2001年度日本オペレーションズ・リサーチ学会
春季研究発表会

A parallel algorithm for finding all hinge vertices of a Circular-Arc graph

01506161 Kushiro National College of Technology HONMA Hirotoshi
01603863 Toyohashi University of Technology *MASUYAMA Shigeru

1 Introduction

Given 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 $V - u$. We define the distance $d_G(x, y)$ as the length of the shortest path between vertices x and y in G. Chang et al. [1] defined that $u \in V$ is a *hinge vertex* if there exist two vertices $x, y \in V - \{u\}$ such that $d_{G-u}(x, y) > d_G(x, y)$.

There exists a trivial $O(n^3)$ sequential algorithm for finding all hinge vertices of a simple graph by a result in Ref. [1], e.g., Theorem 1 in this paper. In general, it is known that more efficient sequential or parallel algorithms can be developed by restricting classes of graphs. For instance, Chang et al. presented an $O(n + m)$ time algorithm for finding all hinge vertices of a *strongly chordal graph* [1]. Ho et al. presented a linear time algorithm for all hinge vertices of a *permutation graph* [4]. Recently, we provided a parallel algorithm, which runs in $O(\log n)$ time with $O(n)$ processors, for finding all hinge vertices of an *interval graph* [3]. In this paper, we shall propose a parallel algorithm, which runs in $O(\log n)$ time with $O(n)$ processors on CREW PRAM (Concurrent-Read Exclusive-Write Parallel Random Access Machine) for finding all hinge vertices of a *circular-arc graph* [5].

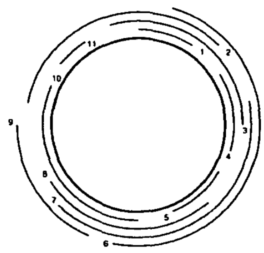

Figure 1: Circular-arc model CA

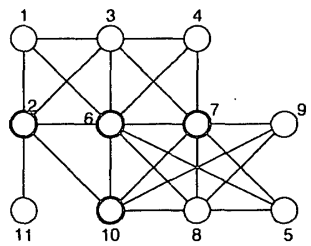

Figure 2: Circular-arc graph G

2 Preliminaries

We first illustrate the *circular-arc model* before defining the circular-arc graph. Suppose that a unit circle C and a set A of n circular-arcs $A_1, A_2, ..., A_n$ along the circumference of C. Each circular-arc A_i has two endpoints, *left endpoint* a_i and *right endpoint* b_i, such that a_i (resp. b_i) is the last point of A_i that we encounter when walking along A_i counterclockwise (resp. clockwise). We denote circular-arc A_i by $[a_i, b_i]$. All left and right endpoints are labeled clockwise with consecutive interger values 1, 2, ..., $2n$. Without loss of generality, assume that all endpoints of n circular-arcs are distinct. We also assume that a circular-arc number is assigned to each circular-arc in increasing order of their right endpoints b_i's, i.e., $A_i < A_j$ if $b_i < b_j$. The geometric representation described above is called a *circular-arc model (CA)*. Fig. 1 shows a circular-arc model CA, consisting of eleven circular-arcs.

A graph $G = (V, E)$ is a *circular-arc graph* if there exists a circular-arc set A such that there is a one-to-one correspondence between the vertices $i \in V$ and the circular-arc $A_i \in A$ in such a way that an edge $(i, j) \in E$ if and only if A_i intersects with A_j in CA. The circular-arc graph G, corresponding to the circular-arc model CA illustrated in Fig. 1, is shown in Fig. 2.

We cut circular-arc CA at endpoint a_1 and next open it out flat. This process changes circular-arcs in CA to real line segments on the horizontal line in the plane. In particular, a circular-arc A_i with $a_i > b_i$ is called a *feedback circular-arc*. Here, if there is the feedback circular-arc $A_i = [a_i, b_i]$ in CA, we modify it to $A_i = [a_i - 2n, b_i]$ and generate an extra circular-arc $A_{i'} = [a_i, b_i + 2n]$. The geometric representation obtained by applying the procedure described above is called an *extended circular-arc model (ECA)*. The ECA constructed from the circular-arc model CA illustrated in Fig. 1 is shown in Fig. 3.

In the following, we define some terms used in this paper. We denote by vertex i, throughout the paper, a vertex in G corresponding to a circular-arc A_i. A set of all vertices adjacent with vertex i is denoted by $N(i)$.

We denote by $M(i)$ the number j of the largest circular-arc A_j ($b_j \geq b_i$) intersecting with A_i. Similarly, we denote by $SM(i)$ the number j of the second largest circular-arc A_j ($b_j \geq b_i$) intersecting with A_i. However, let $M(i) = i$, $SM(i) = i$, respectively when such a circular-arc A_j does not exist. Also, $D(i) = \{k \mid b_{SM(i)} < k < b_{M(i)}\}$ is defined as a *detect set*. In

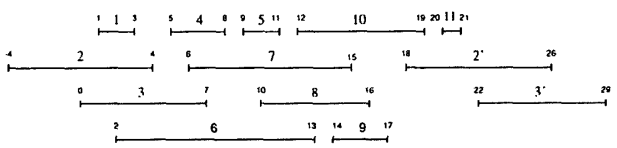

Figure 3: Extended circular-arc model ECA

Table 1: $M(i), SM(i), D(i)$

i	1	2	3	4	5	6	7	8	9	10	11	2'	3'
a	1	-4	0	5	9	2	6	10	14	12	20	18	22
b	3	4	7	8	11	13	15	16	17	19	21	26	29
M	6	6	7	7	8	10	10	10	10	2'	2'		
SM	3	3	6	6	7	8	9	9	9	10	11		
D	8,..,12	8,..,12	14	14	$\emptyset$	17,18	18	18	18	20,21,..,25	22,..,25		

addition, we define *represent vertex sets* (RVS). Let $u_1 < u_2 < ... < u_m$ be different values among $M(i)$'s, $i \in V$ and we divide V into vertex sets $V_1, V_2, ..., V_m$, where $V_j = \{i \mid M(i) = u_j\}$ and $V_j \neq \emptyset$. Next, v_j is the smallest vertex among V_j's, which is called *represent vertex* of V_j. We also define RVS as a set consisting of all vertices v_j, $j = 1, 2, ..., m$.

Table 1 shows $M(i)$, $SM(i)$, $D(i)$ for the extended circular-arc model ECA illustrated in Fig. 3. In this table, RVS is $\{1, 3, 5, 6, 10\}$.

3 Some properties of the hinge vertices in circular-arc graphs

Theorem 1 was due to Chang et al. [1]. It is used to identify the hinge vertices of a simple graph. We apply this theorem for efficiently finding hinge vertices of a circular-arc graph.

Theorem 1 *For a graph $G = (V, E)$, a vertex $u \in V$ is a hinge vertex of G if and only if there exist two nonadjacent vertices $x, y \in N(u)$ such that u is the only vertex adjacent with both x and y in G.* □

Lemma 1 *Vertex u is a hinge vertex of a circular-arc graph G if and only if either of the following two conditions holds in ECA.*

(1) $A_x < A_y$, $A_u = A_{M(x)}$, $a_y \in D(x)$ and $b_{M(y)} < a_x + 2n$.

(2) $A_x < A_y$, $A_u = A_{M(y)}$, $a_x + 2n \in D(y)$ and $b_{M(x)} < a_y$. □

Lemma 2 *Assume that x, y are two vertices of a circular-arc graph $G = (V, E)$. We now consider the vertex set V_u such that $V_u = \{v \mid M(v) = u\}$. Then $D(x) \supseteq D(y)$ for $x, y (x < y) \in V_u$.* □

Lemma 3 *Let $G = (V, E)$ be a circular-arc graph. Assume that $x, y \in V$ are two vertices in G with $x < y$. Then, either $M(x) = M(y)$ or $D(x) \cap D(y) = \emptyset$.* □

We propose a procedure for finding a hinge vertex. Before introducing its formal description, we illustrate it by using the example of Table 1 in detail. We first compute $M(i)$, $SM(i)$, $D(i)$ for i; $1 \leq i \leq n$, and next obtain a represent vertex set RVS. By Lemma 1-(2), if there exist x and y satisfying $A_x < A_y$, $A_u = A_{M(y)}$, $a_x + 2n \in D(y)$ and $b_{M(x)} < a_y$, if and only if u is a hinge vertex of a circular-arc graph. Assume that there exists k such that $k \in D(i)$, $i \in RVS$, and $k \in D(v)$, for v; $i \leq v \leq j$. For the example of Table 1, $k = 18$, $i = 6$ and $j = 9$. We find x satisfying $a_x + 2n = 18$, that is, $x = 2$. Finally, we examine whether there exists y satisfying $b_{M(x)} < a_y$ with $i \leq y \leq j$. For the example of Table 1, $M(x) = 6, b_{M(x)} = 13 < a_y$ when $y = 9$. Hence, $M(9) = 10$ is a hinge vertex of a circular-arc graph. And by Lemma 2, it sufficies to apply $D(i)$ for $i \in RVS$. Also by Lemma 3, it is executed in $O(n)$ time.

Algorithm PHV
Input: The left and right end points $[a_i, b_i]$ in CA.
Output: The set of hinge vertices.
Step 1 (Generation of ECA)
  for all A_i, $1 \leq i \leq n$, **in parallel do**
    If $A_i = [a_i, b_i]$ is a feedback circular-arc **then** we change A_i into $A_i := [a_i - 2n, b_i]$ and generate an extra circular-arc $A_{i'} := [a_i, b_i + 2n]$.
Step 2 (Construction of M_i, SM_i)
  for all A_i, $1 \leq i \leq n$, **in parallel do**
    Compute $M(i)$, where $M(i)$ is the largest $j (\geq i)$ such that A_j intersects with A_i.
  for all A_i, $1 \leq i \leq n$, **in parallel do**
    Compute $SM(i)$, where $SM(i)$ is the second largest $j (\geq i)$ such that A_j intersects with A_i.
Step 3 (Construction of RVS, and $D(i), i \in RVS$)
  $RVS := \{1\}$
  for all i, $2 \leq i \leq n$, **in parallel do**
    If $M(i) > M(i-1)$ and then $RVS := RVS \cup \{i\}$.
  for all i, $i \in RVS$ **in parallel do**
    Compute $D(i) = \{k \mid b_{SM(i)} < k < b_{M(i)}\}$.
    i satisfying $D(i) = \emptyset$ is removed.
Step 4 (Finding all hinge vertices)
  for all $D(x)$, $x \in RVS$ **in parallel do**
    If there exist x and y satisfying $A_x < A_y$, $A_u = A_{M(x)}$, $a_y \in D(x)$ and $b_{M(y)} < a_x + 2n$, then u is a hinge vertex of the circular-arc graph.
  for all $D(y)$, $y \in RVS$ **in parallel do**
    If there exist x and y satisfying $A_x < A_y$, $A_u = A_{M(y)}$, $a_x + 2n \in D(y)$ and $b_{M(x)} < a_y$, then u is a hinge vertex of the circular-arc graph.
End of Algorithm

Theorem 2 *Given a circular-arc graph G, Algorithm PHV finds the set of all hinge vertices of G in $O(\log n)$ time using $O(n)$ processors on CREW PRAM.* □

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, *Information Sciences* **99** (1997) 173-182.

[2] A. Gibbons and W. Rytter, *Efficient parallel algorithms*, Cambridge University Press (1988).

[3] H. Honma and S. Masuyama, A parallel algorithm for finding all hinge vertices of an Interval graph, to appear in IEICE Trans. Information and Systems.

[4] Ting-Yem Ho, Yue-Li Wang and Ming-Tsan Juan, A linear time algorithm for finding all hinge vertices of a permutation graph, *Inform. Process. Lett.* **59** (1996) 103-107.

[5] A. S. Rao and C. P. Rangan, Optimal parallel algorithms on circular-arc graphs, *Inform. Process. Lett.* **33** (1989) 147-156.