Vol. 34 No. 1 Transactions of Information Processing Society of Japan Jan. 1993

Regular Paper

$O(\log^* n)$ Time Parallel Algorithm for Computing Bounded Degree Maximal Subgraphs☆

TOMOYUKI UCHIDA† and SATORU MIYANO†

By using the vertex coloring technique, we give a fast parallel algorithm that finds a maximal vertex-induced subgraph of degree at most k, where k is a given constant. This algorithm runs in $O(\log^* n)$ time using $O(n)$ processors on an EREW PRAM for a constant degree graph $G=(V, E)$ with $|V|=n$. We also describe an $O(\log^* m)$ time $O(m)$ processor EREW PRAM algorithm for finding a maximal edge-induced subgraph of degree at most k, where $m=|E|$. For constant degree graphs, we show that the coloring technique works very successfully to devise faster parallel algorithms with fewer numbers of processors.

1. Introduction

For a given integer k, we consider the problem of finding a maximal subset of vertices (resp., edges) whose induced subgraph is of degree at mosk k. We denote the problem by VIMS(k) (resp., EIMS(k)). Shoudai and Miyano[10),11)] have shown that VIMS(k) and EIMS(k) are in NC by describing algorithms which employ the parallel maximal independent set (MIS) algorithm.[8),9)] In their algorithms, maximal independent sets are repeatedly computed k^2 times for VIMS(k) and $2k$ times for EIMS(k), respectively. Later, Diks et al. [4)] have independently given the same results as Shoudai and Miyano[10)] with the same argument. If we apply the fast parallel MIS algorithm in Ref. 5) to the algorithms in Ref. 4), 10), we can easily see that VIMS(k) (resp., EIMS(k)) for graphs of constantly bounded valence can be solved in $O(\log^* n)$ (resp., $O(\log^* m)$) time with $O(n)$ (resp., $O(m)$) processors on an EREW PRAM, where n (resp., m) is the number of vertices (resp., edges) of an input graph.

Since $\log^* n$ grows extremely slowly and can be viewed as a constant for all practical purposes, it is important to focus on the constants k and the degree Δ of an input graph. In this paper, we apply the coloring technique to VIMS(k) and EIMS(k), and obtain faster parallel algorithms for these problems from this point of view. When the vertex coloring algorithm by Ref. 5) is used, for an input graph with degree at most Δ, our algorithm runs k times as fast as the algorithm in Ref. 4), 10) equipped with the MIS algorithm in Ref. 5). Moreover, the number of processors needed by our algorithm is Δ/k times as few as that of their algorithm. If the degree Δ of an input graph satisfies $\Delta=o(\log n)$, our method also provides an algorithm faster than that in Ref. 4), 10) even though we apply the $O((\log n)^2)$ time parallel MIS algorithm by Ref. 9) to their algorithm.

Furthermore, the edge coloring technique works efficiently to solve EIMS(k). While their algorithms[4),10)] for EIMS(k) need to solve MIS $2k$ times, it is sufficient for our algorithm to compute the edge coloring only once. Therefore, our algorithm runs $2k$ times as fast for graphs with degree at most $\Delta=o(\log n)$.

2. Preliminaries

We consider a graph $G=(V, E)$ as an undirected graph without any multiple edges and self-loops. Let $|V|=n$ and $|E|=m$. For a subset $U\subseteq V$, we define $E[U]=\{\{u, v\}\in E|u, v\in U\}$. The graph $G[U]=(U, E[U])$ is called the *vertex-induced subgraph* of U. We define

† Research Institute of Fundamental Information Science, Kyushu University

☆ A preliminary version of this paper was presented in Kokyuroku of Research Institute of Mathematical Science, No. 754, pp. 104–114, Kyoto University, June, 1991.

$V[F]$ to be the set of endpoints of the edges in F for a subset $F \subseteq E$. We denote by $\langle F \rangle = (V[F], F)$ the graph formed from F and call it the *edge-induced subgraph* of F. For a vertex u of G, the degree of u is denoted by $d_G(u)$. For a graph G, the maximum degree of G is denoted by $deg(G)$.

A vertex coloring C of G is a mapping C: $V \to \mathbf{N}$ from the vertices to positive integers (colors), and it is *valid* if no two adjacent vertices have the same color.

Definition 1. Let $G=(V, E)$ be a graph and let $k \geq 0$ be any integer. The *maximum degree k vertex-induced maximal subgraph problem* (VIMS(k)) is to find a maximal subset $U \subseteq V$ such that $G[U]$ is of degree at most k.

Definition 2. Let $G=(V, E)$ be a graph and let $k \geq 1$ be any integer. The *maximum degree k edge-induced maximal subgraph problem* (EIMS(k)) is to find a maximal subset $F \subseteq E$ such that $\langle F \rangle$ is of degree at most k.

We assume an exclusive-read exclusive-write (EREW) PRAM model of computation where each processor is capable of executing a special operation which counts the number of bit 1's in a word together with conventional simple word and bit operations.[3] The word length is assumed to be $O(\log n)$. We define two functions F and H. Let

$F(0)=1$,

$F(i)=2^{F(i-1)}$, for $i>0$.

The function $H(n)=\log^* n$ is defined to be the smallest integer j such that $F(j) \geq n$. $H(n)(=\log^* n)$ grows extremely slowly and can be viewed as a constant for all practical purposes. (For instance, $H(2^{65536})=\log^* 2^{65536}=5$.)

Goldberg et al.[5] have presented a vertex coloring algorithm that yields the following lemma under the above conditions of the PRAM model:

Lemma 1 (Goldberg et al. [5]**).** *Let Δ be an integer. Given a graph $G=(V, E)$ with degree at most Δ, a valid vertex coloring of G with $\Delta+1$ colors can be computed in $O(\Delta(\Delta+\log^* n)\log \Delta)$ time on an EREW PRAM using Δn processors.*

3. Finding Bounded Degree Vertex-Induced Maximal Subgraphs

In this section we show an algorithm which solves VIMS(k) efficiently.

Theorem 1. *Let k and Δ be nonnegative integers with $0 \leq k \leq \Delta$. For a graph $G=(V, E)$ with degree at most Δ, VIMS(k) can be solved in $O(\log^* n)$ time using $O(n)$ processors on an EREW PRAM.*

Proof. Our VIMS algorithm takes a graph $G=(V, E)$ of degree at most Δ as an input and outputs a maximal subset $S \subseteq V$ such that $G[S]$ is of degree at most k.

We need to prepare some notations in order to describe the algorithm precisely.

Let $C: V \to \mathbf{N}$ be a ($\Delta+1$)-vertex coloring of G with degree at most Δ. For each $i=0, \cdots, \Delta$, let $C_i(V)=\{v \in V \mid C(v)=i\}$. For a subset $S \subseteq V$ and a vertex $v \in V$, let $U_v[S]$ be the set of vertices in S that are adjacent to v. For subsets W and U of vertices with $W \cap U=\phi$, let $E_U^W=\{\{v, w\} \mid v, w \in W,\ w \neq v$ and there is $u \in U$ such that $\{v, u\} \in E$ and $\{w, u\} \in E\}$.

The algorithm is described as follows:

VIMS Algorithm:

1 $S \leftarrow \phi; V' \leftarrow V; i \leftarrow 0;$
2 Compute *a* $(\Delta+1)$*-vertex coloring* C of $G=(V, E)$;
3 **while** $V' \neq \phi$ **do**
4 $\quad X \leftarrow C_i(V')$;
5 $\quad V' \leftarrow V' - X$;
6 $\quad Y \leftarrow \{v \in S \mid d_{G[S \cup X]}(v) > k\}$;
7 $\quad Y' \leftarrow \{v \in X \mid U_v[Y] \neq \phi\}$;
8 $\quad S \leftarrow S \cup (X - Y')$;
9 $\quad$ **if** $Y \neq \phi$ **then**
10 $\quad\quad$ Compute a $(k\Delta+1)$*-vertex coloring* D^i of $G_i=(Y', E_Y^{Y'})$;
11 $\quad\quad W \leftarrow Y'$; $j \leftarrow 0$;
12 $\quad\quad$ **while** $W \neq \phi$ **do**
13 $\quad\quad\quad S \leftarrow S \cup D_j^i(W)$;
14 $\quad\quad\quad W \leftarrow W - D_j^i(W)$;
15 $\quad\quad\quad W \leftarrow W - \{w \in W \mid deg(G[S \cup \{w\}]) > k\}$;
16 $\quad\quad\quad j \leftarrow j+1$
17 $\quad\quad$ **od**
18 $\quad\quad V' \leftarrow V' - \{w \in V' \mid deg(G[S \cup \{w\}]) > k\}$;
19 $\quad\quad i \leftarrow i+1$
20 $\quad$ **od**

We show that the algorithm computes a required maximal subset S. First it colors the input graph G with colors $0, \cdots, \Delta$ at line 2. Then, for each color i, the algorithm determines which vertices colored i are added to S. Let S_i, X_i, Y_i and Y_i' be the contents of S, X, Y and Y' at the end of the ith iteration of lines 3–20, respectively. We assume that S_{i-1} is a maximal subset of $C_0(V)\cup\cdots\cup C_{i-1}(V)$ which induces a subgraph of degree at most k. For $i=0$, the assumption holds obviously since S_0 is an independent set $C_0(V)$. We show that the induced subgraph $G[S_i]$ in $G[S_{i-1}\cup C_i(V)]$ is of degree at most k and is maximal.

Clearly, after executing lines 4-8, the graph $G[S]$ is of degree at most k, but may not be maximal. We now prove the induced subgraph $G[S_i]$ becomes of degree at most k and is maximal after lines 9-19. Let $D^i: Y'\to\mathbf{N}$ be the $(k\Delta+1)$-vertex coloring of the graph $G_i=(Y'_i, E^{Y'_i}_{Y_i})$ computed at line 10. For any two vertices v, w in Y' which are adjacent to a vertex in Y, we can see that $D^i(v)\neq D^i(w)$, since an edge $\{v, w\}$ is in $E^{Y'}_{Y}$ by the definitions of Y, Y' and $E^{Y'}_{Y}$. Since the vertices in V' at line 18, $d_{G[S_{i-1}\cup X_i]}(v)\le k$ for a vertex v in X_i. For a vertex w in S_{i-1}, $d_{G[S_{i-1}\cup X_i]}(w)\le\Delta$. Hence, from the definitions of Y, Y' and $E^{Y'}_{Y}$, the degree of $G_i=(Y'_i, E^{Y'_i}_{Y_i})$ is at most $k\Delta$. Therefore, by using the $(k\Delta+1)$-vertex coloring D^i of the graph G_i, the induced subgraph $G[S]$ can be made maximal, keeping the condition that the degree of the graph $G[S]$ is at most k. Hence, we can see that the induced subgraph $G[S_i]$ is of degree at most k and is maximal. Therefore, our algorithm can solve VIMS(k), correctly.

Finally, we show that our algorithm can compute the VIMS(k) in $O(\log^* n)$ time using $O(n)$ processors on an EREW PRAM when a constant degree graph is given as input. Let $T(G, \Delta)$ be the time needed to compute a valid $(\Delta+1)$-vertex coloring of the input graph G with degree at most Δ using $O(\Delta n)$ processors on an EREW PRAM. Hence, line 2 requires $T(G, \Delta)$ time using $O(\Delta n)$ processors on an EREW PRAM. We show that the time needed in the ith iteration of lines 3-20 as follows $(0\le i\le\Delta)$: Since the degree of the input graph G is at most Δ, lines 4-8 can be processed in $O(\log\Delta)$ time using Δn processors. Since $deg(G_i)\le k\Delta$, line 10 needs the time $T(G_i, k\Delta)$ using $O(k\Delta n)$ processors on an EREW PRAM. It is easy to see that $T(G_i, k\Delta)\le T(G, k\Delta)$ since $|Y'|\le|V|$. Since the graph G_i is colored with at most $k\Delta+1$ colors, the while loop (lines 12-17) repeats at most $k\Delta+1$ times. Therefore, the time needed in the ith iteration is $O(\log\Delta)+T(G, k\Delta)+O(k\Delta\log\Delta)$ time using $O(k\Delta n)$ processors on an EREW PRAM. Hence, the ith iteration runs in $T(G, k\Delta)$ time using $O(k\Delta n)$ processors on an EREW PRAM. Moreover, since the input graph G is colored with at most $\Delta+1$ colors, the while loop (lines 3-20) repeats at most $\Delta+1$ times. Line 3-20 requires $O(\Delta T(G, k\Delta))$ time using $O(k\Delta n)$ processors. Therefore, our algorithm runs in $O(\Delta T(G, k\Delta))$ time using $O(k\Delta n)$ processors.

When we apply Lemma 1 to our algorithm, it runs in $O(k\Delta^2(k\Delta+\log^* n)\log\Delta)$ time using $O(k\Delta n)$ processors. Hence, for the constant degree graphs, our algorithm runs in $O(\log^* n)$ time on an EREW PRAM using $O(n)$ processors. □

Remark 1. Using the MIS algorithm in Ref. 5), the algorithm in Ref. 4), 10) can also solve VIMS(k) in $O(\log^* n)$ time on an EREW PRAM using $O(n)$ processors for constant degree graphs. The MIS algorithm given by Ref. 5) uses a $(\Delta+1)$-coloring of an input graph G. Therefore, for an input graph G with degree at most Δ, their VIMS algorithm runs in $O(k^2\Delta^2(\Delta^2+\log^* n)\log\Delta)$ time with $O(\Delta^2 n)$ processors. Hence, our algorithm is faster than their algorithm with fewer numbers of processors.

4. Finding Bounded Degree Edge-Induced Maximal Subgraphs

In this section we apply the vertex coloring technique to EIMS(k).

Theorem 2. *Let k and Δ be positive integers with $1\le k\le\Delta$. For constant degree graphs, EIMS(k) can be solved in $O(\log^* m)$ time on an EREW PRAM using $O(m)$ processors where m is the number of edges of the input graph.*

Proof. The algorithm takes a graph $G=(V, E)$ with degree at most Δ as input, and outputs a maximal subset $F\subseteq E$ such that $\langle F\rangle$ is a graph of degree at most k.

Let $D:E\to\mathbf{N}$ be a $(2\Delta-1)$-edge coloring of G with degree at most Δ. For each $i=0, \cdots, 2\Delta-2$, let $D_i(E)=\{e\in E|D(e)=i\}$.

Formally the algorithm is described as follows:

EIMS Algorithm:

1 $F\leftarrow\phi$; $Z\leftarrow E$;

2 Compute a $(2\Delta-1)$-edge coloring D of $G=(V, E)$;

3 $i \leftarrow 0$;
4 **while** $Z \neq \phi$ **do**
5 $\quad F \leftarrow F \cup D_i(Z)$;
6 $\quad Z \leftarrow Z - D_i(Z)$;
7 $\quad Z \leftarrow Z - \{e \in Z \mid deg(\langle F \cup \{e\} \rangle) > k\}$;
8 $\quad i \leftarrow i+1$
9 **od**

We show the correctness of the algorithm. Let $F_0 = \phi$ and $Z_0 = E$. For $0 \leq i \leq 2\Delta - 2$, let F_i and Z_i be the contents of F and Z just after the ith iteration. We assume that F_{i-1} is a maximal subset of $D_0(Z_0) \cup \cdots \cup D_{i-1}(Z_{i-1})$ such that $\langle F_{i-1} \rangle$ is a maximal subgraph with the degree at most k of the graph $\langle D_0(Z_0) \cup \cdots \cup D_{i-1}(Z_{i-1}) \rangle$. It is easy to see that $deg(\langle F_i \rangle) \leq k$ since $D_i(Z_i)$ is a matching of $\langle Z_{i-1} \rangle$ and since each edge e in $D_i(Z_i)$ satisfies $deg(\langle F_{i-1} \cup \{e\} \rangle) \leq k$. We can also see that F_i is maximal subset of $F_{i-1} \cup D_i(Z_{i-1})$. Therefore, after $2\Delta - 1$ iterations, we see that the resulting F is a maximal set of edges such that $deg(\langle F \rangle) \leq k$.

Next, we show that the algorithm runs in $O(T(G) + \Delta \log \Delta)$ time on an EREW PRAM with p processors for $p \geq \Delta m$ where $T(G)$ is the time which our algorithm takes at line 2 on an EREW PRAM using p processors. In lines 3-9, since the number of colors of the edge coloring of the graph G is $2\Delta - 1$, it takes $O(\Delta \log \Delta)$ time on an EREW PRAM with m processors. Therefore, we can see that the algorithm runs in $O(T(G) + \Delta \log \Delta)$ time on an EREW PRAM with p processors for $p \geq \Delta m$.

For a constant degree input graph G with degree at most Δ, line 2 can be implemented in time $T(G) = O(\Delta(\Delta + \log^* m) \log \Delta)$ by constructing a line graph G' of G and computing a valid vertex coloring of G' with $2\Delta - 1$ colors. Hence, for the constant degree graphs, our algorithm runs in $O(\log^* m)$ time on an EREW PRAM using $O(m)$ processors. □

Remark 2. By using the MIS algorithm in Ref. 5), EIMS(k) can be solved by the algorithm in Ref. 4), 10) in $O(\log^* m)$ time on an EREW PRAM with $O(m)$ processors for constant degree graphs. However, for a constant degree graph G with degree at most Δ, the algorithm in Ref. 4), 10) must compute the $(2\Delta - 1)$-edge coloring $2k$ times to solve EIMS(k), since the MIS algorithm in Ref. 5) uses the $(\Delta + 1)$-vertex coloring algorithm. Therefore, the algorithm in Ref. 4), 10) requires $O(k\Delta(\Delta + \log^* m) \log \Delta)$ time using $O(\Delta m)$ processors. On the other hand, since our algorithm computes the $(2\Delta - 1)$-edge coloring just once, the running time of our algorithm reduces to $O(\Delta(\Delta + \log^* m) \log \Delta)$ time using the same number of processors to solve EIMS(k).

5. Conclusion

We have shown that the coloring technique is very useful to devise faster parallel algorithms with fewer numbers of processors for VIMS(k) and EIMS(k), when instances are constant degree graphs. This asserts that the idea of Cole and Vishkin[3] helps to solve these problems drastically faster. Other such cases are known for the maximal independent set problem,[5),7)] the $(\Delta + 1)$-vertex coloring problem,[5),7)] the list ranking problem,[2),3)] the tree contraction problem[1)] and the 5-coloring problem for planar graphs.[6)]

Our approach for EIMS(k) does not seem to work for graphs without any degree constraint, since it uses the edge coloring. However, our approach to VIMS(k) seems to work for graphs which allow NC-vertex coloring algorithms with constant colors, for example, planar graphs,[6)] bipartite graphs, etc.

Acknowledgements The authors would like to thank S. Shiraishi and T. Shoudai for helpful discussions.

References

1) Abrahamson, K., Dadoun, N., Kirkpatrick, D. G. and Przytycka, T.: A Simple Parallel Tree Contraction Algorithm, *J. Algorithms*, Vol. 10, No. 2, pp. 287-302 (1989).

2) Anderson, R. J. and Miller, G. L.: Deterministic Parallel List Ranking, *Algorithmica*, Vol. 6, pp. 859-868 (1991).

3) Cole, R. and Vishkin, U.: Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking, *Inf. Control*, Vol. 70, pp. 32-53 (1986).

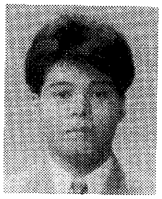
4) Diks, K., Garrido, O. and Lingas, A.: Parallel Algorithms for Finding Maximal k-dependent Sets and Maximal f-matchings, *Proc. 2nd International Symposium on Algorithms, Lecture Notes in Computer Science 557*, pp. 385-395 (1991).

5) Goldberg, A. V., Plotkin, S. A. and Shannon, G. E.: Parallel Symmetry-Breaking in Sparse Graphs, *Proc. 19th ACM STOC*, pp. 315-324 (May 1987).

6) Hagerup, T., Chrobak, M. and Diks, K.: Opti-

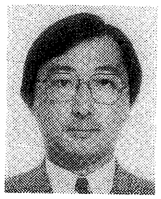
mal Parallel 5-colouring of Planar Graphs, *SIAM J. Comput.*, Vol. 18, pp. 288-300 (1989).
7) Jung, H. and Mehlhorn, K.: Parallel Algorithms for Computing Maximal Independent Sets in Trees and for Updating Minimum Spanning Trees, *Inf. Process. Lett.*, Vol. 27, No. 5, pp. 227-236 (1988).
8) Karp, R. M. and Wigderson, A.: A Fast Parallel Algorithm for the Maximal Independent Set Problem, *J. Assoc. Comput. Mach.*, Vol. 32, pp. 762-773 (1985).
9) Luby, M.: A Simple Parallel Algorithm for the Maximal Independent Set Problem, *SIAM J. Comput.*, Vol. 15, pp. 1036-1053 (1986).
10) Shoudai, T. and Miyano, S.: Bounded Degree Maximal Subgraph Problems are in NC, *Proc. Toyohashi Symposium on Theoretical Computer Science*, pp. 97-101 (1990).
11) Shoudai, T. and Miyano, S.: Using Maximal Independent Sets to Solve Problems in Parallel, *Proc. 17th Int. Workshop on Graph-Theoretic Concepts in Computer Science WG'91, Lecture Notes in Computer Science 570*, pp. 126-134 (1992).

(Received April 6, 1992)
(Accepted October 21, 1992)



Tomoyuki Uchida

Tomoyuki Uchida was born in Kumamoto, Japan on March 3, 1966. He received the B. S. degree in Mathematics and the M. S. degree in Information Systems from Kyushu University, in 1989 and 1991, respectively. He is currently working toward the Dr. Sci. degree at Department of Information Systems, Kyushu University. His research interests include the parallel algorithms and computational complexity.



Satoru Miyano (Member)

Satoru Miyano received the B. S. degree in 1977, the M. S. degree in 1979 and the Dr. Sci. degree in 1984 all in Mathematics from Kyushu University. In 1979, he joined Research Institute of Fundamental Information Science at Kyushu University. Presently, he is an Associate Professor in this institute. He is also an Associate Professor in Department of Information Systems, Interdisciplinary Graduate School at Kyushu University. In 1985-1987, he was a research fellow of Alexander von Humboldt Foundation at Universität Paderborn, Germany. His research interests include parallel algorithms, computational complexity and computational learning theory. Dr. Miyano is a member of Japan Society for Software Science and Technology, Japanese Society for Artificial Intelligence, Mathematical Society of Japan, European Association for Theoretical Computer Science, and IEICE of Japan.