, since $A$ is an independent set. Furthermore, $l_v < \zeta \leq l_u$. Therefore, since $uv \in E(G)$ by assumption, it follows that $l_u \in I_v$. That is, $\xi_u = l_v \in \Xi$ (cf. the definition of $\xi_u$ for a vertex $u \in B$). Therefore $\zeta \leq \xi_u = l_v$, which is a contradiction to our assumption. Thus $\zeta \leq l_v$, for every $v \in V(P_2)$.

Now recall that $I_v \subseteq I_{v_i}$ for every $v \in V(P_2)$, as we proved above, and thus $v_i v \in E(G)$ for every $v \in V(P_2)$. Furthermore, recall that all vertices of $P_2$ appear in $P$ after vertex $v_i$. Therefore, since $P$ is a normal path by assumption, it follows that $P_2$ is also a normal path.

Thus, since $\zeta \leq l_v$ for every $v \in V(P_2)$, as we proved above, it follows that $P_2$ is a normal path of $G_\zeta(v_i)$ that does not contain $v_i$. Therefore Lemma 25 implies that $P_2$ is a path of $G_\zeta(\pi_{y,v_i})$. Thus, since $y$ is the last vertex of $P_2$, it follows that

$$w(P_2) \leq W_\zeta(\pi_{y,v_i}, y).$$

In the remainder of the proof we show that $w(P_1) = W_\xi(\pi_{x,v_i}, x)$ and $w(P_2) = W_\zeta(\pi_{y,v_i}, y)$. Towards a contradiction assume that at least one of the equalities does not hold. Notice first that from Lemma 28, $P = (P_\xi(\pi_{x,v_i}, x), v_i, P_\zeta(\pi_{y,v_i}, y))$ is a normal path of $G_\xi(v_i)$ that has $y$ as its last vertex. Notice now that

$$W_\xi(v_i, y) = w(P_1) + w(v_i) + w(P_2) < W_\xi(\pi_{x,v_i}, x) + w(v_i) + W_\zeta(\pi_{y,v_i}, y),$$

a contradiction. Therefore,

$$w(P_1) = W_\xi(\pi_{x,v_i}, x)$$

and

$$w(P_2) = W_\zeta(\pi_{y,v_i}, y).$$

Summarizing, if $P_2 \neq (y)$ we obtain that

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < \zeta\},$$

$$w(P_2) = W_\zeta(\pi_{y,v_i}, y),$$

and if $P_2 = (y)$ we obtain that

$$w(P_1) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < l_y\}.$$

This completes the proof of the lemma. □

We are now ready to present Algorithm 1, which computes the maximum weight of a path in a given *special weighted* interval graph $G$. It is easy to check that Algorithm 1 can be slightly modified such that it returns the actual path $P$ instead of its weight only.

First we give a brief overview of the algorithm. Using dynamic programming, it computes a 3-dimensional table. In this table, for every point $\xi \in \Xi$, every index $i \in [n]$, and every vertex $y \in V(G_\xi(v_i))$, where $\xi < r_{v_i}$ and $y \in N(v_i)$, the entry $W_\xi(v_i, y)$ (resp. the entry $W_\xi(v_i, v_i)$) keeps the weight of a normal path in the subgraph $G_\xi(v_i)$ which is the largest among those normal paths whose last vertex is $y$ (resp. $v_i$). Thus, since $w(v_0) = 0$ for the dummy isolated vertex $v_0$ (cf. line 1 of the algorithm), the maximum weight of a path in $G$ will be eventually stored in one of the entries $\left\{W_{l_{v_0}}(v_i, v_i) : 1 \leq i \leq n\right\}$ or in one of the entries $\left\{W_{l_{v_0}}(v_i, y) : 1 \leq i \leq n,\ y <_\sigma v_i,\ y \in N(v_i)\right\}$, depending on whether the last vertex $y$ of the desired maximum-weight path coincides with the rightmost vertex $v_i$ of this path in the ordering $\sigma$ (cf. line 18 of the algorithm).

Note that for every computed entry $W_\xi(v_i, y)$ the vertices $v_i$ and $y$ are adjacent, and thus $v_i \in B$ or $y \in B$, since $A$ is an independent set. Thus, since $|B| = O(\kappa)$, there are at most $O(\kappa n)$ such eligible pairs of vertices $v_i, y$. Furthermore, since also $|\Xi| = O(\kappa)$, the computed 3-dimensional table stores at most $O(\kappa^2 n)$ entries $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$. From the *for*-loops of lines 2–3 of the algorithm and from the obvious inductive hypothesis we may assume that during the $\{i, \xi\}$th iteration all previous values $W_{\xi'}(v_{i'}, v_{i'})$ and $W_{\xi'}(v_{i'}, y')$, where $i' < i$ or $\xi' < \xi$, have been correctly computed at a previous iteration.

In the initialization phase for a particular pair $\{i, \xi\}$ (cf. lines 4–6) the algorithm computes some initial values for $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$. For a path with $v_i$ as its last vertex, we are only interested in the case where $v_i \in V(G_\xi(v_i))$; in this case we initialize $W_\xi(v_i, v_i) = w(v_i)$, cf. line 4. For a path with $y \neq v_i$ as its last vertex (cf. lines 5–6), we initialize $W_\xi(v_i, y) = W_\xi(\pi_{y,v_i}, y)$, since the path $P_\xi(\pi_{y,v_i}, y)$ is indeed a normal path of the graph $G_\xi(\pi_{y,v_i})$, which is an induced subgraph of $G_\xi(v_i)$.

For the induction step phase (cf. lines 7–17) the algorithm updates the initialized entries $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$ according to Lemmas 26–29. To update the value $W_\xi(v_i, v_i)$ we only need to consider the case where $v_i \in V(G_\xi(v_i))$; in this case $W_\xi(v_i, v_i)$ is updated in lines 7–9 according to Lemma 27. The values of $W_\xi(v_i, y)$, where $y \neq v_i$, are updated in lines 10–17. In particular, in the case where $l_y < l_{v_i}$ or $v_i \notin V(G_\xi(v_i))$, the value of $W_\xi(v_i, y)$ is updated in lines 11–12 according to Lemma 26. Otherwise, $W_\xi(v_i, y)$ is updated in lines 14–17 according to Lemma 29.

The correctness of the algorithm is proved in Theorem 8 and its running time is proved in Theorem 9.

---

**Algorithm 1:** Computing a maximum-weight path of a special weighted interval graph.

---

**Input:** A special weighted interval graph $G = (V, E)$ with parameter $\kappa \in \mathbb{N}$, along with the special interval representation $\mathcal{I}$ of $G$ and the partition $V = A \cup B$, where $\sigma = (v_1, v_2, \ldots, v_n)$ is a right-endpoint ordering of $V$.
**Output:** The maximum weight of a path in $G$

1: Add an isolated dummy vertex $v_0$ with $w(v_0) = 0$ to set $B$, where $v_0 <_\sigma v_1$; denote $\sigma = (v_0, v_1, v_2, \ldots, v_n)$

2: **for** $i = 0$ to $n$ **do**
3:     **for** every $\xi \in \Xi$ where $\xi < r_{v_i}$ **do**

4:         **if** $v_i \in V(G_\xi(v_i))$ **then** $W_\xi(v_i, v_i) \leftarrow w(v_i)$ {initialization}

5:         **for** every $y \in V(G_\xi(v_i))$ where $y \in N(v_i)$ **do**
6:             $W_\xi(v_i, y) \leftarrow W_\xi(\pi_{y,v_i}, y)$ {initialization}

7:         **if** $v_i \in V(G_\xi(v_i))$ **then**
8:             $W_1 \leftarrow \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < r_{v_i}\}$
9:             $W_\xi(v_i, v_i) \leftarrow \max\{W_\xi(v_i, v_i), W_1 + w(v_i)\}$

10:         **for** every $y \in V(G_\xi(v_i))$ where $y \in N(v_i)$ **do**
11:             **if** $l_y < l_{v_i}$ or $v_i \notin V(G_\xi(v_i))$ **then**
12:                 $W_\xi(v_i, y) \leftarrow W_\xi(\pi_{y,v_i}, y)$
13:             **else**
14:                 $W'_1 \leftarrow \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < l_y\}$
15:                 **for** every $\zeta \in \Xi$ with $l_{v_i} < \zeta \leq l_y$ **do**
16:                     $W_1 \leftarrow \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < \zeta\}$
17:                     $W_\xi(v_i, y) \leftarrow \max\{W_\xi(v_i, y), W'_1 + w(v_i) + w(y), W_1 + w(v_i) + W_\zeta(\pi_{y,v_i}, y)\}$

18: **return** $\max\{W_{l_{v_0}}(v_i, v_i), W_{l_{v_0}}(v_i, y) : 1 \leq i \leq n,\ y <_\sigma v_i,\ y \in N(v_i)\}$

---

**Theorem 8.** *Let $G = (V, E)$ be a special weighted interval graph, given along with a special interval representation $\mathcal{I}$ and a special vertex partition $V = A \cup B$. Then Algorithm 1 computes the maximum weight of a path $P$ in $G$.*

**Proof.** In lines 2–17, Algorithm 1 iterates for every $i \in \{0, 1, 2, \ldots, n\}$ and for every $\xi \in \Xi$ such that $\xi < r_{v_i}$. For every such $i$ and $\xi$, the algorithm computes the values $W_\xi(v_i, v_i)$ and the values $W_\xi(v_i, y)$, for every vertex $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. We will prove by induction on $i$ that these values are the weights of the maximum-weight normal paths $P_\xi(v_i, v_i)$ and the values $P_\xi(v_i, y)$, respectively (cf. Notation 1).

For the induction basis, let $i = 0$. Then, since $v_0$ is an isolated vertex (cf. line 1 of Algorithm 1), the only $\xi \in \Xi$, for which $\xi < r_{v_0}$, is $\xi = l_{v_0}$. Then line 4 of the algorithm is executed and the algorithm correctly computes the value $W_\xi(v_0, v_0) = w(v_0) = 0$. Furthermore, since $v_0$ is a dummy vertex by assumption, the lines 5–6 and the lines 10–17 of the algorithm are not executed at all for $i = 0$. Finally, in lines 7–9 the algorithm recomputes the value $W_\xi(v_0, v_0) = w(v_0) = 0$, since there exists no vertex $x$ such that $l_{v_i} < r_x < r_{v_i}$ (cf. line 8 of the algorithm). This value of $W_\xi(v_0, v_0)$ is clearly correct. This completes the induction basis.

For the induction step, let $i \geq 1$. Consider the iteration of the algorithm for any $\xi \in \Xi$, where $\xi < r_{v_i}$. First the algorithm initializes in lines 4–6 the values $W_\xi(v_i, v_i)$ and the values $W_\xi(v_i, y)$, for every vertex $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. The initialization of line 4 is correct, since the single-vertex path $P = (v_i)$ is clearly a normal path of the graph $G_\xi(v_i)$ which has $v_i$ as its last vertex. The initialization of lines 5–6 is correct, since the path $P_\xi(\pi_{y,v_i}, y)$ is indeed a normal path of $G_\xi(\pi_{y,v_i})$, which is an induced subgraph of $G_\xi(v_i)$ (cf. Definition 6).

In lines 7–9 the algorithm updates the current (initialized) value of $W_\xi(v_i, v_i)$. The correctness of this update follows directly by Lemma 27. Furthermore, in lines 10–17 the algorithm iterates for every vertex $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. For every such value of $y$, the algorithm updates the current (initialized) value of $W_\xi(v_i, y)$.

The correctness of the update in line 12 follows directly by Lemma 26.

During the execution of lines 14–17 the algorithm deals with the case where $v_i \in V(G_\xi(v_i, y))$ and $l_{v_i} < l_y$. If $v_i$ does not belong to the desired path $P_\xi(v_i, y)$, then by Lemma 25 $P_\xi(v_i, y)$ is also a normal path of $G_\xi(\pi_{y,v_i})$, which is an induced subgraph of $G_\xi(v_i)$. Therefore, in this case, $W_\xi(v_i, y) = W_\xi(\pi_{y,v_i}, y)$. The algorithm does not update the current value of $W_\xi(v_i, y)$, since $W_\xi(v_i, y)$ has been initialized to $W_\xi(\pi_{y,v_i}, y)$ in line 6.

For the remainder of the proof, assume that $v_i$ belongs to the desired path $P_\xi(v_i, y)$, i.e., $P_\xi(v_i, y) = (P_1, v_i, P_2)$, for some sub-paths $P_1$ and $P_2$ of $P_\xi(v_i, y)$. In lines 14–16 the algorithm distinguishes between the cases where $P_2 = (y)$ and $P_2 \neq (y)$. To deal with the case where $P_2 = (y)$, i.e., with the case where $P_\xi(v_i, y) = (P_1, v_i, y)$, the algorithm computes in line 14 the value $W_1' = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < l_y\}$ of the desired path $P_1$ (cf. Eq. (12) of Lemma 29). Then it compares in line 17 the current value of $W_\xi(v_i, y)$ with the value $W_1' + w(v_i) + w(y)$, and it stores the greatest value between them in $W_\xi(v_i, y)$. This update is correct by Eq. (12) of Lemma 29.

To deal with the case where $P_2 \neq (y)$, the algorithm iterates in lines 15–16 for every $\zeta \in \Xi$ such that $l_{v_i} < \zeta \leq l_y$. For every such value of $\zeta$ it computes the value $W_1$ of the desired path $P_1$ (cf. Eq. (10) of Lemma 29). Then the algorithm compares in line 17 the current value of $W_\xi(v_i, y)$ with the value $W_1 + w(v_i) + W_\zeta(\pi_{y,v_i}, y)$ and it stores the greatest between them in $W_\xi(v_i, y)$. For every $\zeta \in \Xi$, where $l_{v_i} < \zeta \leq l_y$, Lemma 28 implies that the path $(P_\xi(\pi_{x,v_i}, x), v_i, P_\zeta(\pi_{y,v_i}, y))$ is a normal path of $G_\xi(v_i)$ with $y$ as its last vertex. Therefore $W_1 + w(v_i) + W_\zeta(\pi_{y,v_i}, y) \leq W_\xi(v_i, y)$, for every such value of $\zeta$. Furthermore Lemma 29 implies that there exists at least one such value $\zeta$, such that the values $W_1 = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)), l_{v_i} < r_x < \zeta\}$ and $W_\zeta(\pi_{y,v_i}, y)$ are equal to the weights of the sub-paths $P_1$ and $P_2$ of $P_\xi(v_i, y)$, respectively. Therefore, these updates of $W_\xi(v_i, y)$ for all values of $\zeta$ are correct. This completes the induction step.

Therefore, after the execution of lines 2–17, Algorithm 1 has correctly computed all values $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$, where $i \in \{0, 1, 2, \ldots, n\}$, $\xi \in \Xi$ such that $\xi < r_{v_i}$, and $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. Thus, since for every $i$ and every $\xi$ the graph $G_\xi(v_i)$ is an induced subgraph of $G_{l_{v_0}}(v_i)$, it follows that the maximum weight of a path in $G$ is one of the values $W_{l_{v_0}}(v_i, v_i)$ and $W_{l_{v_0}}(v_i, y)$. Therefore the algorithm returns the correct value in line 18. □

**Theorem 9.** *Let $G = (V, E)$ be a special weighted interval graph with $n$ vertices and parameter $\kappa$. Then Algorithm 1 can be implemented to run in $O(\kappa^3 n)$ time.*

**Proof.** Since $G$ is a special weighted interval graph with $V = A \cup B$ as its special vertex partition (cf. Definition 5), $A$ is an independent set and $|B| \leq \kappa + 1$ (after the addition of the dummy vertex $v_0$ to the set $B$). Recall that the endpoints of the intervals in $\mathcal{I}$ are given sorted increasingly, e.g., in a linked list $M$. The points $\{\xi_v : v \in B\}$ can be efficiently as follows. First we visit all endpoints of the intervals in $\mathcal{I}$ (in increasing order). For every endpoint $l_v$, where $v \in B$, which we visit between the endpoints $l_u$ and $r_u$, for some $u \in A$, we define $\xi_v = l_u$. Thus the points $\{\xi_v : v \in B\}$ can be computed in $O(n)$ time. Furthermore the points $\{l_v : v \in B\}$ can be computed in $O(\kappa)$ time by just enumerating all vertices of $B$. Therefore the set $\Xi$ can be computed in $O(\kappa + n) = O(n)$ time in total. Furthermore recall that there are $O(\kappa n)$ different vertices $\pi_{u,v}$ (cf. Eq. (7)) and note that, given two adjacent vertices $u, v$, we can compute the vertex $\pi_{u,v}$ in $O(\kappa)$ time by enumerating in worst case all vertices of $B$. Thus, all vertices $\pi_{u,v}$ can be computed in total $O(\kappa^2 n)$ time.

Now we provide an upper bound on the number of values $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$ that are computed by Algorithm 1. Since $\xi \in B$ and $v_i \in V$, there are in total at most $O(\kappa n)$ different values $W_\xi(v_i, v_i)$. Furthermore, the values $W_\xi(v_i, y)$, where $y \neq v_i$, are computed for every $i \in \{0, 1, 2, \ldots, n\}$, every $\xi \in \Xi$ such that $\xi < r_{v_i}$, and every $y \in V(G_\xi(v_i))$ such that $y \in N(v_i)$. Thus, due to the condition that $y \in N(v_i)$, it follows that $v_i \in B$ or $y \in B$. That is, there are in total at most $2(\kappa + 1)(n + 1) = O(\kappa n)$ pairs of vertices $v_i, y$ for which we compute the values $W_\xi(v_i, y)$. Therefore, since $\xi \in B$ and $|B| \leq \kappa + 1$, there are at most $O(\kappa^2 n)$ different values $W_\xi(v_i, y)$. Summarizing, Algorithm 1 computes at most $O(\kappa^2 n)$ different values $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$.

We now show that all computations performed in the lines 8, 14, and 16 of the algorithm can be implemented to run in total $O(\kappa^2 n)$ time. Denote by $Q = \{l_v, r_v : v \in V\}$ the set of all endpoints of the intervals in $\mathcal{I}$. Recall that the points of $Q$ are assumed to be already sorted increasingly. Note that, in order to perform all computations of the lines 8, 14, and 16, it suffices to store at each point $q \in Q$ the values

$$\omega_\xi(q, v_i) = \max\{W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)),\ l_{v_i} < r_x < q\} \tag{13}$$

for every $\xi \in \Xi$ and every $i \in [n]$ such that $\xi \leq l_{v_i} < q \leq r_{v_i}$. Indeed, once we have computed all possible values $\omega_\xi(q, v_i)$, lines 8, 14, and 16 of the algorithm can be executed in $O(1)$ time by just accessing the stored values $\omega_\xi(r_{v_i}, v_i)$, $\omega_\xi(l_y, v_i)$, and $\omega_\xi(\zeta, v_i)$, respectively. Observe that for every point $q \in Q$ such that $l_{v_i} < q \leq r_{v_i}$, the vertex which has $q$ as an endpoint is adjacent to vertex $v_i$. Thus, since there are at most $O(\kappa n)$ pairs of adjacent vertices in $G$, it follows that there are $O(\kappa n)$ such pairs of a point $q \in Q$ and a vertex $v_i \in V$.

Given a point $\xi \in \Xi$ and a vertex $v_i$, we can compute all values $\omega_\xi(q, v_i)$ in $O(|N(v_i)|)$ time as follows. Let $q_0 > l_{v_i}$ be the first endpoint after $l_{v_i}$ in the ordering of the endpoints in $Q$. As there does not exist any vertex $x$ such that $l_{v_i} < r_x < q_0$ (cf. Eq. (13)), we store at point $q_0$ the value $\omega_\xi(q_0, v_i) = 0$. Then, we visit in increasing order all points of $q \in Q$ between $q_0$ and $r_{v_i}$. Note that we can visit all these vertices in $O(|N(v_i)|)$ time as the points of $Q$ are already sorted increasingly. Let $q \in Q$ be the currently visited point between $q_0$ and $r_{v_i}$, and let $q'$ be the predecessor of $q$ in the ordering of $Q$. Then it follows by the definition of $\omega_\xi(q, v_i)$ in Eq. (13) that

$$\omega_\xi(q, v_i) = \max\{\omega_\xi(q', v_i),\ W_\xi(\pi_{x,v_i}, x) : x \in V(G_\xi(v_i)),\ q' \leq r_x < q\}. \tag{14}$$

Therefore, since $q'$ and $q$ are two consecutive points of $Q$ between $q_0$ and $r_{v_i}$, the value $\omega_\xi(q, v_i)$ can be computed in $O(1)$ time using the value of $\omega_\xi(q', v_i)$, as follows:

$$\omega_\xi(q, v_i) = \begin{cases} \max\{\omega_\xi(q', v_i),\ W_\xi(\pi_{x,v_i}, x_0)\} & \text{if } q' = r_{x_0}, \text{for some } x_0 \in V(G_\xi(v_i)) \\ \omega_\xi(q', v_i) & \text{otherwise} \end{cases}. \tag{15}$$

Since the value of $\omega_\xi(q, v_i)$ can be computed by Eq. (15) in $O(1)$ time, all these computations of the values $\{\omega_\xi(q, v_i) : q \in Q, l_{v_i} < q \leq r_{v_i}\}$ (for a fixed $\xi$ and a fixed $v_i$) can be performed in $O(|N(v_i)|)$ time in total. Thus, since $|\Xi| = O(\kappa)$ and $\sum_{i\in[n]} |N(v_i)| = O(\kappa n)$, we can compute all values $\omega_\xi(q, v_i)$ in total $O(\kappa^2 n)$ time. That is, all computations performed in the lines 8, 14, and 16 of the algorithm can be implemented to run in total $O(\kappa^2 n)$ time.

In the remainder of the proof we assume that each of the lines 8, 14, and 16 is executed in $O(1)$ time. Each of the lines 4, 8, and 9 is executed for every $i \in \{0, 1, \ldots, n\}$ and at most for every $\xi \in \Xi$, i.e., $O(\kappa n)$ times in total. Furthermore, each of the lines 6, 11, 12, and 14 is executed at most for every $\xi \in \Xi$ and for every pair $\{v_i, y\}$ of adjacent vertices in $G$, i.e., $O(\kappa^2 n)$ times in total. Each of the lines 16–17 is executed at most for every $\xi \in \Xi$, for every $\zeta \in \Xi$, and for every pair $\{v_i, y\}$ of adjacent vertices in $G$, i.e., $O(\kappa^3 n)$ times in total.

Finally, once we have computed all values $W_\xi(v_i, v_i)$ and $W_\xi(v_i, y)$ in lines 2–17, the output of line 18 can be computed in $O(\kappa n)$ time by considering the $O(\kappa n)$ computed values $W_{l_{v_0}}(v_i, v_i)$ and $W_{l_{v_0}}(v_i, y)$, for every vertex $v_i$ and for at most each pair of adjacent vertices $v_i, y$. Summarizing, Algorithm 1 can be implemented to run in total $O(\kappa^2 n + \kappa^3 n + \kappa n) = O(\kappa^3 n)$ time. $\square$

### 4.2. The general algorithm

Here we combine all our results of Sections 2, 3, and 4.1 to present our *parameterized linear-time* algorithm for Longest Path on Interval Graphs. The parameter $k$ of this algorithm is the size of a *minimum proper interval deletion set* $D$ of the input graph $G$ and its running time has a *polynomial dependency* on $k$.

**Theorem 10.** *Let $G = (V, E)$ be an interval graph, where $|V| = n$ and $|E| = m$, and let $k$ be the minimum size of a proper interval deletion set of $G$. Let $\mathcal{I}$ be an interval representation of $G$ whose endpoints are sorted increasingly. Then:*

1. *a proper interval deletion set $D$, where $|D| \leq 4k$, can be computed in $O(n + m)$ time,*
2. *a semi-proper interval representation $\mathcal{I}'$ of $G$ can be constructed in $O(n + m)$ time, and*
3. *given $D$ and $\mathcal{I}'$, a longest path of $G$ can be computed in $O(k^9 n)$ time.*

**Proof.** The first two statements of the theorem follow immediately by Theorems 1 and 2, respectively. For the remainder of the proof we assume that the proper interval deletion set $D$ and the semi-proper interval representation $\mathcal{I}'$ of $G$ have been already computed.

For the third statement of the theorem, we first compute the weighted interval graph $G^\# = (V^\#, E^\#)$ in $O(n)$ time by Theorem 5. Then, given the graph $G^\#$, we compute the weighted interval graph $\widehat{G} = (\widehat{V}, \widehat{E})$ in $O(k^2 n)$ time by Theorem 6. By Theorem 6, this graph $\widehat{G}$ is a special weighted interval graph with parameter $\kappa = O(k^3)$, cf. Definition 5. During the computation of the graph $\widehat{G}$, we can compute in the same time (i.e., in $O(k^2 n)$ time) also a special vertex partition $\widehat{V} = A \cup B$ of its vertex set. Furthermore, it follows by Corollaries 1 and 3 that the maximum number of vertices of a path in the initial interval graph $G$ is equal to the maximum weight of a path in the special weighted interval graph $\widehat{G}$. Therefore, in order to compute a longest path in $G$ it suffices to compute a path of maximum weight in $\widehat{G}$. Thus, since $\widehat{G}$ is a special weighted interval graph with parameter $\kappa = O(k^3)$, we compute the maximum weight of a path in $\widehat{G}$ by Algorithm 1. The running time of Algorithm 1 with input $\widehat{G}$ is $O(\kappa^3 n) = O(k^9 n)$ by Theorem 9. $\square$

## 5. Kernelization of Maximum Matching

For the sake of completeness, in this section we present the details of the algorithm for Maximum Matching[2] that we sketched in Section 1. The parameter $k$ is the solution size; for this parameter we show that a *kernel* with at most $O(k^2)$ vertices and edges can be computed in $O(kn)$ time, thus leading to a total running time of $O(kn+k^3)$. Hence, Maximum Matching, parameterized by the solution size, belongs to the class PL-FPT. First we present two simple data reduction rules, very similar in spirit to the data reduction rules of Buss for Vertex Cover (see e.g., [22,44]).

**Reduction Rule 3.** *If* $\deg(v) > 2(k-1)$ *for some vertex* $v \in V(G)$*, then return the instance* $(G \setminus \{v\}, k-1)$*.*

**Reduction Rule 4.** *If* $\deg(v) = 0$ *for some vertex* $v \in V(G)$*, then return the instance* $(G \setminus \{v\}, k)$*.*

An instance of parameterized Maximum Matching is called *reduced* if none of Reduction Rules 3 and 4 can be applied to this instance. It can be easily checked that Reduction Rule 4 is safe. In the next lemma we show that Reduction Rule 3 is also safe.

**Lemma 30.** *Let $k$ be a positive integer and $G$ be a graph. If $v \in V(G)$ such that* $\deg(v) > 2(k-1)$*, then $(G \setminus \{v\}, k-1)$ is a* yes*-instance if and only if $(G,k)$ is a* yes*-instance.*

**Proof.** We first show that if $(G,k)$ is a yes-instance then $(G \setminus \{v\}, k-1)$ is a yes-instance. For this, let $M$ be a matching of $G$ of size at least $k$. If $v \notin V(M)$, then $V(M) \subseteq V(G \setminus \{v\})$ and hence $M \subseteq E(G \setminus \{v\})$. Therefore, $M$ is a matching of $G \setminus \{v\}$ of size at least $k$ and thus $(G \setminus \{v\}, k-1)$ is a yes-instance. If $v \in V(M)$, then there exists a unique edge $e \in M$ such that $e = uv$. This implies that $V(M \setminus \{e\}) \subseteq V(G \setminus \{v\})$ and $M \setminus \{e\} \subseteq E(G \setminus \{v\})$. Therefore, $M' = M \setminus \{e\}$ is a matching of $G \setminus \{v\}$ and $|M'| = |M| - 1 \geq k-1$. Thus, $(G \setminus \{v\}, k-1)$ is again a yes-instance.

We now show that if $(G \setminus \{v\}, k-1)$ is a yes-instance, then $(G,k)$ is also a yes-instance. Let $M'$ be a matching of $G \setminus \{v\}$ of size at least $k-1$. Note that, since $G \setminus \{v\} \subseteq G$, any matching of $G \setminus \{v\}$ is also a matching of $G$. If $|M'| \geq k$, then $(G,k)$ is clearly a yes-instance, since $M'$ is a matching of $G$ of size at least $k$. Suppose now that $|M'| = k-1$, that is, $|V(M')| = 2(k-1)$. Then, since $\deg(v) > 2(k-1)$ in the graph $G$ by assumption, there exists at least one vertex $u \in N(v) \setminus V(M')$. Thus, since also $v \notin M'$ (as $M'$ is a matching of $G \setminus \{v\}$), it follows that the edge set $M = M' \cup \{uv\}$ is a matching of $G$ and $|M| = |M' \cup \{uv\}| = |M'| + 1 = k$. Thus, $(G,k)$ is a yes-instance. $\square$

In the following, $\mathbf{mm}(G)$ denotes the size of a maximum matching of graph $G$. Furthermore, for every subset $S \subseteq V$ we denote $N(S) = \bigcup_{v \in S} N(v)$.

**Lemma 31.** *Let $G$ be a graph. If* $1 \leq \deg(v) \leq 2(k-1)$ *for every $v \in V(G)$, then* $|V(G)|, |E(G)| \leq (2k-1) \cdot \mathbf{mm}(G)$*.*

**Proof.** Let $m_0 = \mathbf{mm}(G)$ and let $M = \{u_i v_i : i \in [m_0]\}$ be a maximum matching of $G$. Then $V(M)$ is a vertex cover of $G$, and thus $v \in N(V(M))$ for every vertex $v \notin V(M)$. Now suppose that there exists some $i \in [m_0]$ such that $N(u_i) \setminus (N(v_i) \cup V(M)) \neq \emptyset$ and $N(v_i) \setminus (N(u_i) \cup V(M)) \neq \emptyset$. Let then $z \in N(u_i) \setminus (N(v_i) \cup V(M))$ and $y \in N(v_i) \setminus (N(u_i) \cup V(M))$. Then there exists the alternating path $(z, u_i, v_i, y)$, and thus the set $M' = (M \setminus \{u_i v_i\}) \cup \{u_i z, v_i y\}$ is a matching of $G$ of size $m_0 + 1$, which is a contradiction to the maximality of $M$. Therefore, for every $i \in [m_0]$ we have that $N(u_i) \subseteq N(v_i) \cup V(M)$ or $N(v_i) \subseteq N(u_i) \cup V(M)$. Without loss of generality, we assume in the following that $N(v_i) \subseteq N(u_i) \cup V(M)$, for every $i \in [m_0]$.

Let now $w \in V(G) \setminus V(M)$. Then $w \in N(V(M))$, since $V(M)$ is a vertex cover of $G$. Hence, since $N(v_i) \subseteq N(u_i) \cup V(M)$, for every $i \in [m_0]$, it follows that there exists at least one $i_0 \in [m_0]$ such that $w \in N(u_{i_0})$. That is, the set $\{u_i : i \in [m_0]\}$ is also a vertex cover of $G$. Since $\deg(u_i) \leq 2k-2$ by assumption, it follows that $|V(G) \setminus V(M)| \leq (2k-3)m_0$. This implies $|V(G)| \leq (2k-1)m_0$ since $|V(M)| = 2m_0$.

Finally, since $\{u_i : i \in [m_0]\}$ is a vertex cover and $\deg(u_i) \leq 2(k-1)$ for every $i \in [m_0]$, it follows that $|E(G)| \leq (2k-1)m_0$. $\square$

Now we are ready to provide our kernelization algorithm for Maximum Matching, together with upper bounds on its running time and on the size of the resulting kernel.

**Theorem 11.** Maximum Matching*, when parameterized by the solution size $k$, admits a kernel with at most $O(k^2)$ vertices and at most $O(k^2)$ edges. For an $n$-vertex graph the kernel can be computed in $O(kn)$ time.*

[2] Given a graph $G$, find a maximum-cardinality matching in $G$; in its decision version, additionally the desired matching size $k$ is specified as part of the input.

**Proof.** Let $(G, k)$ be an instance of parameterized Maximum Matching. Our kernelization algorithm either returns yes, or it computes an equivalent reduced instance $(G', k')$.

First, we exhaustively apply Reduction Rule 3 by visiting every vertex once and removing every vertex of degree greater than $2(k-1)$ in the current graph. Notably, since vertex removals can only reduce the degree of the remaining vertices, the algorithm does not need to visit any vertex twice. If we construct an instance $(G', 0)$ during this procedure, that is, if we remove $k$ vertices from $G$, then we stop and return yes. The correctness of this decision follows immediately by the facts that $(G', 0)$ is clearly a yes-instance and Reduction Rule 3 is safe by Lemma 30.

The exhaustive application of Reduction Rule 3 can be implemented to run in $O(nk)$ time, as follows. Every time we discover a new vertex $v$ with $\deg(v) > 2(k-1)$ in the current graph (and for the current value of the parameter $k$), then we do not actually remove $v$ from the current graph but we mark it as "removed" and we proceed to the next vertex. Furthermore we keep in a counter $r$ the number of vertices that have been marked so far as "removed". Note that, to check whether we need to apply Reduction Rule 3 on a vertex $v$, we only need to visit at most all *marked* neighbors of $v$ and at most $2(k-r-1)+1 < 2k$ *unmarked* neighbors in the initial graph $G$. Thus, since there exist at every point at most $r < k$ marked vertices, we only need to check less than $3k$ neighbors of $v$ in the initial graph $G$ to decide whether we mark $v$ as a new "removed" vertex. Thus, since there are $n$ vertices in total, the whole procedure runs in $O(kn)$ time. Denote by $r_0$ the total number of vertices that have been marked as "removed" at the end of this process.

Next, we exhaustively apply Reduction Rule 4 by removing every unmarked vertex $v$ that has only marked neighbors in $G$. Since such a vertex $v$ remained unmarked during the exhaustive application of Reduction Rule 3, $v$ has less than $k$ *marked* neighbors and less than $2k$ *unmarked* neighbors in $G$, that is, at most $3k$ neighbors in total. Thus we can check in $O(k)$ time whether a currently unmarked vertex has only marked neighbors; in this case we mark $v$ as "removed". This process can be clearly done in $O(nk)$ time. Let $G'$ be the induced subgraph of $G$ on the unmarked vertices and let $k' = k - r_0$. Note that every vertex of $G'$ has at least one and at most $2(k'-1)$ neighbors in $G'$.

Finally we count the number of vertices and edges of $G'$ in $O(kn)$ time. This can be done by visiting again all unmarked vertices $v$ and their unmarked neighbors in $G$. If $G'$ has strictly more than $(k'-1)(2k'-1)$ vertices or edges, then we stop and return yes. Otherwise the kernelization algorithm returns the kernel $(G', k')$, which has $O(k^2)$ vertices and $O(k^2)$ edges. Consequently, the kernelization algorithm runs in $O(kn)$ time in total. It remains to prove that, if at least one of $|V(G')|$ or $|E(G')|$ is greater than $(k'-1)(2k'-1)$, then $(G', k')$ is a yes-instance. Assume otherwise that $(G', k')$ is a no-instance, that is, $\mathbf{mm}(G') \leq k' - 1$. Then it follows by Lemma 31 that $|V(G')|, |E(G')| \leq (k'-1)(2k'-1)$, which is a contradiction. This completes the proof of the theorem. □

Applying the matching algorithm due to Micali and Vazirani [43] to the kernel we obtained by Theorem 11, we achieve the following result.

**Corollary 4.** Maximum Matching, *when parameterized by solution size k, can be solved in* $O(nk + k^3)$ *time.*

**Proof.** Let $(G, k)$ be an instance of parameterized Maximum Matching, where $k$ is the solution size. First we apply to $(G, k)$ the kernelization algorithm of Theorem 11, which returns either yes or an equivalent instance $(G'', k'')$ with $O(k^2)$ vertices and $O(k^2)$ edges. Then we compute a maximum matching $\mathbf{mm}(G'')$ of the graph $G''$ using any of the known algorithms, e.g., the algorithm of Micali and Vazirani [43]. It computes $\mathbf{mm}(G'')$ in $O\left(|E(G'')| \cdot \sqrt{|V(G'')|}\right) = O(k^3)$ time. Finally, if $\mathbf{mm}(G'') \geq k''$, then return yes, otherwise return no. In total, Maximum Matching can be thus solved in $O(nk + k^3)$ time. □

## 6. Outlook and discussion

Our work heads at stimulating a general research program which systematically exploits the concept of fixed-parameter tractability for polynomially solvable problems. For several fundamental and widely known problems, the time complexities of the currently fastest algorithms are upper-bounded by polynomials of large degrees. One of the most prominent examples is arguably the celebrated polynomial-time recognition algorithm for *perfect graphs*, whose time complexity still remains $O(n^9)$ [19]. Apart from trying to improve the *worst-case* time complexity for such problems, which may be a very difficult (if not impossible) task, the complementary approach that we propose here is to try to spot a *parameter* that causes these high-degree polynomial-time algorithms and to separate the dependency of the time complexity from this parameter such that the dependency on the input size becomes as close to linear as possible. We believe that the "FPT inside P" field is very rich and offers plenty of research possibilities.

We conclude with three related topics that may lead to further interactions. First, we remark that in classical parameterized complexity analysis there is a growing awareness concerning the polynomial-time factors that often have been neglected [9]. Notably, there are some prominent fixed-parameter tractability results giving *linear-time* factors in the input size (but quite large exponential factors in the parameter); these include Bodlaender's famous "linear-time" algorithm for computing treewidth [11] and the more recent "linear-time" algorithm for computing the crossing number of a graph [36]. Interestingly, these papers emphasize "linear time" in their titles, instead of "fixed-parameter tractability". In this spirit, our result for Longest Path in Interval graphs is a "linear-time" algorithm where the dependency on the parameter is not exponential [11,36] but *polynomial*. In this line of research, Fomin et al. [25] studied graph and matrix problems on instances with small treewidth. In particular the authors presented, among other results, an $O(k^3 n \log n)$ randomized algorithm for computing the cardinality of a maximum matching and an $O(k^4 n \log^2 n)$ randomized algorithm for actually constructing a maximum matching, where $k$ is an upper bound for the treewidth of the given graph [25].

Second, polynomial-time solvability and the corresponding *lower bounds* have been of long-standing interest, e.g., it is believed that the famous 3SUM problem is only solvable in quadratic time and this conjecture has been employed for proving relative lower bounds for other problems [27]. Very recently, there was a significant push in this research direction with many new relative lower bounds [1,2,13]. The "FPT inside P" approach might help in "breaking" these nonlinear relative lower bounds by introducing useful parameterizations and striving for PL-FPT results. In this direction an interesting negative result appeared very recently by Abboud et al. [3] who proved that, unless some plausible complexity assumptions fail, for any $\varepsilon > 0$ there does not exist any algorithm with running time $2^{o(k)} n^{2-\varepsilon}$ for $(\frac{3}{2} - \delta)$-approximating the diameter or the radius of a graph, where $k$ is an upper bound for the treewidth. In contrast, they proved that both the diameter and the radius can be computed in $2^{O(k \log k)} n^{1+o(1)}$ time [3].

Finally, coming back to a practical motivation for "FPT inside P", it has been very recently observed that identifying various parameterizations for the same problem may help in designing meta-algorithms that (dynamically) select the most appropriate solution strategy (also specified by respective parameters)—this approach is known as "programming by optimization" [34]. Note that so far this line of research is still in its infancy with only one known study [32] for NP-hard problems; following this approach might also be promising within our "FPT inside P" framework.

## References


[1] A. Abboud, F. Grandoni, V.V. Williams, Subcubic equivalences between graph centrality problems, APSP and diameter, in: Proceedings of the 26th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2015, pp. 1681–1697.

[2] A. Abboud, V.V. Williams, Popular conjectures imply strong lower bounds for dynamic problems, in: Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2014, pp. 434–443.

[3] A. Abboud, V.V. Williams, J. Wang, Approximation and fixed parameter subquadratic algorithms for radius and diameter, in: Proceedings of the 27th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2016.

[4] I. Abraham, A. Fiat, A.V. Goldberg, R.F.F. Werneck, Highway dimension, shortest paths, and provably efficient algorithms, in: Proceedings of the 21st Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2010, pp. 782–793.

[5] N. Alon, R. Yuster, U. Zwick, Color-coding, J. ACM 42 (4) (1995) 844–856.

[6] A. Amir, M. Lewenstein, E. Porat, Faster algorithms for string matching with $k$ mismatches, J. Algorithms 50 (2) (2004) 257–275.

[7] J. Bang-Jensen, G. Gutin, Paths and cycles in extended and decomposable digraphs, Discrete Math. 164 (1–3) (1997) 5–19.

[8] A.A. Bertossi, Finding Hamiltonian circuits in proper interval graphs, Inform. Process. Lett. 17 (2) (1983) 97–101.

[9] R. van Bevern, Fixed-Parameter Linear-Time Algorithms for NP-hard Graph and Hypergraph Problems Arising in Industrial Applications, PhD thesis, Institut für Softwaretechnik und Theoretische Informatik, TU, Berlin, Germany, 2014.

[10] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Narrow sieves for parameterized paths and packings, J. Comput. Syst. Sci. 87 (2017) 119–139.

[11] H.L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317.

[12] M. Bojańczyk, P. Parys, XPath evaluation in linear time, J. ACM 58 (4) (2011) 17:1–17:33.

[13] K. Bringmann, Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails, in: Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2014, pp. 661–670.

[14] S. Cabello, C. Knauer, Algorithms for graphs of bounded treewidth via orthogonal range searching, Comput. Geom. 42 (9) (2009) 815–824.

[15] L. Cai, Fixed-parameter tractability of graph modification problems for hereditary properties, Inform. Process. Lett. 58 (4) (1996) 171–176.

[16] L. Cai, Parameterized complexity of vertex colouring, Discrete Appl. Math. 127 (3) (2003) 415–429.

[17] E.W. Chambers, J. Erickson, A. Nayyeri, Homology flows, cohomology cuts, SIAM J. Comput. 41 (6) (2012) 1605–1634.

[18] J. Chen, S. Lu, S. Sze, F. Zhang, Improved algorithms for path, matching, and packing problems, in: Proceedings of the 18th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 298–307.

[19] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, K. Vušković, Recognizing Berge graphs, Combinatorica 25 (2) (2005) 143–186.

[20] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.

[21] P. Damaschke, Paths in interval graphs and circular arc graphs, Discrete Math. 112 (1–3) (1993) 49–64.

[22] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts Comput. Sci., Springer, 2013.

[23] M.R. Fellows, B.M.P. Jansen, F.A. Rosamond, Towards fully multivariate algorithmics: parameter ecology and the deconstruction of computational complexity, European J. Combin. 34 (3) (2013) 541–566.

[24] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer, 2006.

[25] F.V. Fomin, D. Lokshtanov, M. Pilipczuk, S. Saurabh, M. Wrochna, Fully polynomial-time parameterized computations for graphs and matrices of low treewidth, in: Proceedings of the 28th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2017, pp. 1419–1432.

[26] F.V. Fomin, D. Lokshtanov, S. Saurabh, Efficient computation of representative sets with applications in parameterized and exact algorithms, in: Proceedings of the 25th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2014, pp. 142–151.

[27] A. Gajentaan, M.H. Overmars, On a class of $O(n^2)$ problems in computational geometry, Comput. Geom. 5 (1995) 165–185.

[28] A.C. Giannopoulou, G.B. Mertzios, R. Niedermeier, Polynomial fixed-parameter algorithms: a case study for longest path on interval graphs, in: Proceedings of the 10th International Symposium on Parameterized and Exact Computation (IPEC), 2015, pp. 102–113.

[29] J. Guo, F. Hüffner, R. Niedermeier, A structural view on parameterizing problems: distance from triviality, in: Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC), 2004, pp. 162–173.

[30] G. Gutin, Finding a longest path in a complete multipartite digraph, SIAM J. Discrete Math. 6 (2) (1993) 270–273.

[31] T. Hagerup, J. Katajainen, N. Nishimura, P. Ragde, Characterizing multiterminal flow networks and computing flows in networks of small treewidth, J. Comput. System Sci. 57 (3) (1998) 366–375.

[32] S. Hartung, H.H. Hoos, Programming by optimisation meets parameterised algorithmics: a case study for cluster editing, in: Proceedings of the 9th Learning and Intelligent OptimizatioN Conference (LION), 2015, pp. 43–58.

[33] J.M. Hochstein, K. Weihe, Maximum s-t-flow with $k$ crossings in $O(k^3 n \log n)$ time, in: Proceedings of the 18th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 843–847.

[34] H.H. Hoos, Programming by optimization, Commun. ACM 55 (2) (2012) 70–80.

[35] K. Ioannidou, G.B. Mertzios, S.D. Nikolopoulos, The longest path problem has a polynomial solution on interval graphs, Algorithmica 61 (2) (2011) 320–341.

[36] K. Kawarabayashi, B.A. Reed, Computing crossing number in linear time, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC), 2007, pp. 382–390.
[37] J.M. Keil, Finding Hamiltonian circuits in interval graphs, Inform. Process. Lett. 20 (1985) 201–206.
[38] J. Kneis, D. Mölle, S. Richter, P. Rossmanith, Divide-and-color, in: Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG), 2006, pp. 58–67.
[39] I. Koutis, G.L. Miller, R. Peng, A nearly-$m \log n$ time solver for SDD linear systems, in: Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), 2011, pp. 590–598.
[40] N. Megiddo, Linear programming in linear time when the dimension is fixed, J. ACM 31 (1) (1984) 114–127.
[41] G.B. Mertzios, D.G. Corneil, A simple polynomial algorithm for the longest path problem on cocomparability graphs, SIAM J. Discrete Math. 26 (3) (2012) 940–963.
[42] G.B. Mertzios, A. Nichterlein, R. Niedermeier, The power of data reduction for matching, Technical Report (2016), available at https://arxiv.org/abs/1609.08879.
[43] S. Micali, V.V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, in: Proceedings of the 21st Annual Symposium on Foundations of Computer Science (FOCS), 1980, pp. 17–27.
[44] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford Lecture Ser. Math. Appl., vol. 31, Oxford University Press, 2006.
[45] R. Niedermeier, Reflections on multivariate algorithmics and problem parameterization, in: Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS), 2010, pp. 17–32.
[46] J.B. Orlin, K. Madduri, K. Subramani, M.D. Williamson, A faster algorithm for the single source shortest path problem with few distinct positive lengths, J. Discrete Algorithms 8 (2) (2010) 189–198.
[47] G. Ramalingam, C.P. Rangan, A unified approach to domination problems on interval graphs, Inform. Process. Lett. 27 (5) (1988) 271–274.
[48] F.S. Roberts, Indifference graphs, in: F. Harary (Ed.), Proof Techniques in Graph Theory, Academic Press, New York, 1969, pp. 139–146.
[49] R. Uehara, Y. Uno, On computing longest paths in small graph classes, Internat. J. Found. Comput. Sci. 18 (5) (2007) 911–930.
[50] R. Williams, Finding paths of length $k$ in $O^*(2^k)$ time, Inform. Process. Lett. 109 (6) (2009) 315–318.