Electronic Preprint for Journal of Information Processing Vol.28

Technical Note

An Algorithm for the Influential Hinge Vertex Problem on Interval Graphs

Hirotoshi Honma[1,a)] Yoko Nakajima[1,b)] Shigeru Masuyama[2]

Received: June 23, 2020, Accepted: August 7, 2020

Abstract: Consider a simple undirected graph $G = (V, E)$ with vertex set V and edge set E. The distance $\delta_G(x, y)$ is defined as the length of the shortest path between vertices x and y in G. The vertex $u \in V$ is a *hinge vertex* if there exist two vertices $x, y \in V - \{u\}$ such that $\delta_{G-u}(x, y) > \delta_G(x, y)$. Let U be a set consisting of all hinge vertices of G, and let $AVS(u)$ denote the set of pairs of vertices (x, y)s to which a path between x and y becomes longer after removal of a hinge vertex u from G. The *influential hinge vertex problem* aims to determine the hinge vertex u that maximizes $|AVS(u)|$ in G. In this study, we propose an algorithm that runs in $O(n^2)$ time to solve the influential hinge vertex problem on an interval graph.

Keywords: algorithm design, influential hinge vertex problem, interval graphs

1. Introduction

Consider a simple undirected graph $G = (V, E)$ with vertex set V and edge set E. Let $G - u$ be a subgraph induced by the vertex set $V - \{u\}$. The distance $\delta_G(x, y)$ is defined as the length (i.e., the number of edges) of the shortest path between vertices x and y in G. Chang et al. [4] defined $u \in V$ to be a *hinge vertex* if there exist two vertices $x, y \in V - \{u\}$ such that $\delta_{G-u}(x, y) > \delta_G(x, y)$. In other words, a hinge vertex is a vertex in an undirected graph such that there exist two vertices whose removal makes the distance between them longer than before. Identifying all hinge vertices of a given graph is called the *hinge vertex problem*. It is worth noting that an *articulation vertex* is a special case of a hinge vertex in the sense that its removal changes the finite distance of some nonadjacent vertices x, y to infinity.

The following Lemma 1, proved by Chang et al. [4], characterizes the hinge vertices of a simple graph G. The hinge vertex problem can be solved in $O(n^3)$ time according to the Lemma.

Lemma 1 (Ref. [4]) For a simple graph G, a vertex u is a hinge vertex of G if and only if there exist two nonadjacent vertices x, y such that u is the only vertex adjacent to both x and y in G.

A number of studies concerning hinge vertices have been reported in recent years. For instance, Ho et al. [7] presented an $O(n)$ time algorithm for the hinge vertex problem on permutation graphs. Some minor errors in their approach were corrected by Ref. [8]. Furthermore, Honma and Masuyama [9] presented an $O(n \log n)$ time algorithm for the hinge vertex problem on interval graphs. Their algorithm was later improved to $O(n)$ time by Hsu et al. [11]. The class of *circular-arc graphs* is a superclass of interval graphs, for which Honma and Masuyama [10] developed an $O(n)$ time algorithm for the hinge vertex problem. The class of *trapezoid graphs* properly contains both interval graphs and permutation graphs, and an $O(n \log n)$ time algorithm for the hinge vertex problem on such graphs was developed by Bera et al. [1].

In this study, we introduce a new concept regarding hinge vertices. Let U be a hinge vertex set of G. We define $AVS(u) = \{(x, y) \mid \delta_{G-u}(x, y) > \delta_G(x, y)\}$ for $u \in U$ as the *affected pair vertex sets* of u; that is, $AVS(u)$ denotes the set of pairs of vertices (x, y)s for which a path between x and y becomes longer after removal of a hinge vertex u from G. Furthermore, we define the *influential degree* $inf(u) = |AVS(u)|$. The *influential hinge vertex problem* aims to determine the hinge vertex u that maximizes $inf(u)$ in G.

The hinge vertex problem can be applied to improving the stability and robustness in communication network systems [9]. If some terminal corresponding to a hinge vertex fails, the efficiency of communication across the network will decrease because of the increase in the number of hops between a pair of terminals. Finding the set of hinge vertices in a graph is useful for identifying critical nodes in an actual network. In particular, it is important to find the hinge vertex that renders a serious effect especially when it breaks down. Therefore, the influential hinge vertex problem is motivated by practical applications such as network stabilization under a limited cost [9].

In this study, we propose an algorithm to solve the influential hinge vertex problem on *interval graphs*. The class of interval graphs is an important subclass of perfect graphs and it frequently appears in problem settings in fields such as archaeology [15], molecular biology [14], bioinformatics [17], genetics [13], VLSI design [6], circuit routing [16], and scheduling [3]. An extensive discussion of interval graphs also appears in Ref. [5]. Thus, interval graphs have been studied extensively from both theoretical

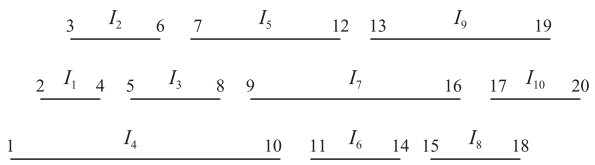
1 Department of Creative Engineering, National Institute of Technology, Kushiro College, Kushiro, Hokkaido 084–0916, Japan
2 Department of School of Management, Tokyo University of Science, Chiyoda, Tokyo 102–0071, Japan
a) honma@kushiro-ct.ac.jp
b) yoko@kushiro-ct.ac.jp

and algorithmic viewpoints.

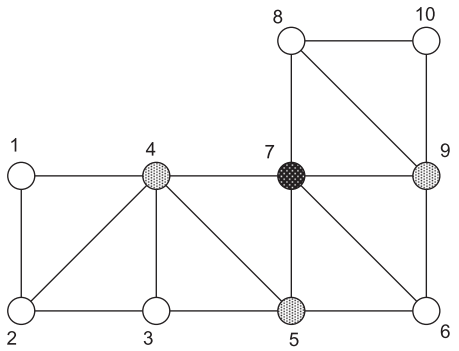
2. Definitions and Notation

We first illustrate the *interval model* before defining the interval graph. The interval model is a set of n horizontal line segments, called *intervals*. Each interval $I_i = \{k \mid a_i \leq k \leq b_i\}$, where $a_i < b_i$, has two terminal points, a_i and b_i. Without loss of generality, we assume that all terminal points are distinct. Let n be the number of intervals, where the terminal points of each interval are labeled from left to right with consecutive integer values $1, 2, \ldots, 2n$. We assume that an interval number is assigned to each interval in increasing order of their right terminal points b_i, i.e., $I_i < I_j$ if $b_i < b_j$. The geometric representation described above is called an interval model and is denoted by M. **Figure 1** (a) shows an interval model M consisting of 10 intervals.

A graph $G = (V, E)$ is an interval graph if its vertices can be assigned in a one-to-one correspondence with the intervals in M such that two vertices are adjacent in G if and only if their corresponding intervals have a non-empty intersection in M. The interval graph G corresponding to the interval model M is shown in Fig. 1 (b). Booth and Lueker [2] have given a linear time algorithm for recognizing interval graphs. Their algorithm relies on maximal cliques and also produces an interval model if the graph is indeed an interval graph. Later, Hsu [12] also gave an algorithm for recognizing interval graphs without using maximal cliques.



(a) Interval model M



(b) Interval graph G

Fig. 1 Interval model M and graph G.

In the example of Fig. 1, vertices 4, 5, 7, and 9 are hinge vertices in G. **Table 1** shows $AVS(u)$ and $inf(u)$ in the example G of Fig. 1 (b). The most influential hinge vertex is 7.

Notation that forms the basis of the algorithm described in Section 5 is defined as follows. Let M be an interval model consisting of n intervals $I_1, I_2, \ldots, I_n$. Then, $mr(i)$ is the largest j, satisfying that I_i intersects with I_j and $b_i < b_j$. Here, $smr(i)$ is the second-largest j, satisfying that I_i intersects with I_j and $b_i < b_j$. Also, $ml(i)$ is k such that a_k is the smallest value among I_k that intersects I_i and $a_k < a_i$. Note that $mr(i) = i$, $smr(i) = i$, and $ml(i) = i$ when such an interval does not exist. These are formally described as follows. Here, the set of all intervals that intersect I_i in M is denoted by $N(i)$. In addition, $N[i] = N(i) \cup \{i\}$.

- $mr(i) = \max\{j \mid j \in N[i]\}$,
- $smr(i) = \max\{j \mid j \in (N[i] - mr(i)) \cup \{i\}\}$,
- $ml(i) = k$ where $a_k = \min\{a_j \mid j \in N[i]\}$.

We define $D(i) = \{k \mid b_{smr(i)} < k < b_{mr(i)}\}$. **Table 2** provides the details of $mr(i)$, $smr(i)$, $ml(i)$, and $D(i)$ for the interval model M shown in Fig. 1.

Suppose U is the hinge vertex set of an interval graph G. We can obtain U in $O(n)$ time in G by applying Hsu et al's algorithm [11]. In the example of Fig. 1, we have $U = \{4, 5, 7, 9\}$. Next, we define $V_u = \{i \mid mr(i) = u\}$ for $u \in U$. For example, $V_4 = \{1, 2\}$, $V_5 = \{3\}$, $V_7 = \{4, 5\}$, and $V_9 = \{6, 7\}$ for $U = \{4, 5, 7, 9\}$. Furthermore, the *representative vertex set* (RVS) is defined as $RVS = \{\min(V_u) \mid u \in U\}$. We have $RVS = \{1, 3, 4, 6\}$.

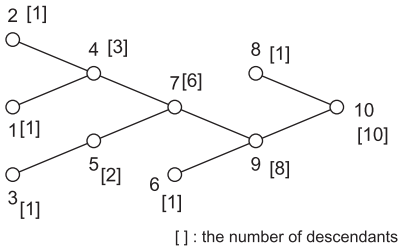
In the following, we define an mr-tree and ml-tree for an interval graph G. An mr-tree is a tree constructed by assigning an edge from i to $mr(i)$ for each vertex i ($i \neq mr(i)$). Furthermore, $dr(i)$ is the number (including i) of descendants of i in the mr-tree. Similarly, an ml-tree is a tree constructed by assigning an edge from i to $ml(i)$ for each vertex i ($i \neq ml(i)$). Furthermore, $dl(i)$ is the number (including i) of descendants of i in the ml-tree. By the property of interval graphs, both an mr-tree and ml-tree construct shortest path trees from all other vertices to a root vertex. **Figure 2** shows an mr-tree and ml-tree of the interval graph G illustrated in Fig. 1. The details of $dr(i)$ and $dl(i)$ are shown in Table 2.

Table 1 $AVS(u)$ and $inf(u)$.

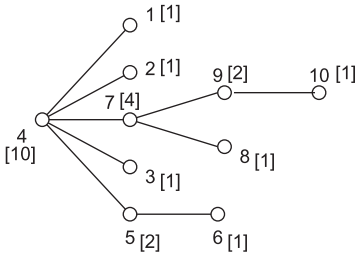
$u \in U$	$AVS(u)$	$inf(u)$
4	(1,5), (1,6), (1,7), (1,8), (1,9), (1,10), (2,7), (2,8), (2,9), (2,10)	10
5	(3,6)	1
7	(1,8), (1,9), (1,10), (2,8), (2,9), (2,10), (3,8), (4,8), (4,9), (4,10), (5,8)	11
9	(6,10)	1

Table 2 Details of $mr(i)$, $smr(i)$, $D(i)$, $ml(i)$, $dr(i)$, and $dl(i)$.

i	1	2	3	4	5	6	7	8	9	10
a	2	3	5	1	7	11	9	15	13	17
b	4	6	8	10	12	14	16	18	19	20
$mr(i)$	4	4	5	7	7	9	9	10	10	10
$smr(i)$	2	3	4	5	6	7	8	9	9	10
$ml(i)$	4	4	4	4	4	5	4	7	7	9
$D(i)$	{7,8,9}	{9}	{11}	{13,14,15}	{15}	{17,18}	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$dr(i)$	1	1	1	3	2	1	6	1	8	10
$dl(i)$	1	1	1	10	2	1	4	1	2	1



(a) An *mr*-tree



(b) An *ml*-tree

Fig. 2 An *mr*-tree and *ml*-tree.

3. Properties of a Hinge Vertex for Interval Graphs

In this section, we demonstrate some properties useful for computing $inf(u)$ of a hinge vertex u on a given interval graph. For simplicity, we assume u is the hinge vertex for x and y when $\delta_{G-u}(x, y) > \delta_G(x, y)$ holds. Furthermore, we assume x and y are *affected pair vertices* for u.

Lemmas 2 to 4 provide useful results on hinge vertices in interval graphs presented by Honma and Masuyama [9].

Lemma 2 (Ref. [9]) Let G be an interval graph corresponding to an interval model M. A vertex u is a hinge vertex for $x < y \in N(u)$ in G if and only if there exist two nonintersecting intervals I_x and I_y such that I_u is the only interval intersecting both I_x and I_y in M.

Lemma 3 (Ref. [9]) Let G be an interval graph corresponding to an interval model M. A vertex u is a hinge vertex for $x < y \in N(u)$ in G if and only if $u = mr(x)$ and $a_y \in D(x)$ hold.

Figure 3 gives an example of the hypothesis of Lemma 3. The following Lemma 4 is required in order to reduce the time complexity of the hinge vertex problem on interval graphs.

Lemma 4 (Ref. [9]) Let G be an interval graph corresponding to an interval model M. Let RVS be a representative vertex set of G. A vertex $u = mr(x)$ is a hinge vertex for $x < y \in N(u)$ if and only if there exists such $a_y \in D(x)$ for $x \in RVS$. □

In the example of Fig. 1, for $x = 4$ ($x \in RVS = \{1, 3, 4, 6\}$) and $y = 9$, we have $D(x) = D(4) = \{13, 14, 15\}$ and $a_y = a_9 = 13 \in D(x)$. Therefore, $u = mr(x) = mr(4) = 7$ is recognized as a hinge vertex by Lemma 4.

All hinge vertices of an interval graph can be obtained in $O(n)$ time by applying the algorithm of Hsu et al. [11] which is also based on Lemma 4.

Lemma 5 Let G be an interval graph corresponding to an interval model M. Assume that u is a hinge vertex for two vertices x and y ($x < y$) in G. Then, for x' satisfying $mr(x') = x$, u is also a hinge vertex for x' and y. □

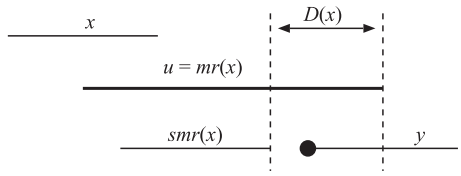
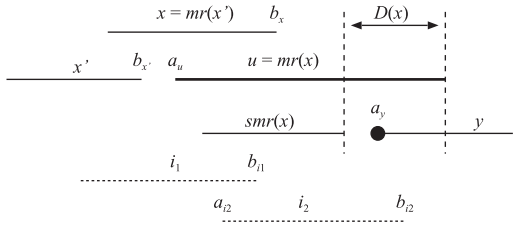
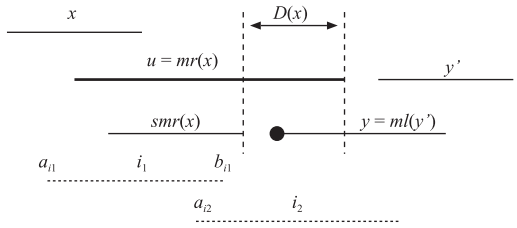


Fig. 3 Illustration of Lemma 3.



(a) Lemma 5



(b) Lemma 6

Fig. 4 Illustration of Lemmas 5 and 6.

Proof: For simplicity, consider the case where u is a hinge vertex for $x, y \in N(u)$. By Lemma 1, x is not adjacent to y, and u is the only vertex adjacent to both x and y in G. Thus, we have $\delta_G(x, y) = 2$ by $x, y \in N(u)$. Moreover, for x' satisfying that $mr(x') = x$, x' is adjacent to x, but not to u. If $b_{x'} > a_u$, u intersects with x' and $mr(x')$ cannot be x (see **Fig. 4** (a)). That is, $\delta_G(x', y) = 3$.

We prove that u is a hinge vertex of x' and y by contradiction. We assume that u is not a hinge vertex for x' and y. The distance between x' and y does not increase by removing u because u is not a hinge vertex for x' and y, i.e., we have $\delta_{G-u}(x', y) = 3$. This means that there exists a shortest path of length 3 $\langle x', i_1, i_2, y\rangle$ from x' to y (see Fig. 4 (a)).

In this case, i_1 intersects both x' and i_2, and i_2 intersects both i_1 and y. Then, we have $a_{i_2} < b_{i_1}$ and $a_y < b_{i_2}$ because i_2 intersects both i_1 and y. Moreover, $b_{i_1} \leqslant b_x$ holds by the assumption $x = mr(x')$. Therefore, we have $a_{i_2} < b_x$ and $a_y < b_{i_2}$. This means that i_2 intersects both x and y (see Fig. 4 (a)). This, however, contradicts the fact that u is the only vertex adjacent to both x and y in G. Thus, u is a hinge vertex of x' and y. □.

Lemma 6 Let G be an interval graph corresponding to an interval model M. Assume that u is a hinge vertex for two vertices x and y ($x < y$) in G. Then, for y' satisfying $ml(y') = y$, u is also a hinge vertex for x and y'.

Proof: The hypothesis of Lemma 6 is symmetric with respect to Lemma 5. Thus, this lemma can be proved using a reasoning similar to the proof of Lemma 5. □

Lemma 7 Let G be an interval graph corresponding to an in-

Table 3 Details of $r(k)$ and $l(k)$.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r	0	0	0	0	0	0	1	1	2	0	1	0	3	3	5	0	1	1	0	0
l	1	1	1	0	10	0	2	0	4	0	1	0	2	0	1	0	1	0	0	0
$r \times l$	0	0	0	0	0	0	2	0	8	0	1	0	6	0	5	0	1	0	0	0

terval model M. Assume that u is a hinge vertex for two vertices x and y ($x < y$) in G. Then u is also a hinge vertex for two vertices x', y' satisfying $x = mr(x')$ and $y = ml(y')$.

Proof: This is obvious from Lemmas 5 and 6. □

Lemma 8 Let G be an interval graph corresponding to an interval model M. Assume that u is a hinge vertex for a pair of vertices x and y ($x < y \in N(u)$) in G. Suppose both an mr-tree and ml-tree are trees constructed from M, and $dr(i)$ and $dl(i)$ are the number (including i) of descendants of i in the mr-tree and ml-tree, respectively. Then the number of affected pair vertices by removing hinge vertex u is $dr(x) \times dl(y)$.

Proof: By Lemma 7, if u is a hinge vertex for two vertices x and y ($x < y$) in G, then u is also a hinge vertex for two vertices x', y' satisfying $x = mr(x')$ and $y = ml(y')$.

From the mr-tree construction method, there are p (including x) vertices connected to x through the mr value of each vertex if $dr(x) = p$. Similarly, from the ml-tree construction method, there are q (including y) vertices connected to y through the ml value of each vertex if $dl(y) = q$.

Therefore, the number of affected pair vertices by removing hinge vertex u is $dr(x) \times dl(y)$. □

Lemma 9 Let G be an interval graph corresponding to an interval model M. Assume that u is a hinge vertex and there are vertices $x_1 < x_2 < \cdots < x_p$ that satisfy $u = mr(x_1) = mr(x_2) = \cdots = mr(x_p)$ and $y_1, y_2, \ldots, y_q$ that satisfy $a_{y_1}, a_{y_2}, \ldots, a_{y_q} \in D(x_1)$. Then the influential degree $inf(u)$ of a hinge vertex u in G is obtained as follows:

$$inf(u) = \sum_{i=1}^{p} \sum_{j=1}^{q} dr(x_i) \times dl(y_j).$$

Lemma 9 is a generalization of Lemma 8. We can efficiently obtain $inf(u)$ for all hinge vertices $u \in U$ by applying Lemma 9.

4. Algorithm IHV and Its Analysis

In this section, we present an algorithm IHV for the influential hinge vertex problem on an interval graph. The algorithm IHV is based on Lemma 9. Now, we concisely describe the outline of our algorithm and analyze its complexity. We use the interval graph G shown in Fig. 1 to illustrate this algorithm.

In Step 1, we obtain a hinge vertex set $U = \{4, 5, 7, 9\}$ by applying Hsu et al.'s algorithm [11]. In this process, $mr(i)$, $smr(i)$, $ml(i)$, and $D(i)$ are computed for $1 \leqslant i \leqslant n$.

In Step 2, the mr-tree and ml-tree are constructed from an interval model M. Moreover, for each vertex i of the mr-tree and ml-tree, the number of descendants $dr(i)$ and $dl(i)$ are computed. In the example of Fig. 2, we have $dr = [1, 1, 1, 3, 2, 1, 6, 1, 8, 10]$ and $dl = [1, 1, 1, 10, 2, 1, 4, 1, 2, 1]$.

In Step 3, we construct auxiliary arrays $r(j)$ and $l(j)$, $1 \leqslant j \leqslant 2n$ using $dr(i)$ and $dl(i)$. These arrays $r(j)$ and $l(j)$ are used to efficiently obtain the influential degree $inf(u)$, $u \in U$. The examples

Algorithm 1: Algorithm IHV

Input: All terminal points a_i, b_i for n intervals in the interval model M of interval graph G.
Output: The maximum influential hinge vertex of an interval graph G.

(Step 1) /* Computation of hinge vertex set U */
Compute a hinge vertex set U of G using Hsu et al.'s algorithm [11] ;
In the above process, $mr(i)$, $smr(i)$, $ml(i)$, and $D(i)$ are obtained for $1 \leqslant i \leqslant n$.

(Step 2) /* Construction of an mr-tree and ml-tree and compute the number of descendants $dr(i)$ and $dl(i)$ */
Construct an mr-tree and ml-tree;
for *each* $1 \leqslant i \leqslant n$ **do**
  Compute $dr(i)$ and $dl(i)$;
end

(Step 3) /* Construction of arrays $r(j)$ and $l(j)$ */
/* Initialization */
for *each* $1 \leqslant j \leqslant 2n$ **do**
  $r(j) := 0$, $l(j) := 0$;
end
/* Construction of $r(j)$ */
for *each* $1 \leqslant i \leqslant n$ **do**
  for *each* $j \in D(i)(\neq \emptyset)$ **do**
    $r(j) := r(j) + dr(i)$;
  end
end
/* Construction of $l(j)$ */
for *each* $1 \leqslant i \leqslant n$ **do**
  $l(a_i) := dl(i)$;
end

(Step 4)/* Computation of influential degree
for *each* $i \in RVS$ **do**
  for *each* $j \in D(i)$ **do**
    $inf(mr(i)) = inf(mr(i)) + (r(j) \times l(j))$
  end
end

(Step 5)
The maximum influential hinge vertex is u such that
$inf(u) = \max\{inf(i) \mid i \in U\}$

of $r(j)$ and $l(j)$ obtained from an interval model M in Fig. 1 are shown in **Table 3**.

In Step 4, based on Lemma 9, we compute the influential degree $inf(u)$ for each hinge vertex $u \in U$. For example, for $i = 4 \in RVS = \{1, 3, 4, 6\}$, we have $D(4) = \{13, 14, 15\}$. $inf(mr(4)) = inf(7) = \sum_{j \in D(4)} r(j) \times l(j) = 3 \times 2 + 3 \times 0 + 5 \times 1 = 11$. In a similar manner, for $i = 1, 3,$ and $6 \in RVS$, we have $D(1) = \{7, 8, 9\}$, $D(3) = \{11\}$, and $D(6) = \{17, 18\}$. Thus, we obtain $inf(mr(1)) = inf(4) = 1 \times 2 + 1 \times 0 + 2 \times 4 = 10$, $inf(mr(3)) = inf(5) = 1 \times 1 = 5$, and $inf(mr(6)) = inf(9) = 1 \times 1 + 1 \times 0 = 1$, respectively.

In Step 5, we obtain the maximum influential hinge vertex. In the above example, the maximum influential hinge vertex is 7 and $inf(7) = 11$.

In the following, we analyze the complexity of Algorithm IHV. In Step 1, we can obtain all hinge vertices of an interval graph G in $O(n)$ time using Hsu et al.'s algorithm [11]. However, we require $O(n^2)$ time to compute $D(i)$ because the size of $\sum_{i=1}^{n} |D(i)|$ is proportional to n^2. Thus, Step 1 can be performed in $O(n^2)$ time.

In Step 2, we construct an mr-tree and ml-tree, and compute $dr(i)$ and $dl(i)$. Step 2 can be executed in $O(n)$ time. In Step 3, we construct auxiliary arrays $r(j)$ and $l(j)$. Step 3 runs in $O(n)$ time. In Step 4, we obtain the influential degree for each hinge vertex. Step 4 can be executed in $O(n^2)$ time because the size of $\sum_{i=1}^{n} |D(i)|$ is proportional to n^2. Hence, we have the following theorem.

Theorem 1 Given an interval model M, algorithm IHV finds a maximum influential hinge vertex of an interval graph G in $O(n^2)$ time.

5. Conclusion

In this study, we propose an $O(n^2)$ time algorithm to solve the influential hinge vertex problem of an interval graph. Algorithm IHV uses Hsu et al.'s algorithm [11] to find all hinge vertices. Reducing the complexity of the algorithm and extending the results to other graphs will be considered as topics for future research.

Acknowledgments We express many thanks to anonymous referees for their valuable advices on the theory of our attacks and their helpful editorial comments. This work was partially supported by JSPS KAKENHI Grant Number 19K11834 and 20K11093.

References

[1] Bera, D., Pal, M. and Pal, T.K.: An efficient algorithm for finding all hinge vertices on trapezoid graphs, *Theory of Computing Systems*, Vol.36, No.1, pp.17–27 (2003).

[2] Booth, K.S. and Lueker, G.S.: Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, *J. Comput. Syst. Sci.*, Vol.13, pp.335–379 (1976).

[3] Carlisle, M.C. and Loyd, E.L.: On the k-coloring of intervals, *International Conference on Computing and Information*, LNCS, Vol.497, pp.90–101, Springer (1991).

[4] Chang, J.M., Hsu, C.C., Wang, Y.L. and Ho, T.Y.: Finding the set of all hinge vertices for strongly chordal graphs in linear time, *Inf. Sci.*, Vol.99, No.3-4, pp.173–182 (1997).

[5] Golumbic, M.C.: *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York (1980).

[6] Ho, K.C.: Interval graph algorithms for two-dimensional multiple folding of array-based VLSI layouts, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.13, No.10, pp.1201–1222 (1994).

[7] Ho, T.Y., Wang, Y.L. and Juan, M.T.: A linear time algorithm for finding all hinge vertices of a permutation graph, *Inf. Process. Lett.*, Vol.59, No.2, pp.103–107 (1996).

[8] Honma, H., Abe, K. and Masuyama, S.: Erratum and addendum to "A linear time algorithm for finding all hinge vertices of a permutation graph" [Information Processing Letters, Vol.59, No.2, pp.103–107 (1996)], *Inf. Process. Lett.*, Vol.111, No.18, pp.891–894 (2011).

[9] Honma, H. and Masuyama, S.: A parallel algorithm for finding all hinge vertices of an interval graph, *IEICE Trans. Inf. & Syst.*, Vol.E84-D, No.3, pp.419–423 (2001).

[10] Honma, H. and Masuyama, S.: An optimal parallel algorithm for finding all hinge vertices of a circular-arc graph, *IEICE Trans. Fundamentals*, Vol. E91-A, No.1, pp.383–391 (2008).

[11] Hsu, F.R., Shan, K., Chao, H.S. and Lee, R.C.: Some optimal parallel algorithms on interval and circular-arc graphs, *J. Inf. Sci. Eng.*, Vol.21, pp.627–642 (2005).

[12] Hsu, W.L.: A simple test for interval graphs, *Proc. 18th Workshop on Graph Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci.*, Vol.657, pp.10–16, Springer (1993).

[13] Jungck, J.R., Dick, O. and Dick, A.G.: Computer assisted sequencing, interval graphs andmolecular evolution, *Biosystem*, Vol.15, pp.259–273 (1982).

[14] Kaplan, H. and Shamir, R.: Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques, *SIAM J. Comput.*, Vol.25, No.3, pp.540–561 (1996).

[15] Kendall, D.G.: Incidence matrices, interval graphs and seriation in archeology, *Pacific J. Math.*, Vol.28, No.3, pp.565–570 (1969).

[16] Ohtsuki, T., Mori, H., Khu, E.S., Kashiwabara, T. and Fujisawa, T.: One dimensional logicgate assignment and interval graph, *IEEE Trans. Circuits and Systems*, Vol.26, pp.675–684 (1979).

[17] Zhang, P., Schon, E.A., Fischer, S.G., Cayanis, E., Weiss, J., Kistler, S. and Bourne, P.E.: An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA, *Bioinformatics*, Vol.10, No.3, pp.309–317 (1994).

Hirotoshi Honma is a professor at the Department of Creative Engineering, National Institute of Technology, Kushiro College. He received his B.E., M.E., and D.E. degrees in Engineering from Toyohashi University of Technology, in 1990, 1992 and 2009, respectively. His research interests include computational graph theory and parallel algorithms.

Yoko Nakajima is a an associate professor at the Department of Creative Engineering, National Institute of Technology, Kushiro College. She received D.E. degree from Kitami Institute of Technology in 2016. She is working on extraction of future trend information with future reference sentences and its application to future event prediction.

Shigeru Masuyama received his B.S., M.S. and Dr. degrees in Engineering from Kyoto University, Kyoto, in 1977, 1979 and 1983, respectively. Since April, 2019, he has been serving as a professor at the Department of Management, School of Management, Tokyo University of Science. His research interests include graph algorithms and natural language processing.