Journal of Engineering and Digital Technology (JEDT)
Vol.11 No.1 January - June 2023

Algorithm for Solving the Detour Hinge Vertex Problem on Circular-arc Graphs

Yoko Nakajima[1] Shoma Nameki[2] Tomonari Izumi[3] Hirotoshi Honma[4*]

[1,2,3,4*]*Department of Creative Engineering, National Institute of Technology, Kushiro College, Japan*
[2]*Department of Electronics and Information Engineering, Hokkaido University, Japan*

*Corresponding Author. E-mail address: honma@kushiro-ct.ac.jp

Received: 2 February 2023; Revised: 23 March 2023; Accepted: 28 April 2023
Published online: 29 June 2023

Abstract

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 $d_G(x, y)$ 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 are two vertices $x, y \in V - u$ such that $d_{G-u}(x, y) > d_G(x, y)$. Let U be a set consisting of all hinge vertices of G. The neighborhood of u, denoted by $N(u)$, is the set of all vertices adjacent to u. We define the detour degree of u as $det(u) = \max\{ d_{G-u}(x, y) - 2 \mid d_{G-u}(x, y) > d_G(x, y), x, y \in N(u)\}$ for $u \in U$. The detour hinge vertex problem aims to determine the hinge vertex u that maximizes $det(u)$ in G. In this study, we proposed an efficient algorithm for solving the detour hinge vertex problem on circular-arc graphs that runs in $O(n^2)$ time, where n is the number of vertices in the graph.

Keywords: Circular-arc graphs, Design and analysis of algorithms, Detour hinge vertex problem, Intersection graphs

I. INTRODUCTION

Consider a simple undirected graph $G = (V, E)$ with vertex set V and edge set E. In this study, n is the number of vertices in the graph. Let $G - u$ be a subgraph induced by the vertex set $V - u$. The distance $d_G(x, y)$ is defined as the length (that is, the number of edges) of the shortest path between vertices x and y in G. Chang et al. [1] defined $u \in V$ as a *hinge vertex* if there are two vertices $x, y \in V - u$ such that $d_{G-u}(x, y) > d_G(x, y)$.

Finding all hinge vertices of a given graph is called *the hinge vertex problem*. This problem has applications regarding the improvement of the stability and robustness of communication network systems [2]. If a terminal in a network corresponding to a hinge vertex stalls, the number of hops between a pair of terminals will increase, resulting in the decreasing efficiency of the communication across the network. The set of hinge vertices in a graph can be used to identify critical nodes, which can be useful in constructing communication network systems with high stability.

Let U be a set consisting of all hinge vertices of G. The *neighborhood* of u, denoted by $N(u)$, is the set of all vertices adjacent to u. Honma et al. [3] defined the *detour degree* of u as $det(u) = \max\{ d_{G-u}(x, y) - 2 \mid d_{G-u}(x, y) > d_G(x, y), x, y \in N(u)\}$ for $u \in U$. That is, $det(u)$ indicates the degree to which a path between x and y becomes longer upon removal of a hinge vertex u from G. The *detour hinge vertex problem* aims to determine the hinge vertex u that maximizes $det(u)$ in G. Solving this problem can promote practical applications such as network stabilization at a limited cost [2].

An *articulation vertex* is a special case of a hinge vertex; its removal changes the finite distance between any pair of non-adjacent vertices x, y to infinity. Every articulation vertex has a maximum detour degree. In this study, we assume that a *circular-arc graph* does not include any articulation vertices, that is, the graph is biconnected.

Despite the existence of polynomial-time algorithms for these problems, there are many problems that are very computationally intensive for large graphs. So far, we have restricted graphs to a class of intersection graphs, and have researched and developed optimal or efficient algorithms for them.

In this study, we proposed an efficient algorithm for solving the detour hinge vertex problem on circular-arc graphs. Circular-arc graphs are used to model problems in periodic resource allocation found in the field of operations research. They have applications in various fields such as genetic research, traffic control, computer compiler design, and statistics [4]. An $O(n + m)$ time algorithm has been used to recognize a circular-arc graph [5]; these graphs have also been extensively discussed in existing literature [6]. Circular-arc graphs belong to the superclass of interval graphs, and have a wider range of practical applications. This study attempts to arrange an algorithm applicable to circular-arc graphs without increasing the time complexity, based on the algorithm for the detour hinge vertex problem that has been developed for interval graphs. Therefore, this study is an interesting theme from the point of view of computational complexity theory in the field of graph theory, and is significant both in terms of practical applications and advances in computational theory.

The remainder of this paper is organized as follows. In Section 2, we discuss previous research related to hinge vertex problems on intersection graphs. Section 3 presents some definitions of circular-arc models and graphs and the notations used. Section 4 describes the lemmas useful for constructing the algorithm for solving the detour hinge vertex problem. Section 5 describes the steps and complexity of the proposed algorithm used for solving this problem. Finally, Section 6 concludes the paper.

II. PREVIOUS WORKS

Lemma 1, proposed by Chang et al. [1], characterizes the hinge vertices of a simple graph G. Using Lemma 1, the problem of finding all hinge vertices in a simple graph can be solved in $O(n^3)$ time.

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

Several studies on hinge vertices have been conducted in recent years. Ho et al. [7] presented an $O(n)$ time algorithm to find all hinge vertices in *permutation graphs*; some minor errors in their approach have been corrected [8]. Hsu et al. [9] presented an $O(n)$ time algorithm for solving this problem on *interval graphs*. Honma and Masuyama [10] developed an $O(n)$ time algorithm for solving it on circular-arc graphs. The class of *trapezoid graphs* contains both interval and permutation graphs. Bera et al. [11] developed an $O(n\log n)$ time algorithm for solving the problem on such graphs. Recently, Honma and Nakajima [12] presented an $O(n^2)$ algorithm for solving it in circular trapezoid graphs, a superclass of trapezoid graphs.

For the detour hinge vertex problem, Honma and Nakajima presented an $O(n^2)$ time algorithm for solving it in interval graphs [3] and permutation graphs [13]. Recently, they also constructed an $O(n^2)$ time algorithm for solving the probrem of finding the maximum influential hinge vertices in interval graphs [14].

III. DEFINITIONS

Here, we introduced the terms and notations used in the paper.

A. Circular-arc Model and Graph

First, we shall define the *circular-arc model* before defining the circular-arc graph. Consider a unit circle C and a family A of n arcs $A_1, A_2, \dots, A_n$ along the circumference of C. Each arc A_i has two endpoints, namely the *left endpoint* a_i and *right endpoint* b_i, and $A_i = [a_i, b_i]$. The left endpoint a_i (resp., right endpoint b_i) is the last point of A_i that is encountered when walking counterclockwise along A_i (resp., clockwise). Without loss of generality, the coordinates of all the left and right endpoints are distinct and are assigned clockwise positions with consecutive integer values $1, 2, \dots, 2n$. The arc numbers i, j are assigned to each arc in the increasing order of the right endpoints b_i's, i.e., $A_i < A_j$ if $b_i < b_j$. Note that an arc A_i with $a_i > b_i$ is called a *feedback arc*. All these geometric representations comprise the definition of a *circular-arc model*. Figure 1(a) illustrates a circular-arc model M, which consists of 12 arcs (A_1 and A_2 are feedback arcs). Table 1 lists the details of M.

This model is considered *proper* if any two arcs do not cover the entire circumference C. In this study, since we only considered proper circular-arc models and graphs, the word "proper" shall be omitted henceforth.

A graph $\boldsymbol{G = (V, E)}$ is called a circular-arc graph if there exists a family of arcs $\boldsymbol{A = \{A_1, A_2, \dots, A_n\}}$ such that there is a one-to-one correspondence between vertex $\boldsymbol{i} \in \boldsymbol{V}$ and the arc $\boldsymbol{A_i} \in \boldsymbol{A}$ such that an edge $\boldsymbol{(i, j)} \in \boldsymbol{E}$ if and only if $\boldsymbol{A_i}$ intersects with $\boldsymbol{A_j}$ in $\boldsymbol{M}$. Figure 1(b) illustrates the circular-arc graph $\boldsymbol{G}$ corresponding to $\boldsymbol{M}$ shown in Fig. 1(a). In this example, the hinge vertices of $\boldsymbol{G}$ are vertices $\boldsymbol{v_1}$, $\boldsymbol{v_2}$, $\boldsymbol{v_5}$, and $\boldsymbol{v_{11}}$. The detour degree of each hinge vertex is $\boldsymbol{det(v_1) = det(v_2) = 2}$, $\boldsymbol{det(v_{11}) = 1}$, and $\boldsymbol{det(v_5) = 3}$. If hinge vertex $\boldsymbol{v_5}$ is removed from $\boldsymbol{G}$, the distance between vertices $\boldsymbol{v_3}$ and $\boldsymbol{v_7}$ lengthens from 2 to 5. Thus, the detour degree of vertex $\boldsymbol{v_5}$ is $\boldsymbol{det(v_5) = 3}$; vertex $\boldsymbol{v_5}$ is the maximum detour hinge vertex in $\boldsymbol{G}$.

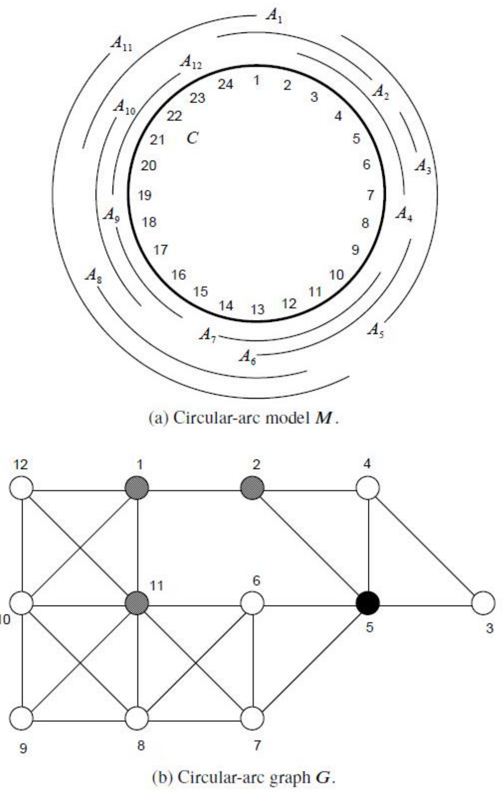

Figure 1: Circular-arc Model and Graph

Table 1: Details of Circular-arc Model *M*

i	1	2	3	4	5	6	7	8	9	10	11	12
a_i	20	24	5	2	3	8	9	12	15	16	11	19
b_i	1	4	6	7	10	13	14	17	18	21	22	23

B. Extended Circular-arc Model

Next, we shall introduce an *extended circular-arc model* (EM) constructed from M. First, M is cut at points between 1 and $2n$. Then, it is unrolled onto the real horizontal line. Each arc $A_i = [a_i, b_i]$ in M is also changed to interval I_i. Hereafter, for clarity, the arcs in M and EM are denoted as A_i and I_i, respectively. For each I_i, $1 \le i \le n$, copies I_{i+n} and I_{i-n} are created by shifting $2n$ to the right and left, respectively. This process can be executed in $O(n)$ time [10]. Figure 2 illustrates the EM constructed from M shown in Fig. 1. For simplicity, we use "arc," "interval," and "vertex" are used interchangeably if no confusion arises.

The neighborhood of vertex i is the set of all the vertices adjacent to i, and it is denoted by $N(i)$. Moreover, a neighbourhood in which i itself is included, called the *closed neighbourhood* and denoted by $N[i]$. For an interval I_i, $1 \le i \le n$, in EM, we define $l_a(i) = k$, where $a_k = \min\{ a_j \mid j \in N[i]\}$ and $r_b(i) = k$, where $b_k = \max\{ b_j \mid j \in N[i]\}$. Table 2 lists the details of EM illustrated in Fig. 2.

C. Shortest Circular Circuit

The *shortest circular circuit* (SCC) is a set of the smallest number of arcs covering the entire circumference C in M. In Fig. 1, the sample of SCC is $\langle A_1, A_2, A_5, A_7, A_{11}, A_1 \rangle$.

Let U be a set that consists of all hinge vertices of the circular-arc graph G. Set U is classified into U_1 and U_2 according to the following conditions: $U_1 = \{u \mid u \in SCC, u \in U\}$ and $U_2 = \{u \mid u \notin SCC, u \in U\}$. Regarding the example shown in Fig. 2, SCC is $\langle v_1, v_2, v_5, v_7, v_{11}, v_1 \rangle$ and a hinge vertex set $U = \{v_1, v_2, v_5, v_{11}\}$; $U_1 = \{v_1, v_2, v_5, v_{11}\}$; and $U_2 = \emptyset$.

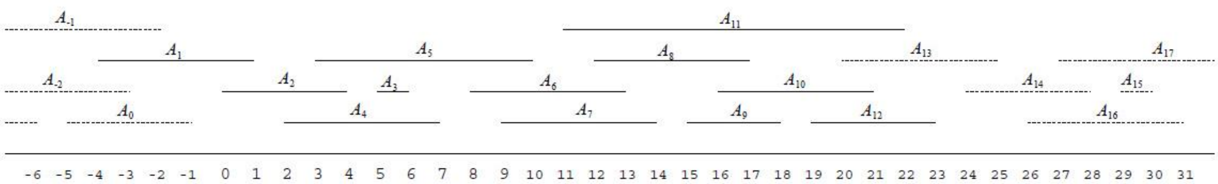

Figure 2: Extended Circular-arc Model *EM*

Table 2: Details of arrays $N(i)$, $l_a(i)$, and $r_b(i)$

i	...	1	2	3	4	5	6	7	8	9	10	11	12	...
a_i	...	-4	0	5	2	3	8	9	12	15	16	11	19	...
b_i	...	1	4	6	7	10	13	14	17	18	21	22	23	...
$N(i)$	...	{2, 10, 11, 12}	{1, 4, 5}			{2, 3, 4, 6, 7}		{5, 6, 8, 11}				{1, 6, 7, 8, 9, 10, 11, 12}		...
$l_a(i)$	...	-1	1	4	2	2	5	5	6	11	11	6	11	...
$r_b(i)$	...	2	5	5	5	7	11	11	11	11	13	13	13	...

IV. PROPERTIES OF HINGE VERTICES OF CIRCULAR-ARC GRAPH

We have described some lemmas that are useful for constructing the algorithm for solving the detour hinge vertex problem on a circular-arc graph G.

Lemma 2([10]): Let M be a circular-arc model and $G = (V, E)$ be a circular-arc graph with u, x, y $(x < y) \in V$ corresponding to M. Let EM be an extended circular-arc model constructed from M. A vertex u is a hinge vertex for x and y of G if and only if at least one of the following conditions holds in EM.

1. I_x does not intersect I_y, $I_{r_b(x)}$ does not intersect I_y, and I_u is the only interval intersecting both I_x and I_y.
2. I_y does not intersect I_{x+n}, $I_{r_b(y)}$ does not intersect I_{x+n}, and I_u is the only interval intersecting both I_y and I_{x+n}.

Lemma 2 was proposed by Honma et al. [10]. Using this lemma, all the hinge vertices in a circular-arc graph can be obtained in $O(n)$ time.

Lemma 3: Let SCC be the shortest circular circuit of a circular-arc graph G. Then, any vertex in G is adjacent to at most three vertices in SCC.

(Proof) Consider the shortest circular circuit $SCC = \langle v_1, v_2, v_3, v_4, \dots, v_l, v_1 \rangle$ of length l.

Suppose that vertex v', which is not included in SCC, is adjacent to four vertices v_1, v_2, v_3, and v_4 in SCC. Here, there are cycles $\langle v', v_4, \dots, v_l, v_1, v' \rangle$ of length $l - 1$ (Fig. 3). This contradicts the assumption that SCC is the shortest circular cycle.

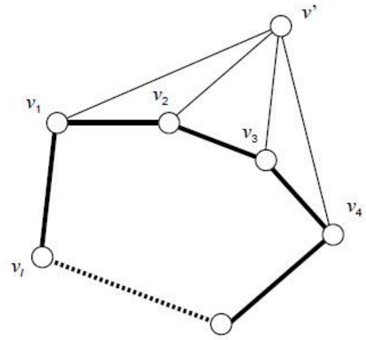

Figure 3: Illustration of Lemma 3.

Lemma 4: Let SCC be the shortest circular circuit of a circular-arc graph G. If vertex u is a hinge vertex of G, and it is not in SCC, then the detour degree of u is 1. In other word, $det(u) = 1$ for $u \notin U_2$.

(Proof) Since vertex u is a hinge vertex of G, according to Lemma 2, there exist intervals I_x, I_y, and I_u in EM such that I_x does not intersect I_y, and I_u is the only interval intersecting both I_x and I_y (Fig. 4). For simplicity, we assume that $x = l_a(u)$ and $y = r_b(u)$. Based on this assumption, since u is not in SCC, there is a need for two intervals v_1 and v_2 that intersect each other (v_1 covers x and u, v_2 covers y and u). In this case, the shortest path between v_x and v_y in a graph $G - u$ is $\langle v_x, v_1, v_2, v_y, v_x \rangle$. Then, $d_{G-u}(v_x, v_y) = 3$; $det(u) = 1$.

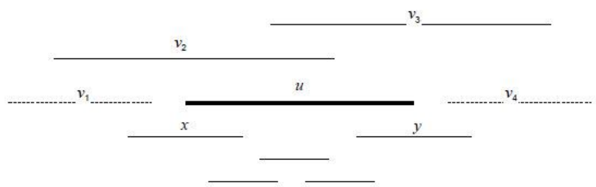

Figure 4: Illustration of Lemma 4

Hsu et al. [9] presented Lemma 5. By using this lemma, the shortest length between two vertices in a circular-arc graph G can be obtained in $O(n)$ time.

Lemma 5([9]): Let M be a circular-arc model and G be a circular-arc graph corresponding to M. Any shortest length query between two vertices in G can be answered in $O(1)$ time (it requires a preprocessing that runs in $O(n)$ time using $O(n)$ space data structure).

V. ALGORITHM AND ANALYSIS

In this section, we present the algorithm for solving the detour hinge vertex problem on a circular-arc graph; an outline of this algorithm is provided below.

First, we obtained a hinge vertex set U by applying the algorithm introduced by Honma et al. [10]. Next, we compute the shortest circular circuit SCC of G and construct $U_1 = \{u \mid u \in SCC, u \in U\}$ and $U_2 = \{u \mid u \notin SCC, u \in U\}$. From Lemma 4, $det(u) = 1$ for $u \notin U_2$. We obtained $det(u)$ for $u \in U_1$ by computing $d_{G-u}(x,y) - 2$ for all pairs $x, y \in N(u)$ adjacent to u. Finally, we obtained u as the maximum detour hinge vertex such that $det(u)$ has maximum value for $u \in U$.

Algorithm 1: MDHV algorithm

Input: Arcs $A_i = [a_i, b_i]$ of a circular-arc model M.
Output: Maximum detour hinge vertices of a circular-arc graph G.

(Step 1) /* Preparation */
1: Construct an EM from M;
2: Find the hinge vertex set U of G by applying Honma's algorithm [10];
3: Find a shortest circular circuit (SSC) of G;
4: Divide all hinge vertex set U to U_1 and U_2;

(Step 2) /* Obtain $det(u)$ for $u \in U_2$ */
for $u \in U_2$ **do**
    $det(u) := 1$;
end

(Step 3) /* Obtain $det(u)$ for $u \in U_1$ */
for $x, y \in N(u)$, $u \in U_1$ **do**
    $det(u) := \max\{d_{G-u}(x,y) - 2\}$;
end

(Step 4) /* Obtain maximum detour hinge vertex */
The maximum detour hinge vertex is k such that
$d(k) = \max\{ d(u) \mid u \in U\}$;

We described the MDHV algorithm and the analysis of the complexity in each step. The inputs to the MDHV algorithm are the left and right endpoints of the circular-arc model M; the output is a vertex number with the maximum detour degree.

Step 1 is the preprocessing step. EM is constructed from input M, and all the hinge vertices of the circular-arc graph G are obtained using the algorithm proposed by Honma et al. [10]. Next, SCC of G is computed, and the hinge vertex sets U_1 and U_2 are constructed. All these processes can be executed in $O(n)$ time.

In Step 2, $det(u)$ is obtained for $u \in U_2$. From Lemma 4, $det(u) = 1$ for $u \in U_2$. This step can be completed in $O(n)$ time.

In Step 3, $max\{ d_{G-u}(x,y) \mid x, y \in N(u)\}$ is computed for $u \in U_1$ to obtain the detour degree $det(u)$. According to Lemma 3, any vertex in G is adjacent to at most three vertices in SCC. This implies that the sum of the degrees of $v_i \in SCC$ needs to be less than $3n$, i.e., $\sum_{i \in SCC} \rho(v_i) \leq 3n$, where $\rho(v)$ denotes the degree of vertex v. Therefore, $O(n^2)$ time is required to compute $max\{ d_{G-u}(x,y) \mid x, y \in N(u)\}$.

Step 4 can be run in $O(n)$ time. Based on these steps, we introduced the following theorem:

Theorem 1: Let G be a circular-arc graph corresponding to a circular-arc model M. The proposed algorithm solves the detour hinge vertex problem on G in $O(n^2)$ time when the input is a set of arcs of M where the arcs are sorted with respect to the right endpoints.

In the following, we discuss the efficiency and contribution of the MDHV algorithm developed in this study.

The MDHV algorithm first computes all hinge vertices of a given circular-arc graph. for a given circular-arc graph. This process can execute in $O(n^3)$ time using the properties of Lemma 1 in a naive way, but can complete in $O(n)$ time by applying our algorithm [10]. Next, we compute the detour degree for each hinge vertex extracted, which requires $O(n^3)$ time in the naive method.

We focused on shortest circular circuits included in circular-arc graphs and showed their useful features in Lemmas 2 and 3. The MDHV algorithm efficiently derives the detour degree of each hinge vertex in $O(n^2)$ time by applying Lemma 2 and 3, and the shortest path algorithm on the circular-arc graph [9].

VI. CONCLUSION

In this study, we proposed an MDHV algorithm for solving the detour hinge vertex problem on a circular-arc graph in $O(n^2)$ time. The algorithm uses the algorithm proposed by Honma et al. to find the hinge vertices and that proposed by Hsu et al. to determine the shortest path across a circular-arc graph. We have also developed an algorithm [3] that solves the same problem in $O(n^2)$ time for interval graphs, which are subclasses of circular-arc graphs. We showed that it can be realized for circular-arc graphs of a class larger than interval graphs without increasing the time complexity. For this reason, we think this study is also worthy. In future studies, we shall consider reducing the complexity of the algorithm and extending the results to other graphs.

ACKNOWLEDGEMENT

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, and Cooperative Education/ Research Project between Toyohashi University of Technology and National Institute of Technology.

REFERENCES

[1] J. M. Chang, C. C. Hsu, Y. L. Wang, and T. Y. Ho, "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.

[2] H. Honma and S. Masuyama, "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.

[3] H. Honma, Y. Nakajima, Y. Igarashi, and S. Masuyama, "Algorithm for finding maximum detour hinge vertices of interval graphs," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E97-A, no. 6, pp. 1365–1369, 2014.

[4] M. Pal, "Intersection graphs: An introduction," *Annals Pure Appl. Math. (APAM)*, vol. 4, no. 1, pp. 43–91, 2013.

[5] R. M. McConnell, "Linear-time recognition of circular-arc graphs," *Algorithmica*, vol. 37, pp. 93–147, 2003.

[6] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs.* New York, NY, USA: Academic Press, 1980.

[7] T. Y. Ho, Y. L. Wang, and M. T. Juan, "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] H. Honma, K. Abe, and S. Masuyama, "Erratum and addendum to a linear time algorithm for finding all hinge vertices of a permutation graph," *Inf. Process. Lett.*, vol. 111, no. 18, pp. 891–894, 2011.

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

[10] H. Honma and S. Masuyama, "An optimal parallel algorithm for finding all hinge vertices of a circular-arc graph," *IEICE Trans. Fundam. Electron. Commun. Comput.*, vol. E91.A, no. 1, pp. 383–391, 2008.

[11] D. Bera, M. Pal, and T. K. Pal, "An efficient algorithm for finding all hinge vertices on trapezoid graphs," *Theory Comput. Syst.*, vol. 36, no. 1, pp. 17–27, 2003.

[12] H. Honma, Y. Nakajima, and S. Masuyama, "An algorithm for hinge vertex problem on circular trapezoid graphs," *J. Inf. Process.*, vol. 25, pp. 945–948, 2017.

[13] H. Honma, Y. Nakajima, Y. Igarashi, and S. Masuyama, "Algorithm for identifying the maximum detour hinge vertices of a permutation graph," *IEICE Trans. Fundam. Electron. Commun. Comput.*, vol. E98-A, no. 6, pp. 1161–1167, 2015.

[14] H. Honma, Y. Nakajima, and S. Masuyama, "An algorithm for the influential hinge vertex problem on interval graphs," *J. Inf. Process.*, vol. 28, pp. 1047–1051, 2020.