数理解析研究所講究録
1093 巻 1999 年 27-32

Tractable and Intractable Problems on Generalized Chordal Graphs*

上原 隆平 (Ryuhei Uehara)
uehara@komazawa-u.ac.jp

Faculty of Natural Sciences, Komazawa University

Abstract

A generalized chordal graph is characterized by some positive integer $k \geq 3$, and whose each cycle of length greater than k has a chord. Several tractable and intractable problems on a k-chordal graph are considered. When k is a fixed positive integer, the recognition problem for k-chordalness is in NC. On a k-chordal graph, the maximal acyclic set problem is in NC when $k=3$, and the problem is in RNC for any fixed $k>3$. Next we show that the recognition problem for k-chordalness is coNP-complete for $k=\Theta(n)$, where n is the number of vertices. We also show that for any positive constant ϵ, any NP-complete problem on a general graph is also NP-complete on a k-chordal graph even if $k=\Theta(n^{\epsilon})$.

Key words: Generalized chordal graph, maximal acyclic subgraph, NC algorithm, RNC algorithm, NP-complete problem, coNP-complete problem.

1 Introduction

A *chordal graph* is the graph whose each cycle of length greater than 3 has a chord. Chordal graphs have applications in Gaussian elimination and databases and have been the object of much algorithmic study. On the other hand, chordal graphs are also important from the graph theoretical point of view. The class of chordal graphs is subclass of the class of perfect graphs, and superclass of the class of interval graphs (see [17, 12] for details). It is natural to generalize about the length of the cycle on the chordal graph. We introduce a notion of a *k-chordal graph* whose each cycle of length greater than k has a chord. An ordinary chordal graph is a 3-chordal graph, and so-called chordal bipartite graph can be defined by a bipartite 4-chordal graph. In this paper, we consider the complexities of the recognition problem for k-chordalness, and some graph problems on a k-chordal graph.

We first consider the case that k is a fixed positive integer. We extend the characterization of a 3-chordal graph by Edenbrandt [8] in a nontrivial way, and show that the recognition problem of a k-chordal graph is in NC for any fixed k. We modify the NC algorithm, and show that the bipartite k-chordal graph also can be recognized in parallel. We remark that for a bipartite 4-chordal graph, the ordinary chordal bipartite graph, only polynomial time algorithms are known (see e.g. [14, 18]).

For a fixed k, we next consider the *maximal acyclic set* (MAS) problem on a k-chordal graph. The MAS problem is to find a maximal vertex induced acyclic subgraph for a given graph. The MAS problem has been solved partially; in the case that some properties are added to the acyclicness [19, 21], and on some restricted graphs [3]. Our results are in the latter case; the problem is in NC on a 3-chordal graph, and in RNC on a k-chordal graph for any fixed $k>3$.

We here remark that no graph has a simple cycle of length greater than n, where n is the number of vertices. Hence we can regard every graph as an n-chordal graph. On the other hand, there are several problems such that they are NP-complete on a general graph, and they are polynomial time solvable on a 3-chordal graph. For example, maximum independent set problem, chromatic number problem, and maximum clique problem are NP-complete on a general graph [9], and they are polynomial time solvable on a 3-chordal graph [10]. (These problems are even in NC on a 3-chordal graph [17, 12].) These facts motivate us to consider the case of general k as a function of n. We first show that the recognition problem for k-chordalness is coNP-complete for $k=\Theta(n)$. We next show that, for any positive con-

*This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Encouragement of Young Scientists, 10780203, 1998.

stant ϵ, any NP-complete problem on a general graph is also NP-complete on a k-chordal graph even for $k = \Theta(n^\epsilon)$. This also implies coNP-completeness of the recognition problem for k-chordalness even for $k = \Theta(n^\epsilon)$.

2 Preliminaries

A graph $G = (V, E)$ consists of a finite set V of *vertices* and a collection E of 2-element subsets of V called *edges*. For $U \subseteq V$, the *subgraph of G induced by U* is the graph (U, F), where $F = \{\{u, v\} | \{u, v\} \in E$, and $u, v \in U\}$, and is denoted by $G[U]$. We define the *neighborhood* of a vertex v in G, denoted $N_G(v)$, by the set of vertices u with $\{u, v\} \in E$. For a set of vertices U, $N_G(U)$ is defined by $\{v | v \in N_G(u)$ for $u \in U\}$. The *degree* of a vertex v in G is $|N_G(v)|$. A graph G is *d-regular* if every vertex in G has degree d. A sequence of the vertices $v_0, v_1, \cdots, v_l$ is a *path*, denoted by $(v_0, v_1, \cdots, v_l)$, if $\{v_j, v_{j+1}\} \in E$ for each $0 \le j \le l - 1$. The path is *simple* if all the vertices are distinct with each other. We say the first and last vertices of the sequence the *endpoints* of the path. The *length* of a path is the number of edges on the path. A *cycle* is a path beginning and ending with the same vertex. A cycle is *simple* if all the vertices are distinct with each other except the common endpoint. A *Hamiltonian cycle* is a simple cycle that visits all vertices in the graph. The *Hamiltonian cycle problem* is to find a Hamiltonian cycle in a given graph. A vertex set $U = \{v_0, v_1, \cdots, v_{l-1}\}$ is *cyclic* if there exists at least one permutation π of size l such that $(v_{\pi(0)}, v_{\pi(1)}, \cdots, v_{\pi(l-1)}, v_{\pi(0)})$ is a simple cycle. An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. A graph is *k-chordal* if each of its cycles of length greater than k has a chord. (An ordinary chordal graph can be defined by a 3-chordal graph [6].) The *recognition problem for k-chordalness* is to decide whether for a given graph and a given integer k the graph is k-chordal. An *acyclic set* is a set of vertices that contains no cyclic vertex set. A *maximal acyclic set* (MAS) of G is an acyclic set that is not properly contained in any other acyclic set of G. The MAS problem of G is to find an MAS of G. We say a cyclic vertex set U is *minimal* if $G[U]$ has no chord. We say *cyclic decomposition set* of G, denoted by $\mathcal{C}[G]$, a family of all minimal cyclic vertex set in G.

Proposition 1 *For any graph G, G is k-chordal if and only if each minimal vertex set in $\mathcal{C}[G]$ has size at most k.*

A hypergraph $H = (V, E)$ consists of a finite set V of vertices and a collection E of non-empty subsets of V called *hyperedges*. The *dimension* in H is the maximum size of a hyperedge in E. An *independent set* in H is a subset of V not containing any edge of H. A *maximal independent set* (MIS) of H is an independent set of H that is not properly contained in some other independent set. The MIS problem on H is to find an MIS in H.

The definitions and the details of the complexity classes (NC, RNC, NP, and coNP) and the parallel computation models (EREW PRAM, and CRCW PRAM) can be found in [13] and omitted here.

3 k-Chordal Graphs with Fixed k

In this section, we suppose that k is a given fixed positive integer greater than or equal to 3.

3.1 Recognition of a k-Chordal Graph

The following lemma is crucial in this subsection:

Lemma 2 *A graph $G = (V, E)$ is not k-chordal if and only if there is a subset U of V with size k such that*

(1) *$G[U]$ is a path $(v_0, v_1, \cdots, v_{k-1})$ of length $k-1$ and the number of the edges in $G[U]$ is $k-1$, and*

(2) *v_0 is reachable to v_{k-1} in graph $G[V - \hat{U}]$, where $\hat{U}$ is defined by $N_G(U - \{v_0, v_{k-1}\}) - \{v_0, v_{k-1}\}$.*

Intuitively, when U witnesses that G is not k-chordal, $G[U]$ is just a path, not including extra edges, and v_0 is reachable to v_{k-1} through the vertices not in the neighbors of the intermediate vertices in the path $G[U]$ (see Figure 1).

Proof. We first assume that G is not k-chordal. Then there is at least one minimal cycle C of length at least $k+1$. We let U be k consecutive vertices on C. Since C has no chord, $G[U]$ is just a path, and $\hat{U}$ contains no vertex in $C - \hat{U}$. Moreover, the endpoints of the path $G[U]$ are reachable to each other through the vertices $G[C - U]$, or not through the vertices in $\hat{U}$. Thus both conditions (1) and (2) hold for the U.

Next we assume that we can take a set U satisfying both conditions (1) and (2). Let P be the shortest path from v_0 to v_{k-1} not through the vertices in $\hat{U}$. Then, we have a cycle C joining P and $G[U]$. Since v_0 and v_{k-1} is not adjacent to each other by (1), the length of C is at least $k+1$. Thus, it is sufficient to

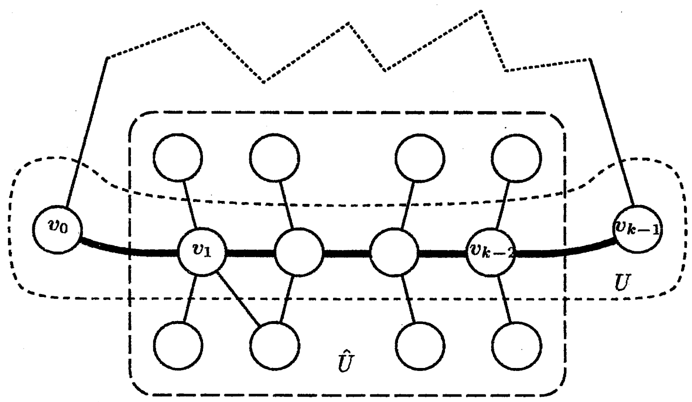

Figure 1: k-chordalness of a graph

show that C has no chord. By (1), C has no chord with two vertices both in P. Since P is the shortest path, C has no chord with two vertices both in P. Moreover, since P and $\hat{U}$ have no common vertices, C has no chord between the vertices in P and $G[U]$. Thus G is not k-chordal. ∎

Now we are ready to show the NC recognition algorithm of a k-chordal graph.

Algorithm RECOG

Input: A graph $G = (V, E)$, and an integer k.

Output: "Yes" or "No"

1. For every set U of size k do in parallel;
   - 1-1. If $G[U]$ is not a path return;
   - 1-2. If two endpoints in the path $G[U]$ are reachable to each other in the graph $G[V - \hat{U}]$, return "No";
2. If there exists a set returning "No", output "No"; else output "Yes".

Theorem 3 Algorithm RECOG solves the recognition problem for k-chordalness in $O(\log n)$ time using $O(n^{k+1} + n^k m)$ processors.

Proof. The correctness of the algorithm follows immediately from Lemma 2. Thus it is sufficient to show the complexity of Algorithm RECOG.

In addition to the processors corresponding to the vertices and edges, we assign a processor to each set of vertices of size k. Thus, the step 1-1 can be performed in a constant time. The step 1-2 can be carried out using an algorithm that computes connected components. The parallel algorithm for this task runs in $O(\log n)$ time and uses $O(n + m)$ processors [20].

Next we count the number of processors. The number of the set of vertices of size k is $\binom{n}{k} = O(n^k)$. For each set we check the reachability that requires $O(n+m)$ processors. Hence, we need at most a total of $O(n^{k+1} + n^k m)$ processors. ∎

Now we turn to the recognition problem of a k-chordal bipartite graph.

Lemma 4 A graph G is not bipartite if and only if $\mathcal{C}[G]$ contains at least one minimal cyclic vertex set of odd size.

Proof. A graph G is not bipartite if and only if it contains at least one odd length cycle (see e.g. [6]). We assume that G contains at least one odd length cycle. A chord of any odd length cycle divides the cycle into an odd length cycle and an even length cycle. Thus, $\mathcal{C}[G]$ must contain at least one minimal cyclic vertex set of odd size. ∎

Theorem 5 A k-chordal bipartite graph can be recognized in $O(\log n)$ time using $O(n^{k+1} + n^k m)$ processors.

Proof. In addition to the recognition for k-chordalness, by Proposition 1 and Lemma 4, it is sufficient to check whether the graph contains at least one minimal cyclic vertex set of odd size being less than or equal to k. The check can be done by replacing the step 1-1 in Algorithm RECOG by the following two steps:

1-1(a). If $G[U]$ contains an odd length cycle return "No";

1-1(b). If $G[U]$ is not a path return;

This implies the theorem. ∎

3.2 The MAS Problem on a k-Chordal Graph

In this section, we consider the MAS problem on a k-chordal graph. For any given graph $G = (V, E)$, we denote by H_G a hypergraph $(V, \mathcal{C}[G])$. That is, the hypergraph H_G contains every minimal cyclic vertex set in G as a hyperedge. We moreover let W_G be a set of vertices defined by $\{v | v \text{ is not in any } U \in \mathcal{C}[G]\}$. That is, the set W_G contains every vertex not included in any cycle in G. Then we have the following lemma.

Lemma 6 For any graph G and a subset U of V, U is an MAS of G if and only if $W_G \subseteq U$ and $U - W_G$ is an MIS of H_G.

Proof. For lack of space, we omit the proof. ∎

Now we have the following algorithm for solving the MAS problem.

Algorithm MAS

Input: A graph $G = (V, E)$.

Output: An MAS U.

1. Find W_G and set $U := W_G$; $V := V - W_G$.
2. Find an MIS of H_G, and add them into U.

The correctness of the algorithm follows immediately from Lemma 6. Now assuming that the graph G is k-chordal, and we have the main theorem in this section:

Theorem 7 The MAS problem on a k-chordal graph is in RNC for any fixed k greater than 3. An MAS of a 3-chordal graph can be computed in $O(\log^4 n)$ time with $O(n^3)$ processors of an EREW PRAM.

Proof. We assume that the given graph is k-chordal. We show that Algorithm MAS can be implemented in NC except finding an MIS. In addition to the processors corresponding to vertices and edges in G, we assign each vertex set of size at most k. The assignment requires $\sum_{i=3}^{k} \binom{n}{k} = O(n^k)$ processors. The step 1 can be done by the following steps:

1-1. In parallel, each processor corresponding to a vertex set of size at most k marks its all elements if it is cyclic.

1-2. Set all unmarked vertices into W_G.

The steps can be done in a constant time since k is a constant. Moreover, through the steps, each processor can know whether the vertex set is minimal cyclic set. By Proposition 1, every minimal cyclic set can be found by the steps. This implies that the algorithm can construct the hypergraph H_G after the step 1 in a constant time. Thus, using the following facts, we have the theorem: An MIS in a hypergraph of dimension 3 can be computed in $O(\log^4 n)$ time using $n + m$ processors of an EREW PRAM, where n and m are the number of vertices and edges, respectively [4]. The MIS problem on a hypergraph of dimension d is in RNC for any fixed positive integer d [11]. ∎

Remark: Chen proposed an NC algorithm for finding a maximal k-cycle free set on a given chordal graph [1]. Using his result, we can find an MAS of a 3-chordal graph in $O(\log^4 n)$ time with $O(n + m)$ processors of an EREW PRAM [2].

4 k-Chordal Graphs with General k

We now turn to consider the k-chordal graphs such that k is a function of n, the number of vertices.

4.1 coNP-completeness of the Recognition for k-Chordalness

Theorem 8 The recognition problem for k-chordalness is coNP-complete for $k = \Theta(n)$.

Proof. Hamiltonian cycle problem is NP-complete even on a 3-regular graph [9]. We show a polynomial time reduction from the problem to the complement of the recognition problem. Let $G = (V, E)$ be a given 3-regular graph. Let $G' = (V', E')$ be the graph obtained from G by the following replacements:

(1) For each edge in E, replace it by a path of length 2 (see Figure 2(1)).

(2) For each vertex in V, replace it by the gadget given in Figure 2(2).

Let $|V| = n$, then $|E| = \frac{3}{2}n$ since G is 3-regular. For the resulting graph G', it is not difficult to see that $|V'| = n(n+5) + \frac{3}{2}n = n^2 + \frac{13}{2}n$ and $|E'| = n(n+7) + 2\frac{3}{2}n = n^2 + 10n$. This reduction can be done in polynomial time.

We now consider the complement of the recognition problem for k-chordalness on G' as follows:

Input: G' and $k = n^2 + n - 1$.

Question: Is G' not k-chordal?

It is sufficient to show that G has a Hamiltonian cycle if and only if G' is not k-chordal. We first remark that each edge in G' has at least one endpoint of degree 2, and it can not be a chord. That is, the resulting graph G' has no chord. Thus, G' is not k-chordal if and only if G' has a cycle of length greater than k.

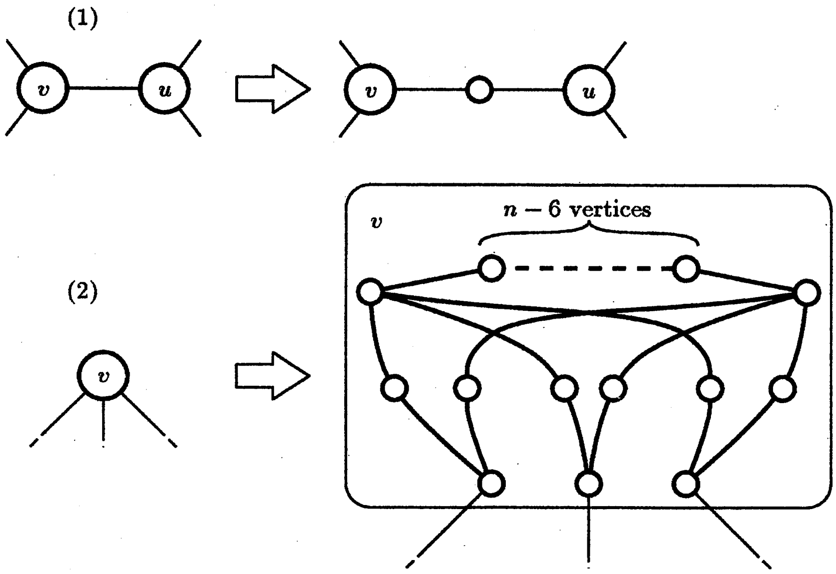

Figure 2: Replacements

We first assume that G has a Hamiltonian cycle, or G has a simple cycle of length n. Then, we can trace n gadgets following the order of the vertices on the Hamiltonian cycle, and in each gadget we can have a simple path of n vertices (including the entrance and exit vertices). Thus, counting the number of vertices on the trace, we have a cycle of length $n^2+n=k+1$ in G', consequently, G' is not k-chordal.

We next assume that G' has a simple cycle of length at least $k+1=n^2+n$. Since each gadget has three points to enter and exit, the simple cycle can visit each gadget at most once. Thus each gadget can contribute at most n vertices to the cycle. To derive a contradiction, we moreover assume that the cycle visits at most $n-1$ gadgets. The number of the edges in G is $\frac{3}{2}n$, which is equal to the number of the vertices not included in the gadgets in G'. Thus the number of the vertices on the cycle is bounded above by $\frac{3}{2}n+(n-1)n=n^2+\frac{1}{2}n<n^2+n$, which contradicts. Therefore, the cycle visits every gadget once. Thus the order of the gadgets on the cycle gives a Hamiltonian cycle in G. ∎

4.2 Lower Bound of NP-completeness

For any NP-complete problem on a general graph, we give a lower bound of k such that it is also NP-complete on a k-chordal graph by a padding argument:

Theorem 9 Every NP-complete problem on an n-chordal graph is also NP-complete on a cn^ϵ-chordal graph for any fixed positive constants c and ϵ.

Proof. We reduce the problem on an n-chordal graph to the problem on a cn^ϵ-chordal graph. For a given graph $G=(V,E)$ with n vertices, we construct a new graph $G'=(V',E')$ that consists of $f(n)=\left\lceil \left(\frac{1}{c}\right)^{\frac{1}{\epsilon}} n^{\frac{1}{\epsilon}-1}\right\rceil$ copies of G as follows: Fix any vertex v in V, joint $f(n)$ copies of G using a path of length $f(n)-1$ through every copy of v.

Clearly, this is a polynomial time reduction. Moreover, G' is an n-chordal graph with $n'=nf(n)$ vertices. Since $n=c\left(\left(\frac{1}{c}\right)^{\frac{1}{\epsilon}} n^{\frac{1}{\epsilon}}\right)^\epsilon \le c(nf(n))^\epsilon = cn'^\epsilon$, this completes the proof. ∎

Combining Theorem 8 and the padding in the proof of Theorem 9, we also have the following corollary:

Corollary 10 For any fixed positive constant ϵ, the recognition problem for k-chordalness is coNP-complete even for $k=\Theta(n^\epsilon)$.

5 Concluding Remarks

Recently, parameterized problems are widely investigated [7], and the recognizing k-chordalness can be regarded as a kind of such problems. From the point of view, it is further work to determine if the recognition problem for k-chordalness is fixed-parameter tractable.

We can regard Lemma 2 as a nontrivial generalization of [8, Lemma 23] in the sense that we can obtain the same claim from Lemma 2 by setting $k=3$.

These lemmas use the property of a k-chordal graph straightforwardly. On the other hand, Naor, Naor, and Schäffer [17], and Klein [12] use two nontrivial characterizations of a 3-chordal graph; one is the *intersection graph characterization*, and the other is the *perfect elimination ordering*. From both algorithmical and graph theoretical points of view, it is worth finding such nontrivial characterizations of a k-chordal graph.

Some results for 4-chordal bipartite graphs immediately state NP-completeness of finding the following sets on a 4-chordal graph: Steiner tree and dominating set [16], variants of dominating set [5], and Hamiltonian cycle [15]. On the other hand, as mentioned in Introduction, several NP-complete problems on a general graph are polynomial time solvable on a 3-chordal graph. The following are natural questions: whether the NP-complete problems on a 4-chordal graph are polynomial time solvable on a 3-chordal graph; and whether the polynomial time solvable problems on a 3-chordal graph are NP-complete on a 4-chordal graph.

Acknowledgment

I thank Professor Zhi-Zhong Chen, who read my earlier draft, sent me his unpublished document, noticed the efficient algorithm for the MAS problem on a 3-chordal graph, and gave me some fruitful suggestions. My thanks also to Professor Koichi Yamazaki for discussions and encouragement.

References

[1] Z.-Z. Chen. *NC* Algorithms for Finding Maximal k-Cycle Free Sets in Simple Graphs. manuscript, 1993.

[2] Z.-Z. Chen. Personal communication. 1998.

[3] Z.-Z. Chen and X. He. Parallel Algorithms for Maximal Acyclic Sets. *Algorithmica*, 19:354–368, 1997.

[4] E. Dahlhaus, M. Karpinski, and P. Kelsen. An efficient parallel algorithm for computing a maximal independent set in a hypergraph of dimension 3. *IPL*, 42:309–313, 1992.

[5] P. Damaschke, H. Müller, and D. Kratsch. Domination in Convex and Chordal Bipartite Graphs. *IPL*, 36:231–236, 1990.

[6] R. Diestel. *Graph Theory*. Springer, 1996.

[7] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer, 1999.

[8] A. Edenbrandt. Chordal Graph Recognition is in *NC*. *IPL*, 24:239–241, 1987.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[10] F. Gavril. Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph. *SIAM J. on Comp.*, 1(2):180–187, 1972.

[11] P. Kelsen. On the Parallel Complexity of Computing a Maximal Independent Set in a Hypergraph. In *Proc. 24th STOC*, pages 339–350. ACM, 1992.

[12] P.N. Klein. Efficient Parallel Algorithms for Chordal Graphs. *SIAM J. on Comp.*, 25(4):797–827, 1996.

[13] J. van Leeuwen. *Handbook of Theoretical Computer Science*. Elsevier Science Publishers, 1990.

[14] A. Lubiw. Doubly Lexical Orderings of Matrices. *SIAM J. on Comp.*, 16(5):854–879, 1987.

[15] H. Müller. Hamiltonian Circuit in Chordal Bipartite Graphs. *Disc. Math.*, 156:291–298, 1996.

[16] H. Müller and A. Brandstädt. The *NP*-Completeness of Steiner Tree and Dominating Set for Chordal Bipartite Graphs. *TCS*, 53:257–265, 1987.

[17] J. Naor, M. Naor, and A.A. Schäffer. Fast Parallel Algorithms for Chordal Graphs. *SIAM J. on Comp.*, 18(2):327–349, 1989.

[18] R. Paige and R.E. Tarjan. Three Partition Refinement Algorithms. *SIAM J. on Comp.*, 16(6):973–989, 1987.

[19] D. Pearson and V.V. Vazirani. Efficient Sequential and Parallel Algorithms for Maximal Bipartite Sets. *J. of Algorithms*, 14:171–179, 1993.

[20] Y. Shiloach and U. Vishkin. An $O(\log n)$ parallel connectivity algorithm. *J. of Algorithms*, 3:57–67, 1982.

[21] R. Uehara and Z.-Z. Chen. Parallel Algorithms for Maximal Linear Forests. *The Trans. of the IEICE*, E80-A(4):627–634, 1997.