A Parallel Algorithm for the Maximal Co-Hitting Set Problem

Shoudai, Takayoshi
Faculty of Engineering, Yamaguchi University

Miyano, Satoru
Research Institute of Fundamental Information Science Kyushu University

http://hdl.handle.net/2324/3167

出版情報：RIFIS Technical Report. 65, 1992-10-30. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-65

RIFIS Technical Report

A Parallel Algorithm for the Maximal Co-Hitting Set Problem

Takayosi Shoudai
Satoru Miyano

October 30, 1992

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

E-mail: miyano@rifis.sci.kyushu-u.ac.jp Phone: 092-641-1101 ex. 4471

A Parallel Algorithm for the Maximal Co-Hitting Set Problem

Takayoshi Shoudai
Faculty of Engineering,
Yamaguchi University, Ube-shi, 755, Japan

Satoru Miyano
Research Institute of Fundamental Information Science
Kyushu University 33, Fukuoka-shi, 812, Japan

Abstract

Let $C = \{c_1, \ldots, c_m\}$ be a family of subsets of a finite set $S = \{1, \ldots, n\}$, a subset S' of S is a co-hitting set if S' contains no element of C as a subset. By using an $O((\log n)^2)$ time EREW PRAM algorithm for a maximal independent set problem (MIS), we show that a maximal co-hitting set for S can be computed on an EREW PRAM in time $O(\alpha\beta(\log(n+m))^2)$ using $O(n^2 m)$ processors, where $\alpha = \max\{|c_i| \mid i = 1, \ldots, m\}$ and $\beta = \max\{|d_j| \mid j = 1, \ldots, n\}$ with $d_j = \{c_i \mid j \in c_i\}$. This implies that if $\alpha\beta = O((\log(n+m))^k)$ then the problem is solvable in NC.

Keywords: algorithm and computational complexity, parallel algorithm, maximal co-hitting set problem, minimal set cover problem, maximal independent set problem

1 Introduction

Let $C = \{c_1, \ldots, c_m\}$ be a family of subsets of a finite set $S = \{1, \ldots, n\}$. A subset S' of S is called a co-hitting set for C if $c_i \not\subseteq S'$ for all $i = 1, \ldots, m$. The maximal co-hitting set problem (MCHS) is to find a maximal co-hitting set when a finite set S and a family C of subsets of S are given. We study a way of employing the parallel algorithms for the maximal independent set problem (MIS)[4, 5, 6] to solve maximal or minimal problems in NC, which is the class of problems computable by PRAMs with a polynomial number of processors in $O((\log n)^k)$ time for some $k \geq 0$. For the bounded degree maximal subgraph problems, by using the NC algorithms for MIS, we have constructed an NC algorithm[8]. Moreover, we have shown that MIS is useful to solve some class of the maximal vertex-induced subgraph problem for a hereditary and local graph property in parallel[9]. This paper presents a parallel algorithm for solving MCHS by employing the same technique as the previous results.

The maximal independent set problem for an undirected graph $G = (V, E)$ is to find a maximal subset of vertices U such that no two vertices in U are adjacent. It is easy to see that MIS is a special case of MCHS for $|c_i| = 2$ for all $i = 1, \ldots, m$. These problems are easily solved in polynomial time by straightforward greedy sequential algorithms. However, these algorithms are hardly parallelizable since they are P-complete[2, 6, 7]. On the other hand, the problem of finding *any* maximal independent set of a graph was shown to be in NC[4, 5, 6]. Our algorithm finds *any* maximal co-hitting set for a given family of subsets by using an NC maximal independent set algorithm. Berger et al.[1] showed an NC approximation algorithm for the minimum set cover problem, which is known to be NP-complete[3]. Since the minimum set cover problem is easily reduced from MCHS as stated in the next section, their algorithm also computes a maximum co-hitting set approximately. But their algorithm does not necessarily produce a maximal co-hitting set. It is not known whether *any* maximal co-hitting set is computed in NC.

In this paper, we show an algorithm for the maximal co-hitting set problem which employs the MIS algorithm. For example, by using the Luby's parallel MIS algorithm[6] as subroutines, our algorithm can be implemented on an EREW PRAM in time $O(\alpha\beta(\log(n+m))^2)$ using $O(n^2m)$ processors, where $\alpha = \max\{|c_i| \mid i = 1, \ldots, m\}$ and $\beta = \max\{|d_j| \mid j = 1, \ldots, n\}$ with $d_j = \{c_i \mid j \in c_i\}$. This implies that if $\alpha\beta = O((\log(n+m))^k)$ then the problem is solvable in NC.

2 Solving the maximal co-hitting set problem using MIS

Let $C = \{c_1, ..., c_m\}$ be a family of subsets of a finite set $S = \{1, ..., n\}$. A subset S' of S is called a *co-hitting set* if $c_i \not\subseteq S'$ for all $i = 1, ..., m$.

Theorem 1 Let $C = \{c_1, ..., c_m\}$ be a family of distinct subsets of a finite set $S = \{1, ..., n\}$. Let $\alpha = \max\{|c_i| \mid i = 1, ..., m\}$ and $\beta = \max\{|d_j| \mid j = 1, ..., n\}$, where $d_j = \{c_i \mid j \in c_i\}$. Then a maximal co-hitting set for C can be computed on an EREW PRAM in time $O(\alpha\beta(\log(n+m))^2)$ using $O(n^2m)$ processors.

Hence, if $\alpha\beta = O((\log(n+m))^k)$, then a minimal co-hitting set can be computed in NC.

```
/* A family C0 = {c1, ..., cm} with ci ⊆ S0 = {1, ..., n} for i = 1, ..., m is given. */
/* We assume that S0 = ⋃_{c∈C0} c and |ci| ≥ 2 for i = 1, ..., m. */
1  begin
2      S ← S0; C ← C0;
3      W ← ∅; /* W gets a maximal co-hitting set */
4      while S ≠ ∅ do
5          begin
6              E ← ∅;
7              par c ∈ C do
8                  begin
9                      Choose two distinct vertices v, w from c ∩ S;
10                     Add the edge {v, w} to E
11                 end;
12             Find a maximal independent set I of the graph G = (S, E);
13             W ← W ∪ I;
14             S ← S − I;
15             U ← {u ∈ S | c ∩ S ⊆ W ∪ {u} for some c ∈ C};
16             par c ∈ C do if c ∩ U ≠ ∅ then delete c from C;
17             S ← S − U;
18             V ← S − ⋃_{c∈C} c
19             W ← W ∪ V
20             S ← S − V;
21         end
22 end
```

Figure 1: Algorithm MCHS

Proof. We consider the algorithm MCHS (Figure. 1) that finds a maximal co-hitting set for C_0.

The variable W gets a maximal co-hitting set. Let I_i, C_i, U_i, W_i and S_i be the contents of the variables I, C, U, W and S just after the ith iteration of the while-loop, respectively. For convenience, let $W_0 = \emptyset$ and $U_0 = \emptyset$. Let $U_i^* = U_0 \cup \cdots \cup U_i$. We also let E_i be the set of edges constructed during lines 7-11. Then from the algorithm we can easily see that S_0, S_{i-1} and W_i are represented as the following disjoint unions (Figure 2):

(1) $S_i \cup W_i \cup U_i^* = S_0$.

(2) $S_{i-1} = I_i \cup U_i \cup V_i \cup S_i$.

(3) $W_i = W_{i-1} \cup I_i \cup V_i$.

Claim 1. For $c \in C_i$, $c \cap S_i$ contains at least two elements.

Proof. By the assumption on the input, Claim 1 obviously holds for $i = 0$. Assume that the claim holds for i and $S_{i+1} \neq \emptyset$. Let c be in C_i. Then $c \cap U_i = \emptyset$ from line 16 and $c \cap V_i = \emptyset$ from line 18. Therefore from (2) we see that $c \cap S_i = c \cap (S_{i-1} - I_i)$. If $c \cap S_i = \emptyset$, then $U_i = S_{i-1} - I_i$ from line 15. This yields $S_i = \emptyset$ from line 17. This is a contradiction since S_i is assumed not empty. On the other hand, if $c \cap S_i = \{u\}$, then $c \cap S_i \subseteq W_{i-1} \cup I_i \cup \{u\}$. This means that u is in U_i and, therefore, $c \cap U_i \neq \emptyset$, a contradiction. Thus $|c \cap S_i| \geq 2$.

Claim 2. W_i is a co-hitting set for C_0.

Proof. We assume that $S_{i-1} \neq \emptyset$. Obviously, $W_0 = \emptyset$ is a co-hitting set for C_0. Assume that W_{i-1} is a co-hitting set for C_0. Let c be in C_0.

Case 1. $c \notin C_i$: c was deleted during the jth iteration for some $1 \leq j \leq i$. Then $c \cap U_j \neq \emptyset$. Hence there is u in $c \cap U_j \subseteq U_i^*$. By (1) u is not in W_i. Therefore we have $c \not\subseteq W_i$.

Case 2. $c \in C_i$: c is also in C_{i-1}. Then by Claim 1 there are v, w in $c \cap S_{i-1}$ with $v \neq w$ and $\{v, w\} \in E_i$. Since I_i is an independent set, $v \notin I_i$ or $w \notin I_i$. Since W_{i-1} is a co-hitting set for C_0, we have $c \not\subseteq W_{i-1}$. Since no element in S_{i-1}, hence no element in I_i, is in W_{i-1}, v or w is not in $W_{i-1} \cup I_i$. Therefore $c \not\subseteq W_{i-1} \cup I_i$. On the other hand, $c \cap V_i = \emptyset$ by line 18. Therefore $c \not\subseteq W_{i-1} \cup I_i \cup V_i = W_i$.

Claim 3. For any $u \in U_i$, there is $c \in C_{i-1}$ such that $c \subseteq W_i$.

Proof. By line 15, for $u \in U_i$ there is $c \in C_{i-1}$ such that $c \cap (S_{i-1} - I_i) \subseteq W_{i-1} \cup I_i \cup \{u\}$. Then $c \cap S_{i-1} \subseteq W_{i-1} \cup I_i \cup \{u\}$. Note that for $c \in C_{i-1}$ we have $c \cap U_{i-1}^* = \emptyset$ by line 16. Then

$$
\begin{aligned}
c &= c \cap (S_{i-1} \cup W_{i-1} \cup U_{i-1}^* && \text{(by (1))} \\
&= (c \cap S_{i-1}) \cup (c \cap W_{i-1}) \cup (c \cap U_{i-1}^*) && \\
&\subseteq W_{i-1} \cup I_i \cup \{u\}. && (\text{by } c \cap U_{i-1}^* = \emptyset)
\end{aligned}
$$

Let t be the integer such that $S_t = \emptyset$. Then by (1) $S_0 = W_t \cup U_t^*$. From Claim 2 W_t is a co-hitting set. Claim 3 asserts that for any $u \in U_t^*$ there is some c with $c \subseteq W_t \cup \{u\}$. Therefore W_t is a maximal co-hitting set for C_0.

Claim 4. $t \leq \alpha\beta$.

Proof. For $u \in S_i$, we define

$$B_i(u) = \{v \mid u \neq v \text{ and } \{u, v\} \subseteq c \cap S_i \text{ for some } c \in C_i\}.$$

It is easy to see that $|B_i(u)| \leq \alpha\beta$. Then it suffices to show that

$$|B_i(u)| < |B_{i-1}(u)|$$

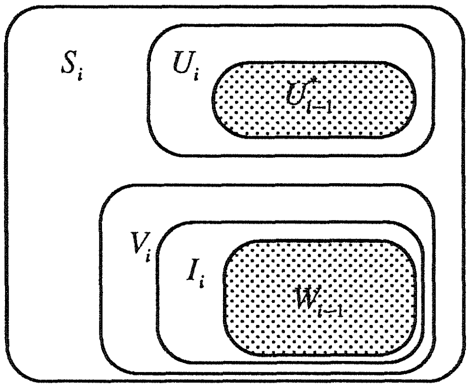

Figure 2: Relation between S_i, I_i, V_i, W_{i-1} and U_i^*

for each $u \in S_i$. If $u \in S_i$, then u is not in I_i from line 14. Since I_i is a maximal independent set, there is v with $\{u,v\} \in E_i$. Therefore $\{u,v\} \subseteq c \cap S_{i-1}$ for some $c \in C_{i-1}$. Hence v is in $B_{i-1}(u)$. However, v is not in S_i since v is in I_i. Therefore v is not in $B_i(u)$.

As in the proof of Theorem 1, the part of finding a maximal independent set can be implemented on an EREW PRAM in $O((\log(n+m))^2))$ time using $O(n^2m)$ processors[6]. The other steps can also be implemented with at most the same amount of time and processors.□

A subset S' of S is called a *hitting set* for C if $c_i \cap S' \neq \emptyset$ for all $i = 1, ..., m$. We say that C is a *set cover* if $\bigcup_{i=1}^{m} c_i = S$.

It should be noticed that S' is a hitting set for C and only if $S - S'$ is a co-hitting set for C. Therefore, S' is a minimal hitting set for C if and only if $S - S'$ is a maximal co-hitting set for C.

The problem of finding a hitting set is closely related to the set cover problem. For a family $C = \{c_1, ..., c_m\}$ with $\bigcup_{i=1}^{n} c_i = \{1, ..., n\}$, let

$$d_j = \{c_i \mid j \in c_i \in C\}$$

for $j = 1, ..., n$. Then each d_j is not empty. Let $D = \{d_1, ..., d_n\}$ and $C' \subseteq C$ be a minimal hitting set for D. Then $d_j \cap C' \neq \emptyset$ for each $j = 1, ..., n$. Therefore there is some $c_i \in d_j \cap C'$. Thus $j \in c_i$. Hence C' is a set cover of $\{1, ..., n\}$ and also can be seen that C' is minimal. The following corollary is obtained in a straightforward way from Theorem 1:

Corollary 1 Let $C = \{c_1, ..., c_m\}$ be a family of subsets of a finite set $S = \{1, ..., n\}$ such that $S = \bigcup_{i=1}^{m} c_i$. Let $\alpha = \max\{|c_i| \mid i = 1, ..., m\}$ and $\beta = \max\{|d_j| \mid j = 1, ..., n\}$, where

$d_j = \{c_i \mid j \in c_i\}$. Then a minimal set cover for S can be computed on an EREW PRAM in time $O(\alpha\beta(\log(n+m))^2)$ using $O(n^2m)$ processors. Hence, if $\alpha\beta = O((\log(n+m))^k)$, then a minimal set cover can be computed in NC.

3 Conclusion

We have shown that parallel MIS algorithms are useful to solve the maximal co-hitting set problem. However, the idea of using MIS does not seem to work unless α and β can be bounded. 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 [2]. It is not difficult to see that the algorithms shown in this paper can be transformed to NC^1-reductions to MIS. Hence the result in this paper give some new problems NC^1-reducible to MIS.

References

[1] Berger, B., Rompel, J. and Shor, P.W., Efficient NC algorithm for set cover with application to learning and geometry, 30th IEEE FOCS, **30**, pp. 54-59 (1989).

[2] Cook, S.A., A taxonomy of problems with fast parallel algorithms, Inf. Contr., **64**, pp. 2-22 (1985).

[3] Garey, M.R. and Johnson, D.S., Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, (1979).

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

[5] Karp, R.M. and Wigderson, A., A fast parallel algorithm for the maximal independent set problem, J. ACM, **32**, pp. 762-773 (1985).

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

[7] Miyano, S., Shiraishi, S. and Shoudai, T., A list of P-complete problems, RIFIS-TR-CS-17, Kyushu University (1989) (revised and extended in December, 1990).

[8] Shoudai, T. and Miyano, S., Bounded degree maximal subgraph problems are in NC, Proc. Toyohashi Symposium on Theoretical Computer Science, pp. 97-101 (1990).

[9] Shoudai, T. and Miyano, S., Using maximal independent sets to solve problems in parallel, Proc. 17th Intern. Workshop on Graph-Theoretic Concepts in Computer Science WG91, LNCS, **570**, pp. 126-134 (1992).