ELSEVIER

Theoretical Computer Science 148 (1995) 57–65

Theoretical Computer Science

Using maximal independent sets to solve problems in parallel

Takayoshi Shoudai[a], Satoru Miyano[b,*]

[a] *Computer Science Laboratory, Faculty of Science, Kyushu University 01, Fukuoka 810, Japan*

[b] *Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan*

Received June 1992; revised August 1994

Communicated by M.S. Paterson

Abstract

We show that the problem of finding a maximal vertex-induced (resp., edge-induced) subgraph of maximum degree k is in NC^2 for $k \geqslant 0$ (resp., $k \geqslant 1$). For these problems, we develop a method which exploits the NC algorithm for the maximal independent set problem. By using the same scheme, the maximal vertex-induced subgraph which satisfies a hereditary graph property π and whose maximum degree is at most Δ can be found in time $O(\Delta^{\lambda(\pi)} T_\pi(n)(\log(n+m))^2)$ using a polynomial number of processors, where $\lambda(\pi)$ is the maximum of diameters of minimal graphs violating π and $T_\pi(n)$ is the time needed to decide whether a graph with n vertices and m edges satisfies π.

1. Introduction

Karp and Wigderson [7], Luby [8], Goldberg and Spencer [4, 5] have shown that the maximal independent set problem, called MIS, is solvable in NC, which is known to be the class of problems computed by PRAMs with a polynomial number of processors in $O((\log n)^k)$ time for some $k \geqslant 0$ [14–16]. In this paper, we show that the problem of finding a maximal subset of vertices whose induced subgraph is of degree at most k allows an NC algorithm for any $k \geqslant 0$. For the proof, we devise a systematic scheme which employs the NC algorithm for MIS [7, 8]. We also show that the problem of finding a maximal set of edges which forms a subgraph of degree at most k is in NC.

In general, a problem of this kind is stated as a maximal subgraph problem for a given property π, which is to find a maximal subset of vertices which induces a subgraph satisfying π. For example, MIS is the problem for property "no two vertices are adjacent". This paper deals with the problem for property "maximum

* Corresponding author.

SSDI 0304-3975(94)00221-5

degree k". A graph property π is called *local* if the diameter of any minimal graph violating π is bounded by some constant. For such local properties π, our scheme can solve the maximal subgraph problem. In particular, if π is local, hereditary and testable in NC, the maximal subgraph problem for "π and maximum degree k" can be solved in NC.

It has been shown that most of the lexicographically first maximal (abbreviated to lfm) subgraph problems are P-complete [12]. Therefore, no NC algorithms exist for the lfm subgraph problems if P $\neq$ NC. On the other hand, the problem of finding *any* maximal subgraph which satisfies a given property seems to allow NC algorithms for many properties. However, only a few are shown to be in NC. As we mentioned above, MIS is one of them. In [12], the maximal edge-induced forest problem and the maximal edge-induced bipartite subgraph problem are shown to be in NC. With some restrictions, the maximal edge-induced outerplanar subgraph problem [10] and the maximal vertex-induced acyclic subgraph problem restricted to directed graphs with degree at most 3 also allow NC algorithms [11]. The results in this paper add a new family of such problems.

2. Preliminaries and definitions

A graph $G = (V, E)$ means an undirected graph without any multiple edges and self-loops. For a subset $U \subseteq V$, we define as $E[U] = \{\{u, v\} \in E \mid u, v \in U\}$. The graph $G[U] = (U, E[U])$ is called the *vertex-induced subgraph of* U. For a subset $F \subseteq E$, we define $V[F]$ to be the set of endpoints of the edges in F. 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, the degree of u is denoted by $deg_G(u)$. We denote by $deg(G) = \max\{deg_G(u) \mid u \in V\}$.

Let $k \geqslant 0$ be any integer. The *maximum degree k vertex-induced subgraph problem* (VIMS(k)) is stated as follows:

VIMS(k)
Instance: A graph $G = (V, E)$.
Problem: Find a maximal subset $U \subseteq V$ such that $G[U]$ is of degree at most k.

In a similar way, the *maximum degree k edge-induced subgraph problem* (EIMS(k)) is defined as follows:

EIMS(k)
Instance: A graph $G = (V, E)$.
Problem: Find a maximal subset $F \subseteq E$ such that $\langle F \rangle$ is of degree at most k.

3. Finding bounded degree maximal subgraphs

Theorem 1. *Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. $VIMS(k)$ for G can be computed in time* $\mathrm{O}(k^2(\log(n+m))^2)$ *with n^2m processors on an EREW PRAM, for* $k \geqslant 0$.

Proof. We show an NC algorithm by employing the NC algorithm for MIS. Let $G = (V, E)$ be a graph for which we are finding a maximal subset U of vertices whose induced subgraph $G[U]$ is of degree at most k.

For subsets W and U of vertices with $W \cap U = \emptyset$, let $E_U^W = \{\{v, w\} \mid v \neq w, v, w \in W$ and there is $u \in U$ such that $\{v, u\} \in E$ and $\{w, u\} \in E\}$. Then let $H_U^W = (W, E[W] \cup E_U^W)$. The required set U of vertices is computed together with a set W of vertices such that $W \cap U = \emptyset$. Initially, let $W = V$ and $U = \emptyset$. At each iteration of the algorithm, a maximal independent set I of H_U^W is computed and added to U while vertices which make the degree of some vertex greater than k are deleted from W together with I. This is iterated k^2 times. Formally the algorithm is described as follows:

1 **begin** /* $G = (V, E)$ is an input */
2 $\quad W \leftarrow V$; $U \leftarrow \emptyset$;
3 $\quad$**for** $i \leftarrow 1$ **to** k^2 **do**
4 $\quad\quad$**begin**
5 $\quad\quad\quad$Find a maximal indepdendent set I of H_U^W;
6 $\quad\quad\quad U \leftarrow U \cup I$;
7 $\quad\quad\quad W \leftarrow W - I$;
8 $\quad\quad\quad W \leftarrow W - \{w \in W \mid deg(G[U \cup \{w\}]) > k\}$
9 $\quad\quad$**end**
10 **end**

We show that this algorithm computes a maximal subset U whose induced subgraph is of degree at most k.

Let $W_0 = V$ and $U_0 = \emptyset$. Then, the graph $H_{U_0}^{W_0}$ is the same as $G = (V, E)$. Therefore, in the first iteration, a maximal independent set of G is computed at line 5. For $i = 1, \ldots, k^2$, let U_i, I_i and W_i be the contents of variables U, I and W at the end of the ith iteration, respectively. Obviously, $W_i \cap U_i = \emptyset$ for $i = 0, \ldots, k^2$. We assume that the induced subgraph $G[U_{i-1}]$ is of degree at most k.

Let $\{w, u\}$ be an edge in E with $w \in W_i$ and $u \in U_i$. Line 8 deletes every vertex which is adjacent to more than k vertices in U_i or adjacent to a vertex v in U_i with $deg_{G[U_i]}(v) = k$. Therefore, u is adjacent to at most k vertices in U_i and $deg_{G[U_i \cup \{w\}]}(u) \leqslant k$. Hence, for each w in W_i, we see that

$$A_i(w) = \sum_{u \in U_i \text{ with } \{w,u\} \in E} deg_{G[U_i \cup \{w\}]}(u) \leqslant k^2.$$

To show that W becomes empty after k^2 iterations, it suffices to prove that each w in W_i satisfies

$$A_i(w) > A_{i-1}(w)$$

for $i = 1, \ldots, k^2$. Since w is not in the maximal independent set I_i of $H_{U_{i-1}}^{W_{i-1}}$ computed by line 5, w is adjacent to a vertex v in $I_i \subseteq W_{i-1}$ via an edge $\{w, v\}$ in $E[W_{i-1}]$ or $E_{U_{i-1}}^{W_{i-1}}$.

Case 1: If $\{w, v\} \in E[W_{i-1}]$, then $\{w, v\}$ is an edge in $G[U_i \cup \{w\}]$. Hence, $deg_{G[U_i \cup \{w\}]}(v) \geq 1$. Since $v \in U_i$, $v \notin U_{i-1}$ and $\{w, v\} \in E$, we see that $A_i(w) \geq A_{i-1}(w) + deg_{G[U_i \cup \{w\}]}(v) > A_{i-1}(w)$.

Case 2: If $\{w, v\} \in E_{U_{i-1}}^{W_{i-1}}$, then there is a vertex $u \in U_{i-1}$ with $\{w, u\} \in E$ and $\{v, u\} \in E$. Since $v \in W_{i-1}$, $W_{i-1} \cap U_{i-1} = \emptyset$ and $w \neq v$, we see $v \notin U_{i-1} \cup \{w\}$. Hence, $\{v, u\}$ is not an edge in $G[U_{i-1} \cup \{w\}]$. On the other hand, v is in U_i and u is in $U_{i-1} \subseteq U_i$. Hence, $\{v, u\}$ is an edge in $G[U_i \cup \{w\}]$. Therefore, $deg_{G[U_i \cup \{w\}]}(u) > deg_{G[U_{i-1} \cup \{w\}]}(u)$. Since $u \in U_i$ and $\{w, u\} \in E$, we see that $A_i(w) > A_{i-1}(w)$.

We now show that $deg(G[U_i]) \leq k$. For a vertex u in U_{i-1}, if u is adjacent to a vertex w in I_i via an edge in E, then no other vertex in I_i is adjacent to u since I_i is also an independent set with respect to $E_{U_{i-1}}^{W_{i-1}}$. Therefore, the degree of u in $G[U_{i-1} \cup I_i]$ remains at most k since $deg(G[U_{i-1} \cup \{w\}]) \leq k$ by the algorithm. For a vertex u in I_i, $deg_{G[U_{i-1} \cup I_i]}(u)$ is at most k since u is adjacent to at most k vertices in U_{i-1} and since I_i is an independent set with respect to $E[W_{i-1}]$. Hence, $deg_{G[U_{i-1} \cup I_i]}(u) \leq k$.

Since only vertices which violate the condition of maximum degree k are deleted from W, the resulting set U is a maximal subset inducing a subgraph of maximum degree k when W becomes empty. Then, this algorithm correctly computes a maximal vertex-induced subgraph of maximum degree k. Since the problem of finding a maximal independent set in a graph is solvable in $O((\log(n + m))^2)$ time using n^2m processors on an EREW PRAM [8] and since the other part of the for-loop can be easily executed using the same amount of time and processors, we can see that VIMS(k) is computed in $O(k^2(\log(n + m))^2)$ time using n^2m processors on an EREW PRAM. □

This theorem also states that the problem can be solved in NC^2 since MIS can be solved in NC^2 [8].

Theorem 2. *Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. EIMS(k) for G can be computed in time $O(k(\log(n + m))^2)$ with n^2m processors on an EREW PRAM, for $k \geq 1$.*

Proof. For this problem, we use maximal matchings instead of maximal independent sets. The algorithm is similar to that in Theorem 1 and repeats the following procedure $2k$ times, where initially $Z = E$ and $F = \emptyset$.

```
1 begin
2    Find a maximal matching M of ⟨Z⟩;
3    F ← F ∪ M;
4    Z ← Z − M;
5    Z ← Z − {e ∈ Z | deg(⟨F ∪ {e}⟩) > k}
6 end
```

Let $Z_0 = E$ and $F_0 = \emptyset$. In the same way as Theorem 1, let F_i, M_i and Z_i be the contents of F, M and Z just after the ith iteration.

For an edge $e = \{u, v\} \in Z_i$,

$$B_i(e) = deg_{\langle F_i \cup \{e\}\rangle}(u) + deg_{\langle F_i \cup \{e\}\rangle}(v) \leqslant 2k$$

holds since all edges making the degree greater than k are deleted from Z by line 5. To see that Z becomes empty after $2k$ iterations, it suffices to show that

$$B_i(e) > B_{i-1}(e)$$

holds for edges in Z_i.

Since e is not in M_i and M_i is a maximal matching in $\langle Z_{i-1}\rangle$, e shares a vertex with some edge e' in M_i. Without loss of generality, we may assume that u is shared by e and e'. Then, $deg_{\langle F_i \cup \{e\}\rangle}(u)$ is greater than $deg_{\langle F_{i-1} \cup \{e\}\rangle}(u)$ since edge e' is not contained in $\langle F_{i-1} \cup \{e\}\rangle$.

It is easy to see that $deg(\langle F_i\rangle) \leqslant k$ since M_i is a matching of $\langle Z_{i-1}\rangle$ and since each edge e in M_i satisfies $deg(\langle F_{i-1} \cup \{e\}\rangle) \leqslant k$.

By the argument above we see that the resulting F is a maximal set of edges such that $deg(\langle F\rangle) \leqslant k$. Since a maximal matching can be found in NC [6, 8], the total algorithm can be implemented in NC. For example, by Luby's parallel MIS algorithm [8], our algorithm can be implemented on an EREW PRAM in time $O(k(\log(n+m))^2)$ using n^2m processors. $\square$

4. Maximal subgraph problem for a local property

Let π be a property on graphs. We say that a graph $G = (V, E)$ is a *minimal graph violating* π with respect to vertices if G violates π and the vertex-induced subgraph $G[U]$ of U satisfies π for every subset U of V with $U \neq V$. The property π is called *local* with respect to vertices if $\lambda(\pi) = \sup\{\text{diameter}(G) \mid G \text{ is a minimal graph violating } \pi \text{ with respect to vertices}\}$ is finite.

Remark 1. A minimal graph violating a property π with respect to vertices must be connected if π is local.

A property π on graphs is called *hereditary* with respect to vertices if for every graph $G = (V, E)$ satisfying π, the vertex-induced subgraph $G[U]$ also satisfies π for every subset $U \subseteq V$.

Theorem 3. *Let π be a graph property which is local and hereditary with respect to vertices. Then a maximal subgraph of a graph $G = (V, E)$ which satisfies π and whose maximum degree is at most Δ can be computed in $O(\Delta^{\lambda(\pi)} T_\pi(n)(\log(n+m))^2)$ time using a polynomial number of processors on an EREW PRAM, where $T_\pi(n)$ is the time needed to decide whether a graph with n vertices and m edges satisfies π.*

Proof. For subsets W and U of vertices with $W \cap U = \emptyset$, let $E_U^W = \{\{v, w\} \subseteq W \mid dist_{G[U \cup \{v,w\}]}(v, w) \leqslant \lambda(\pi) \text{ with } v \neq w\}$ and $N_U(w) = \{u \in U \mid dist_{G[U]}(u, w) \leqslant \lambda(\pi) - 1\}$, where $dist_G(\{v, w\})$ is the length of the shortest path between v and w in G. Then, let $H_U^W = (W, E[W] \cup E_U^W)$. The required set U of vertices is computed together with a set W of vertices such that $W \cap U = \emptyset$. The algorithm is described as follows:

1 **begin** /* $G = (V, E)$ is an input */
2 $\quad W \leftarrow V$; $U \leftarrow \emptyset$;
3 $\quad$ **while** $W \neq \emptyset$ **do**
4 $\quad\quad$ **begin**
5 $\quad\quad\quad$ Find a maximal independent set I of H_U^W;
6 $\quad\quad\quad$ $U \leftarrow U \cup I$;
7 $\quad\quad\quad$ $W \leftarrow W - I$;
8 $\quad\quad\quad$ $W \leftarrow W - \{w \in W \mid G[U \cup \{w\}]$ violates π or $deg(G[U \cup \{w\}]) > \Delta\}$
9 $\quad\quad$ **end**
10 **end**

E_U^W represents tuples of two vertices such that if the vertices are added to U at the same step, an induced subgraph of U may violate the property. We show that this algorithm computes a maximal subset U whose induced subgraph is of degree at most Δ and satisfies π.

Let $W_0 = V$ and $U_0 = \emptyset$. For $i = 1, \ldots, \Delta^{\lambda(\pi)}$, let U_i, I_i and W_i be the contents of variables U, I and W at the end of the ith iteration, respectively. Obviously, $W_i \cap U_i = \emptyset$ for $i = 0, \ldots, \Delta^{\lambda(\pi)}$. We assume that the induced subgraph $G[U_{i-1}]$ satisfies π and the maximum degree of $G[U_{i-1}]$ is at most Δ.

Let $\{w, u\}$ be an edge in E with $w \in W_i$ and $u \in U_i$. Line 8 deletes every vertex which is adjacent to more than Δ vertices in U_i or adjacent to a vertex v in U_i with $deg_{G[U_i]}(v) = \Delta$. Therefore, u is adjacent to at most Δ vertices in U_i and $deg_{G[U_i \cup \{w\}]}(u) \leqslant \Delta$. Moreover, $|N_{U_i}(w)| \leqslant \Delta^{\lambda(\pi)-1}$. Hence, for each w in W_i, we see that

$$A_i(w) = \sum_{u \in N_{U_i}(w)} deg_{G[U_i \cup \{w\}]}(u) \leqslant \Delta^{\lambda(\pi)}.$$

Claim 1. *$A_i(w) > A_{i-1}(w)$ for each w in W_i.*

Proof. Since w is not in the maximal independent set I_i of $H_{U_{i-1}}^{W_{i-1}}$ computed by line 5, w is adjacent to a vertex v in $I_i \subseteq W_{i-1}$ via an edge $\{w, v\}$ in $E[W_{i-1}]$ or $E_{U_{i-1}}^{W_{i-1}}$.

Case 1: $\{w, v\} \in E[W_{i-1}]$. Then $\{w, v\}$ is an edge in $G[U_i \cup \{w\}]$. Hence, $deg_{G[U_i \cup \{w\}]}(v) \geqslant 1$. Since $v \in N_{U_i}(w)$ and $v \notin N_{U_{i-1}}(w)$, we see that $A_i(w) \geqslant A_{i-1}(w) + deg_{G[U_i \cup \{w\}]}(v) > A_{i-1}(w)$.

Case 2: $\{w, v\} \in E_{U_{i-1}}^{W_{i-1}}$. Then there is a path $w, u_1, \ldots, u_{k-1}, v$ with $k \leqslant \lambda(\pi)$ and $u_j \in U_{i-1}$ $(j = 1, \ldots, k-1)$ in $G[U_{i-1} \cup \{w, v\}]$. Since $v \in W_{i-1}$, $W_{i-1} \cap U_{i-1} = \emptyset$ and $w \neq v$, we see $v \notin U_{i-1} \cup \{w\}$. Hence, $\{v, u_{k-1}\}$ is not an edge in $G[U_{i-1} \cup \{w\}]$. On the other hand, v is in U_i and u_{k-1} is in $U_{i-1} \subseteq U_i$. Hence, $\{v, u_{k-1}\}$ is an edge in $G[U_{i-1} \cup \{w\}]$. Therefore, $deg_{G[U_i \cup \{w\}]}(u_{k-1}) > deg_{G[U_{i-1} \cup \{w\}]}(u_{k-1})$. Since $u_{k-1} \in N_{U_{i-1}}(w) \subset N_{U_i}(w)$, we see that $A_i(w) > A_{i-1}(w)$.

Thus, W becomes empty within $\Delta^{\lambda(\pi)}$ iterations of the while-loop. □

Claim 2. *$deg(G[U_i]) \leqslant \Delta$.*

The proof of this claim is similar to the case of Theorem 1.

Claim 3. *$G[U_i]$ satisfies π.*

Proof. We assume that $G[U_i]$ does not satisfy π. Then, there is a minimal subset $S \subseteq U_i$ such that $G[S]$ violates π. Since $S \subseteq U_i$ and $U_i = U_{i-1} \cup I_i$, we see that $S = (S \cap U_{i-1}) \cup (S \cap I_i)$. If $S \cap I_i$ consists of only one vertex v, $U_{i-1} \cup \{v\}$ already violates π. Therefore, there are two distinct vertices v, w such that $\{v, w\} \in E$ or there are at most $\lambda(\pi) - 1$ vertices in $S \cap U_{i-1}$ which construct a path between v and w since diameter $(G[S]) \leqslant \lambda(\pi)$. For each case, $\{v, w\}$ are in $E[W_{i-1}]$ or $E_{U_{i-1}}^{W_{i-1}}$ since $v, w \in I_i \in W_{i-1}$. This contradicts the fact that $v, w \in S \cap I_i \subset I_i$ and I_i is a maximal independent set with respect to $E[W_i] \cup E_{U_{i-1}}^{W_{i-1}}$. Hence, $G[U_i]$ satisfies π. □

Since only vertices which violate the property π or the condition of maximum degree Δ are deleted from W and since π is hereditary, the resulting set U is a maximal subset which induces a subgraph satisfying π when W becomes empty.

MIS can be solved on an EREW PRAM in $O((\log(n + m))^2)$ time using a polynomial number of processors [8]. It is not hard to see that the steps other than MIS can also be implemented on a EREW PRAM in $O((\log(n + m))^2)$ time using a polynomial number of processors. Hence, the total algorithm can be implemented using the same amount of time and processors. □

Remark 2. At line 8 of the algorithm, for each $w \in W$, it is sufficient to decide whether $G[N_U(w) \cup \{w\}]$ satisfies π and $deg(G[N_U(w) \cup \{w\}]) \leqslant \Delta$. Therefore, the time needed to compute line 8 depends only on Δ and $\lambda(\pi)$.

Theorem 3 also states that the problem can be solved in NC^2 for graphs of a constant degree.

Theorem 1 is a special case of Theorem 3 for $\pi =$ "maximum degree k", $\lambda(\pi) = 2$ and $\Delta = k$. For a graph of maximum degree Δ and $\pi =$ "k-cycle free", it takes $O(\Delta^{\lfloor k/2 \rfloor}(\log n)^2)$ time to find a maximal subgraph satisfying π of maximum degree Δ since $\lambda(\pi) = \lfloor k/2 \rfloor$.

5. Concluding remarks

A straightforward method to solve VIMS(k) (resp., EIMS(k)) is to use the polynomial-time greedy sequential algorithm that computes the lfm subset U of vertices (resp., F of edges) such that $deg(G[U])$ (resp., $deg(\langle F \rangle)$) is at most k [2, 12].

Most problems computed by greedy algorithms of this kind are known to be P-complete and therefore hardly efficiently parallelizable [1, 12, 13]. In fact, the lfm maximum degree k vertex-induced subgraph problem is P-complete [12]. However, the situation is different for edge-induced subgraphs.

The class CC is defined to be the class of sets log-space reducible to C-CVP, the comparator circuit value problem [9]. A comparator circuit is a usual circuit such that it contains only comparators C which are gates with two inputs u, v and two outputs uv, $u + v$ and no duplication of the value of an output is allowed.

CC lies as NLOG $\subseteq$ CC $\subseteq$ P [3] and is closed under complement [9]. Currently, CC-complete problems are believed to be neither P-complete nor in NC. Some CC-complete problems are reported in [9]. The lfm matching problem is one of them (stated as a work due to S.A. Cook in [9]). Since a matching is a subgraph of degree at most 1, it is natural to guess that the lfm maximum degree k *edge-induced* subgraph problem, denoted LF-EIMS(k), is also CC-complete for all $k \geqslant 1$. We can show that this is the case. Since the proof technique is the same as that for the lfm matching problem, we omit the proof.

Theorem 4. LF-EIMS(k) *is CC-complete for* $k \geqslant 1$.

We have shown that parallel MIS algorithms are useful to solve the maximal subgraph problem for a property "local and of degree at most Δ". However, the idea of using MIS does not seem to work for other properties, for example, "acyclic", "planar", which are not local. MIS locates at an interesting position in the NC hierarchy. It is in NC^2 but unlikely to belong to classes such as AC^1 and DET shown in [2]. It is not difficult to see that the algorithms shown in this paper can be transformed by NC^1-reductions to MIS. Hence, the results in this paper give some new problems NC^1-reducible to MIS.

References

[1] R. Anderson and E.W. Mayr, Parallelism and the maximal path problem, *Inform. Process. Lett.* **24** (1987) 121–126.

[2] S.A. Cook, A taxonomy of problems with fast parallel algorithms, *Inform. and Control* **64** (1985) 2–22.

[3] T. Feder, A new fixed point approach for stable networks and stable marriages, *J. Comput. System Sci.* **45** (1992) 233–284.

[4] M. Goldberg and T. Spencer, A new parallel algorithm for the maximal independent set problem, *SIAM J. Comput.* **18** (1989) 419–427.

[5] M. Goldberg and T. Spencer, Constructing a maximal independent set in parallel, *SIAM J. Discrete Math.* **2** (1989) 322–328.

[6] A. Israeli and Y. Shiloach, An improved parallel algorithm for maximal matching, *Inform. Process. Lett.* **22** (1986) 57–60.

[7] R.M. Karp and A. Wigderson, A fast parallel algorithm for the maximal independent set problem, *J. Assoc. Comput. Mach.* **32** (1985) 762–773.

[8] M. Luby, A simple parallel algorithm for the maximal independent set problem, *SIAM J. Comput.* **15** (1986) 1036–1053.

[9] E.W. Mayr and A. Subramanian, The complexity of circuit value and network stability, *J. Comput. System Sci.* **44** (1992) 302–323.

[10] S. Miyano, A parallelizable lexicographically first maximal subgraph problem, *Inform. Process. Lett.* **29** (1988) 75–78.

[11] S. Miyano, Δ_2^p-complete lexicographically first maximal subgraph problems, *Theoret. Comput. Sci.* **88** (1991) 33–57.

[12] S. Miyano, The lexicographically first maximal subgraph problems: P-completeness and NC algorithms, *Math. Systems Theory* **22** (1989) 47–73.

[13] S. Miyano, S. Shiraishi and T. Shoudai, A list of P-complete problems, RIFIS-TR-CS-17, Research Institute of Fundamental Information science, Kyushu University, 1989 (revised in December, 1990).

[14] N. Pippenger, On simultaneous resource bounds, in: *Proc. 20th IEEE FOCS* (1979) 307–311.

[15] W.L. Ruzzo, On uniform circuit complexity, *J. Comput. System Sci.* **22** (1981) 365–383.

[16] L. Stockmeyer and U. Vishkin, Simulation of parallel random access machines by circuits, *SIAM J. Comput.* **13** (1984) 409–422.