アルゴリズム 33－4
(1993. 5. 28)

部分k木を辺彩色する並列アルゴリズム

周 暁　中野 眞一　西関 隆夫

東北大学大学院情報科学研究科

部分k木に対しては数多くの組合せ問題を解く並列アルゴリズムが知られている．しかし，辺彩色問題に対しては効率のよい並列アルゴリズムが得られていなかった．本論文は与えられた部分k木を最小色数で辺彩色する効率のよい並列アルゴリズムを与える．

A Parallel Algorithm for Edge-Coloring Partial k-Trees

Xiao Zhou†, Shin-ichi Nakano†† and Takao Nishizeki†††
Department of System Information Sciences
Graduate School of Information Sciences
Tohoku University, Sendai 980, Japan
† zhou@ecei.tohoku.ac.jp
†† nakano@ecei.tohoku.ac.jp
††† nishi@ecei.tohoku.ac.jp

Abstract

A parallel algorithm is considered efficient if its time complexity is polylogarithmic with polynomially many processors on the PRAM. *NC* is the class of problems that have such algorithms. Many combinatorial problems can be efficiently solved in parallel for partial k-trees. The edge-coloring problem is one of a few combinatorial problems for which no NC algorithms have been obtained for partial k-trees. This paper gives a first NC parallel algorithm which finds an optimal edge-coloring of a given partial k-tree.

1. Introduction

This paper deals with the edge-coloring problem which asks to color all edges of a given graph, using a minimum number of colors, so that no two adjacent edges are colored with the same color. It is known that many combinatorial problems can be solved very efficiently for series-parallel graphs or partial k-trees [ACPD, AL, BPT, C, TNS]. Such a class of problems has been characterized in terms of "forbidden graphs" or "extended monadic logic of second order" [ACPD, AL, BPT, C, TNS]. The edge-coloring problem does not belong to such a class of the "maximum (or minimum) subgraph problems," and is indeed one of the "edge-covering problems" which do not appear to be efficiently solved for partial k-trees. Furthermore the edge-coloring problem is one of a few problems for which no NC algorithms have been obtained even for partial k-trees. However NC parallel algorithms have been obtained for the following classes of graphs: planar graphs with maximum degree $\Delta \geq 9$ [CN]; outerplanar graphs [CD, GR]; and series-parallel multigraphs [ZSN].

In this paper we give an NC algorithm which solves the edge-coloring problem for partial k-trees G in $O(\log n)$ time with $O(n/\log n)$ processors if a "decomposition tree" of any partial k-tree $H = (V_H, E_H)$ can be found in $O(\log |V_H|)$ time with $O(|V_H|/\log |V_H|)$ processors. In the paper n denotes the number of vertices in G. It is known that a decomposition tree of H can be found in $O(\log^3 |V_H|)$ time with $O(|V_H|)$ processors [BK]. Therefore an optimal edge-coloring of G can be found total in $O(\log^3 n)$ time with $O(n)$ processors. This is the first NC algorithm for optimally edge-coloring partial k-trees. We use the parallel random-access machine (PRAM) as a model of parallel computation. In a PRAM all the processors access to a common memory and run synchronously. This model can be divided into subclasses by allowing or disallowing two or more processors to gain access to the same memory location: exclusive-read and exclusive-write (EREW); concurrent-read and exclusive-write (CREW); and concurrent-read and concurrent-write

(CRCW). The parallel computation model we use is an concurrent-read exclusive-write parallel random access machine (CREW PRAM).

2. Terminology and definitions

In this section we give some definitions. Let $G = (V, E)$ denote a graph with vertex set V and edge set E. We often denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. The paper deals with *simple* graphs without multiple edges or self-loops. An edge joining vertices u and v is denoted by (u, v). The *degree* of vertex $v \in V(G)$ is denoted by $d(v, G)$ or simply by $d(v)$. The *maximum degree* of G is denoted by $\Delta(G)$ or simply by Δ.

The class of *k-trees* is defined recursively as follows:

(a) A complete graph with k vertices is a k-tree.

(b) If $G = (V, E)$ is a k-tree and k vertices $v_1, v_2, \cdots, v_k$ induce a complete subgraph of G, then $H = (V \cup \{w\}, E \cup \{(v_i, w) | 1 \leq i \leq k\})$ is a k-tree where w is a new vertex not contained in G.

(c) All k-trees can be formed with rules (a) and (b).

A graph is a *partial k-tree* if and only if it is a subgraph of a k-tree. Thus partial k-trees are simple graphs. In this paper we assume that k is a constant.

A *decomposition tree* of a graph $G = (V, E)$ is a tree $T = (V_T, E_T)$ with a family V_T of subsets of V satisfying the following properties:

- $\bigcup_{X_i \in V_T} X_i = V$;
- for every edge $e = (v, w) \in E$, there is a node $X_i \in V_T$ with $v, w \in X_i$; and
- if node X_j lies on the path in T from node X_i to node X_l, then $X_i \cap X_l \subseteq X_j$.

Figure 1 illustrates a partial 3-tree and its decomposition tree. The *treewidth* of a decomposition tree $T = (V_T, E_T)$ is $\max_{X_i \in V_T} |X_i| - 1$. The treewidth of G, denoted by *treewidth*(G) is the minimum treewidth of a decomposition tree of G, taken over all possible decomposition trees of G. It is known that every graph with treewidth $\leq k$ is a partial k-tree, and conversely, that every partial k-tree has a decomposition tree of treewidth $\leq k$.

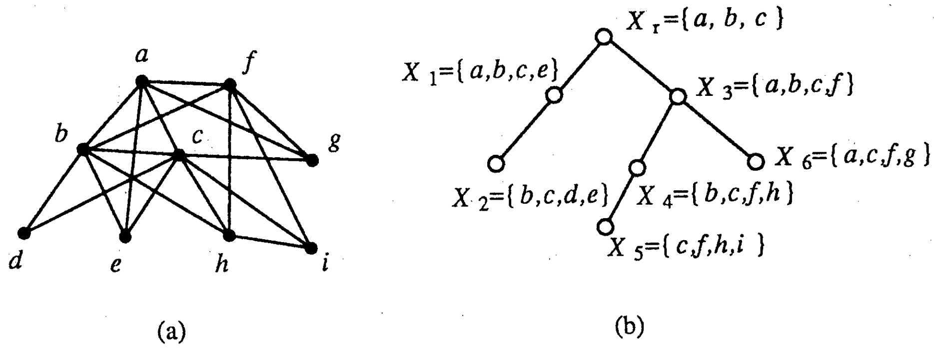

Figure 1. (a) A partial 3-tree and (b) its decomposition tree.

3. NC Parallel Algorithm

In this section we give an NC parallel algorithm for the edge-coloring problem of partial k-trees. It runs in $O(\log n)$ time with $O(n/\log n)$ processors if a decomposition tree of any partial k-tree $H = (V_H, E_H)$ can be found in $O(\log|V_H|)$ time with $O(|V_H|/\log|V_H|)$ processors. It is known that a decomposition tree of H can be found in $O(\log^3|V_H|)$ time using $O(|V_H|)$ processors [BK]. Therefore the edge-coloring problem of partial k-trees G can be solved total in $O(\log^3 n)$ time using $O(n)$ processors.

There is a general theorem which relates the number of processors to the parallel time and to the total number of basic operations m that have to be carried out [GR].

Theorem 3.1. *Let A be a given algorithm with a parallel computation time of t. Suppose that A involves a total number of m computational operations. Then A can be implemented using p processors in $O(m/p + t)$ parallel time.*

It is rather straightforward to modify Bodlaender's sequential algorithm [B] of time complexity $O(n\Delta^{2^{2(k+1)}})$ to an NC parallel algorithm as in the following theorem, the proof of which is omitted in the paper for lack of space.

Theorem 3.2. *Let G be a partial k-tree with maximum degree Δ given by its decomposition tree. Then there is an algorithm which solves the edge-coloring problem for G in $O(\log n)$ time using $O(n\Delta^{2^{4(k+1)}}/\log n)$ processors.*

Although the algorithm in Theorem 3.2 is an NC parallel algorithm, it uses too many processors and is not optimal. However, the algorithm runs optimally for every partial k-tree G with small maximum degree $\Delta(G)$, say $\Delta(G) < 5k$. Therefore it suffices to give an optimal parallel algorithm for edge-coloring any partial k-tree G with large maximum degree $\Delta(G)$, say $\Delta(G) \geq 5k$. We have the following lemmas, the proofs of which are omitted in the paper for lack of space.

Lemma 3.3. *If a partial k-tree G has maximum degree $\Delta(G) \geq 2k$, then $\chi'(G) = \Delta(G)$.*

Lemma 3.4. *Let $G = (V, E)$ be a partial k-tree. Then E can be partitioned into subsets $E_1, E_2, \cdots, E_s$ such that the subgraphs H_j, $1 \leq j \leq s$, of G induced by E_j satisfy*

(a) $\Delta(H_j) = 2k$ *for each* j, $1 \leq j \leq s-1$, *and*

(b) $3k \leq \Delta(H_s) = \Delta(G) - 2k(s-1) < 5k$.

Furthermore such a partition of E can be found in $O(\log n)$ time using $O(n/\log n)$ processors if a decomposition tree of any partial k-tree $H = (V_H, E_H)$ can be found in $O(\log |V_H|)$ time using $O(|V_H|/\log |V_H|)$ processors.

Then we have the following algorithm to edge-color a partial k-tree $G = (V, E)$.

```
Procedure Edge-Color(G);
begin
 if Δ(G) < 5k then
  begin
   find a decomposition tree of G;
   edge-color G using χ'(G) colors by the algorithm in Theorem 3.2
  end
```

6 **else**

7 **begin**

8 find a partition $E_1, E_2, \cdots, E_s$ of E mentioned in Lemma 4.1; { Lemma 3.4 }

9 **for** each i, $1 \leq i \leq s$, **in parallel, do**

10 find edge-color H_i using $\Delta(H_i)$ colors; { Theorem 3.2 }

{ H_i is the subgraph of G induced by the edges in E_i. }

11 extend the edge-colorings of $H_1, H_2, \cdots, H_s$ to an edge-coloring of G using $\Delta(G)$ colors

12 **end**

end;

By Theorems 3.1 and 3.2 and Lemmas 3.3 and 3.4 we conclude the following theorem.

Theorem 3.5. *Suppose that a decomposition tree of any partial k-tree $H = (V_H, E_H)$ can be found in $O(\log |V_H|)$ time with $O(|V_H|/\log |V_H|)$ processors. Then an edge-coloring of a given partial k-tree G with a minimum number of colors can be found in $O(\log n)$ time with $O(n/\log n)$ processors.*

Proof. The correctness of the algorithm above is an immediate consequence of Theorems 3.1 and 3.2 and Lemmas 3.3 and 3.4. Therefore we prove the claim on the time and processor complexities. From the supposition line 3 can be done in $O(\log n)$ parallel time. By Theorem 3.2 and Lemma 3.3 lines 4, 8, 9 and 10 can be done in $O(\log n)$ parallel time. Line 11 can be done in $O(1)$ parallel time. Furthermore the total number of computational operations above is $O(n)$. Therefore by Theorem 3.1 the algorithm above runs in $O(\log n)$ parallel time using $O(n/\log n)$ processors. $\mathcal{Q.E.D.}$

Acknowledgment

We would like to thank Dr. Hitoshi Suzuki for various discussions. This research is partly supported by Grant in Aid for Scientific Research of the Ministry of Education, Science, and Culture of Japan under a grant number: General Research (C) 04650300.

References

[ACPD] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese, *An algebraic theory of graph reduction,* Tech. Rept. 91-36, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, 1991.

[AL] S. Arnborg and J. Lagergren, *Easy problems for tree-decomposable graphs,* Journal of Algorithms, 12, 2, pp.308-340, 1991.

[B] H. L. Bodlaender, *Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees,* Journal of Algorithms, 11, 4, pp.631-643, 1990.

[BK] H.L. Bodlaender and T. Kloks, *Better algorithms for the pathwidth and treewidth of graphs*, Proceedings of 18'th International Colloquium on Automata, Languages and Programming, Springer Verlag, Lecture Notes in Computer Science 510, pp.544-555, Berlin, 1991.

[BPT] R.B. Borie, R.G. Parker and C.A. Tovey, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families,* Algorithmica, 7, pp.555-581, 1992.

[C] B. Courcelle, *The monadic second-order logic of graphs I: Recognizable sets of finite graphs,* Information and Computation, 85, pp.12-75, 1990.

[CD] Y. Caspi and E. Dekel, *A near-optimal parallel algorithm for edge-coloring outerplanar graphs,* manuscript, Computer Science Program, University of Texas at Dallas, Richardson, Tx 75083-0688, 1992.

[CN] M. Chrobak and T. Nishizeki, *Improved edge-coloring algorithms for planar graphs*, J. Algorithms, 11, pp.102-116, 1990.

[FW] S. Fiorini and R.J. Wilson, *Edge-Colouring of Graphs,* Pitman, London, 1977.

[GR] A. Gibbons and W. Rytter, *Efficient Parallel Algorithms,* Cambridge, 1988.

[Hol] I.J. Holyer, *The NP-completeness of edge-coloring,* SIAM J. on Computing, 10, pp.718-720, 1981.

[NC] T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms,* North-Holland, Amsterdam, 1988.

[TN] O. Terada and T. Nishizeki, *Approximate algorithms for the edge-coloring of graphs,* Trans. Inst. of Electronics and Communication Eng. of Japan, J65-D, 11, pp. 1382-1389, 1982.

[TNS] K. Takamizawa, T. Nishizeki, and N. Saito, *Linear-time computability of combinatorial problems on series-parallel graphs,* J. of ACM, 29, 3, pp. 623-641, 1982.

[ZSN] X. Zhou, H. Suzuki and T. Nishizeki, *Sequential and parallel algorithms for edge-coloring series-parallel multigraphs*, Proc. of Third Conference on Integer Programming and Combinatorial Optimization, April 29-May 1, 1993, Erice, Italy.