AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **33** (2005), Pages 3–14

Parallel algorithms for generalized clique transversal problems

Elias Dahlhaus

Institut für Informatik
Universität Bonn
Germany

Paul D. Manuel

Department of Information Science
Kuwait University
Kuwait 13060

Mirka Miller

School of ITMS
University of Ballarat
Ballarat, Vic. 3353
Australia

Abstract

For a hypergraph $H(V, \bar{E})$, let $\Im = \{S_1, S_2, ..., S_r\}$ be a family of subsets of V such that each S_i is a subset of some hyperedge of $\bar{E}$. A $\Im$-**transversal problem** is to find a minimum subfamily $\Im'$ of $\Im$ such that a hyperedge of H contains a member of $\Im'$ whenever it contains a member of $\Im$. This problem reduces to the transversal problem when $\Im = V$ and each S_i is a singleton set consisting of a vertex of V. The K_l-clique transversal problem becomes a particular case of $\Im$-transversal problem when hyperedges are the maximal cliques and $\Im$ is the family of all cliques of size l. We give an NC-algorithm to solve $\Im$-transversal problem on totally balanced hypergraphs. The main result of this paper is that the K_l-clique transversal on strongly chordal graphs is solvable in polylogarithmic time with polynomial number of processors.

1 Introduction

A $0-1$ matrix is **balanced** if it does not contain as a submatrix, an edge-vertex incidence matrix of an odd cycle. A $0-1$ matrix is **totally balanced** if it does not contain as a submatrix, an edge-vertex incidence matrix of any cycle.

A $0-1$ matrix is called **Γ-free** if it does not contain the submatrix

$$\Gamma = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

It is known that a matrix is totally balanced if and only if it can be permuted to a Γ-free matrix [13].

A **hypergraph** H is an ordered pair $(V, \bar{E})$ where V is a set of vertices and $\bar{E}$ is a family of subsets of V. The members of $\bar{E}$ are called hyperedges of H. Let $V = \{v_1, v_2, ..., v_n\}$ and $\bar{E} = \{E_1, E_2, ..., E_m\}$. Let $A(H)$ denote the hyperedge-vertex incidence matrix of a hypergraph H. A hypergraph H is **balanced** (respectively **totally balanced**) if $A(H)$ is balanced (respectively totally balanced). A clique hypergraph $H(V, \bar{E})$ of a graph $G(V, E)$ is a hypergraph whose hyperedges are the maximal cliques of G. A graph is said to be **balanced** if its clique hypergraph is balanced. A graph is said to be **chordal** if it contains no induced cycle of length 4 or greater. A chordal graph is said to be **strongly chordal** if every cycle on six or more vertices contains a chord joining two vertices with an odd distance between them. It is known that a chordal graph is strongly chordal if its clique hypergraph is totally balanced [6, 9].

A k-**fold clique transversal** of a graph G is a set S of vertices such that every maximal clique of G has at least k vertices of S. This problem has been shown to be in the polynomial class for balanced graphs [9]. Corneil and Fonlupt [5] have introduced the C_{ij}-cover problem which is to find a minimum family of cliques of size j such that every clique of size i of G contains at least one member of the family. The C_{ij}-cover problem has been studied in [3, 4, 5, 14] . We study a similar concept called the K_l-clique transversal problem. It is usual to denote a clique of size r by K_r. A K_l**-clique transversal** of a graph G is a collection of cliques of size l such that every maximal clique of size greater than or equal to l in G contains at least one member of the collection. A K_l**-clique transversal problem** is to locate a K_l-clique transversal with the minimum cardinality. A problem similar to this is the k-fold clique transversal problem which is to determine the minimum cardinality of a subset D of V such that $D \cap C_i$ has at least l vertices for every maximal clique C_i of G. This problem has been shown to be polynomial for balanced graphs [9]. Chang et al [2] call this "generalized clique transversal problem" and have given complexity results for strongly chordal graphs, k-trees, split graphs and path graphs. A clique transversal is a K_1-clique transversal and it has been widely studied [1, 6, 17, 18].

Note that the K_l-clique transversal problem is different from the l-fold clique transversal problem. In Figure 1, $K_2^1 = \{1, 2\}$, $K_2^2 = \{4, 5\}$ is a minimum K_2-clique transversal of the graph. But $\{1, 2, 4, 5\}$ is not a minimum 2-fold clique transversal whereas $\{1, 3, 4\}$ is a minimum 2-fold clique transversal of the graph. It is interesting

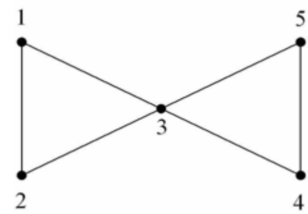

Figure 1: A minimum K_2-clique transversal does not necessarily give a minimum 2-fold clique transversal

to explore whether a minimum K_l-clique transversal can be efficiently extracted from a minimum l-fold clique transversal of a graph.

The notion of a K_l-clique transversal can be extended to hypergraphs as follows: For a hypergraph $H(V, \bar{E})$, let $\Im = \{S_1, S_2, ..., S_r\}$ be a family of subsets of V such that each S_i is a subset of some hyperedge of $\bar{E}$. A $\Im$-**transversal problem** is to find a minimum subfamily $\Im'$ of $\Im$ such that a hyperedge of H contains a member of $\Im'$ whenever it contains a member of $\Im$. This problem reduces to the transversal problem when $\Im = V$ and each S_i is a singleton set consisting of a vertex of V. The K_l-clique transversal problem becomes a particular case of $\Im$-transversal problem when hyperedges are the maximal cliques and $\Im$ is the family of all cliques of size l.

We will later use the following result, due to Dahlhaus and Damaschke [7]:

Theorem 1 *The transversalproblem can be solved on totally balanced hyoergraphs in $O(\log^3 n)$ time with $O(n+m)$ processors where n is the number of vertices and hyperedges and m is the sum of the sizes of the hyperedges.*

This result can be stated as

Theorem 2 *Consider the integer programming problem*

$$\left.\begin{array}{lll} \textit{minimize} & & \sum_{j=1}^{n} x_j \\ \textit{subject} & \textit{to} & M\overline{x} \geq \widehat{1} \end{array}\right\} \tag{1}$$

where M is a totally balanced matrix and $\overline{x} = (x_1, x_2, \ldots, x_n)$ is such that $x_i = 0$ or 1. Here $\widehat{1}$ stands for the all-one vector and vectors will be considered columnwise. The integer programming problem (1) can be solved in $O(log^3 n)$ time with $O(n+m)$ processors. □

In this paper, we study the K_l-clique transversal and $\Im$-transversal problems. We give an NC-algorithm to solve $\Im$-transversal problem on totally balanced hypergraphs using Theorem 2. The main result of this paper is that the K_l-clique transversal on strongly chordal graphs is solvable in polylogarithmic time with polynomial number of processors.

	v_1	v_2	$\dots$	v_n
E_1	$a(1,1)$	$a(1,2)$	$\dots$	$a(1,n)$
E_2	$a(2,1)$	$a(2,2)$	$\dots$	$a(2,n)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
E_m	$a(m,1)$	$a(m,2)$	$\dots$	$a(m,n)$

Matrix A(H)

	S_1	S_2	$\dots$	S_r
E_1	$b(1,1)$	$b(1,2)$	$\dots$	$b(1,r)$
E_2	$b(2,1)$	$b(2,2)$	$\dots$	$b(2,r)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
E_m	$b(m,1)$	$b(m,2)$	$\dots$	$b(m,r)$

Matrix B(H,F)

Figure 2: Figure 2(a) is matrix $A(H)$. Figure 2(b) is matrix $B(H,\Im)$. If $A(H)$ is Γ-free, then $B(H,\Im)$ is Γ-free

2 $\Im-$ transversal problems

Given $H(V,\bar{E})$ and a family $\Im = \{S_1, S_2, \dots, S_r\}$ of subsets of V, consider the incidence matrix $B(H,\Im) = (b(i,j))$ with m rows representing the edges $E_1, E_2, ..., E_m$ of $\bar{E}$ and r columns representing the sets $S_1, S_2, \dots, S_r$ with

$$b(i,j) = \begin{cases} 1 & if \quad S_i \subset E_j \\ 0 & if \quad S_i \not\subseteq E_j \end{cases}$$

The $\Im$- transversal problem can be formulated as:

$$\left.\begin{array}{lcr} minimize & & \sum_{j=1}^{r} x_j \\ & & \\ subject & to & B\overline{x} \geq \widehat{1} \end{array}\right\} \qquad (2)$$

where $B = B(H,\Im)$ is the incidence matrix and $\overline{x} = (x_1, x_2, \dots, x_r)$ is such that $x_i = 0$ or 1. Here again, $\widehat{1}$ stands for the all-one vector and vectors will be considered columnwise.

In this section, we show that the $\Im$–transversal problem is in NC-class for totally balanced hypergraphs. It is enough to prove that $B(H,\Im)$ is totally balanced whenever $H(V,\bar{E})$ is totally balanced.

Let $H(V,E)$ be a hypergraph with the vertex set $V = \{v_1, v_2, \dots, v_n\}$. Let $\Im = \{S_1, S_2, \dots, S_r\}$ be a family of ordered subsets of V such that the vertices of S_i, $i = 1, 2, \dots, r$, are sorted by its indices. For example, (v_2, v_5, v_7, v_8) is sorted, whereas (v_2, v_7, v_5, v_8) is not. The ordered sets S_i are sorted by lexicographic ordering ' $<$ '. The lexicographic ordering on S_i can be defined as follows: Let $(v_{\alpha_1}, v_{\alpha_2}, ..., v_{\alpha_r})$ and $(v_{\beta_1}, v_{\beta_2}, ..., v_{\beta_s})$ be such that $\alpha_1 < \alpha_2 < ... < \alpha_r$ and $\beta_1 < \beta_2 < ... < \beta_s$. We say $(v_{\alpha_1}, v_{\alpha_2}, ..., v_{\alpha_r}) < (v_{\beta_1}, v_{\beta_2}, ..., v_{\beta_s})$ if and only if $(\alpha_1, \alpha_2, ..., \alpha_r) < (\beta_1, \beta_2, ..., \beta_s)$ under the standard dictionary ordering. For example, let $S_1 = (v_3, v_5, v_9, v_{10})$, $S_2 = (v_3, v_5, v_{10}, v_{11})$ and $S_3 = (v_3, v_5, v_{10})$. Then $S_1 < S_2$, $S_1 < S_3$ and $S_3 < S_2$ by lexicographic ordering.

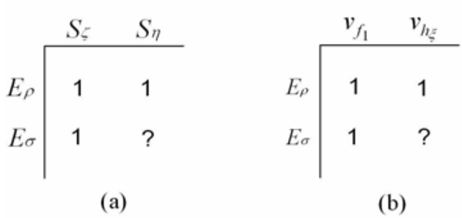

Figure 3: Submatrices of (a) matrix $B(H,\Im)$ and (b) matrix $A(H)$

Lemma 1 *Let $A(H) = (a(i,j))$ be the hyperedge-vertex incidence matrix of a hypergraph $H(V,E)$ with n columns representing the vertices in the order $v_1, v_2, \ldots, v_n$. See Figure 2. If $A(H)$ is Γ-free, then $B(H,\Im) = (b(i,j))$ is Γ-free where the r columns of $B(H,\Im)$ are represented by the sets $S_1, S_2, \ldots, S_r$ sorted lexicographically so that $S_1 < S_2 < \ldots < S_r$.*

Proof. It is given that the vertices of S_i, $i = 1, 2, \ldots, r$, are sorted by its indices and the sets $S_1, S_2, \ldots, S_r$ are sorted by lexicographic ordering such that $S_1 < S_2 < \ldots < S_r$. We will prove that $B(H,\Im)$ is Γ-free with r columns representing the sets in the order $S_1, S_2, \ldots, S_r$. If $b(\rho,\zeta) = b(\rho,\eta) = b(\sigma,\zeta) = 1$, it is enough to prove that $b(\sigma,\eta) = 1$. See Figure 3(a).

Let $S_\zeta = (v_{f_1}, v_{f_2}, \ldots, v_{f_\lambda})$, $S_\eta = (v_{h_1}, v_{h_2}, \ldots, v_{h_\mu})$ such that $f_1 < f_2 < \ldots < f_\lambda$ and $h_1 < h_2 < \ldots < h_\mu$. It is enough to prove that $S_\eta \subset E_\sigma$. Suppose $S_\eta \not\subseteq E_\sigma$. Then there exists v_{h_ξ} such that $v_{h_\xi} \in S_\eta$ and $v_{h_\xi} \notin E_\sigma$. Now f_1 and h_ξ must be distinct because $v_{f_1} \in E_\sigma$ and $v_{h_\xi} \notin E_\sigma$. Since $S_\zeta < S_\eta$, $f_1 < h_\xi$. Now we consider the columns f_1 and h_ξ, and rows ρ and σ in $A(H)$. The vertex $v_{f_1} \in E_\rho$ since $S_\zeta \subset E_\rho$. The vertex $v_{h_\xi} \in E_\rho$ since $S_\eta \subset E_\rho$ and $v_{f_1} \in E_\sigma$ since $S_\zeta \subset E_\sigma$. Therefore, $a(\rho, f_1) = a(\rho, h_\xi) = a(\sigma, f_1) = 1$. See Figure 3(b). Now since $v_{h_\xi} \notin E_\sigma$, $a(\sigma, h_\xi) = 0$. This shows that $A(H)$ is not Γ-free which is a contradiction. ■

It is known [13] that a 0-1 matrix is totally balanced if and only if it can be permuted to a Γ-free matrix. Thus, it follows from Lemma 1

Theorem 3 *If $H(V,\bar{E})$ is a totally balanced hypergraph, then $B(H,\Im)$ is totally balanced where $\Im = \{S_1, S_2, \ldots, S_r\}$ is a family of subsets of V.* □

Using Theorem 2 and Theorem 3, we conclude from (2) that

Theorem 4 *The $\Im$–transversal problem is in NC-class for totally balanced hypergraphs.* □

It would be worthwhile to investigate whether the same method can be extended to balanced hypergraphs.

3 K_l-clique transversal problem

The K_l-clique transversal problem becomes a particular case of $\Im$–transversal problem when $\Im$ is the family of all cliques of size l. Since the number of maximal cliques of a chordal graph is $O(n)$, the number of cliques size l of a chordal graph is $O(n^{l+1})$. Thus the K_l-clique transversal problem is polynomially solvable for strongly chordal graphs by Theorem 4 if l is fixed. The running time of this algorithm is exponential in l. In this section, we give a simple sequential algorithm and an NC-algorithm to solve the K_l-clique transversal problem on strongly chordal graphs.

3.1 A sequential top-down scan

Let $G(V, E)$ be a strongly chordal graph and $D(G) = (d(i, j))$ its **clique matrix** with p rows representing the maximal cliques $C_1, C_2, ..., C_p$ of G and n columns representing the vertices $v_1, v_2, ..., v_n$ of G with

$$d(i,j) = \begin{cases} 1 & if \quad v_i \in C_j \\ 0 & if \quad v_i \notin C_j \end{cases}$$

Let us recall the result of [13] that a 0-1 matrix is totally balanced if and only if it can be permuted to a Γ-free matrix. Therefore $D(G)$ admits a Γ-free matrix. Hence without loss generality we assume that $D(G)$ is Γ-free. Here onwards, $D(G)$ is denoted by D. To compute a minimum K_l-clique transversal of a strongly chordal graph, the algorithm scans the clique matrix row by row from the top row to the bottom row. That is, the algorithm scans rows in increasing order and in each row it scans the nonzero entries from right to left. Let Y_i denote the l rightmost columns which have nonzero entries in row i of D. Note that Y_i is clique of size l in C_i. Say $\Re$ denotes the minimum K_l-clique transversal of G computed by the following algorithm and $\Re_i$ denotes the partial answer set constructed by the algorithm at row i. Here is the algorithm:

Algorithm 1

1. Let $\Re_0 = \phi$.
2. Add Y_1 to $\Re_1$. Thus now $\Re_1 = \{Y_1\}$.
3. For $i = 2$ to p, do the following: If C_i contains a member of $\Re_{i-1}$, let $\Re_i = \Re_{i-1}$; otherwise let $\Re_i = \Re_{i-1} \cup \{Y_i\}$.
4. Let $\Re = \Re_p$.

Now we show that $\Re$, the set constructed by the algorithm, is a minimum K_l-clique transversal of G. We need the following observations to prove the correctness of the algorithm.

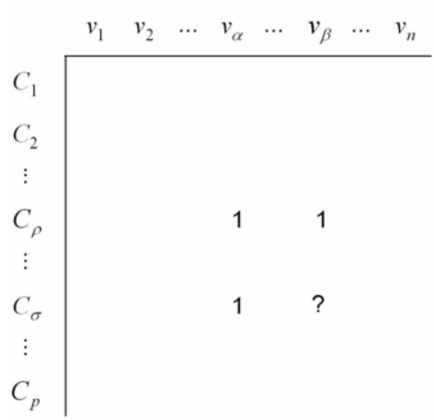

Figure 4: Matrix D in Γ-free form

Observation 1 *Suppose that D is Γ-free. Let $v_\alpha, v_\beta \in C_\rho$ where $\alpha < \beta$. If $v_\alpha \in C_\sigma$ for any $\sigma > \rho$, then $v_\beta \in C_\sigma$.*

Proof. The proof follows from the fact that D is Γ-free. See Figure 5. ■

Observation 2 *Suppose that D is Γ-free. Let M_ρ be any clique of size l in C_ρ. Let Y_ρ be a clique of size l in C_ρ such that the vertices of Y_ρ form the l rightmost columns of D which have nonzero entries in row ρ of D. Then if M_ρ is in C_σ for some $\sigma \geq \rho$, Y_ρ is also in C_σ.*

Proof. The proof follows from Observation 1. ■

Lemma 2 *Output $\Re$ of algorithm 1 is a minimum K_l-clique transversal of G.*

Proof. It is easy to see that $\Re$ is a K_l-clique transversal of G. To show that $\Re$ is minimum, we prove that $\Re_i$ is a subset of some minimum K_l-clique transversal of G for every $i = 1, 2, ..., p$. We use induction on the number of rows of D.

Initially, $\Re_0$ is empty and hence a subset of any minimum K_l-clique transversal of G. Suppose $\Re_{i-1}$ is a subset of some minimum K_l-clique transversal S of G. If C_i has a member of $\Re_{i-1}$, then $\Re_i = \Re_{i-1}$ and hence $\Re_i$ is a subset of S, a minimum K_l-clique transversal of G. Suppose C_i does not have any member of $\Re_{i-1}$. Let M_i be member of S which is a subset of C_i. Say $\overline{S} = S \setminus \{M_i\} \cup \{Y_i\}$. By Observation 2, if a maximal clique C_α, $\alpha \geq i$, contains M_i, then it contains Y_i. Thus $\overline{S}$ is a K_l-clique transversal of G. Since S and $\overline{S}$ are of the same cardinality, $\overline{S}$ is a minimum K_l-clique transversal of G. Thus $\Re_i$ is a subset of $\overline{S}$, a minimum K_l-clique transversal of G. By induction, $\Re$ is a minimum K_l-clique transversal of G. ■

A $\Gamma-$ free clique matrix for a strongly chordal graph can be achieved in time $O(L \log n)$ [15] and in time $O(n^2)$ [16] where L is the number of nonzero entries of the $n \times n$ Boolean matrix. Thus we state that

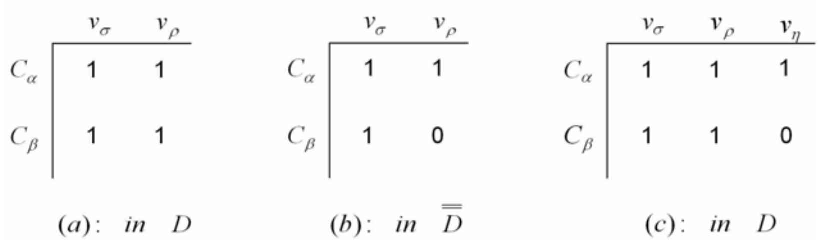

Figure 5: Submatrices of D and $\overline{\overline{D}}$

Theorem 5 *The K_l-clique transversal problem has a polynomial solution for strongly chordal graphs.*

3.2 A parallel top-down scan

In this section, we give an NC algorithm to solve the K_l-clique transversal problem for strongly chordal graphs. We shall transform the K_l-clique transversal problem on strongly chordal graphs to the clique transversal problem on strongly chordal graphs and solve the K_l-clique transversal problem on strongly chordal graphs. To do so, we need to prune the clique matrix D of G.

3.2.1 Pruning matrix D

Let $G(V, E)$ be a strongly chordal graph. As before, we assume that the clique matrix D of G is Γ-free. We prune the matrix D to form $\overline{\overline{D}}$ as follows:

If a row of D has less than $l-1$ nonzero entries, replace all the nonzero entries of the row by zeroes. If a row of D has more that $l-1$ nonzero entries, replace the $l-1$ rightmost nonzero entries by zeroes. The resultant matrix of D is denoted by $\overline{\overline{D}}$. The graph corresponding to $\overline{\overline{D}}$ is denoted by $\overline{\overline{G}}$.

Observation 3 *$\overline{\overline{D}}$ is Γ-free and $\overline{\overline{G}}$ is strongly chordal.*

Proof. Suppose $\overline{\overline{D}}$ is not Γ-free. Then $\overline{\overline{D}}$ has a Γ submatrix which is Γ-free. Say $\overline{\overline{D}}$ has the submatrix given in Figure 5(b). Since $d(\beta, \rho) = 0$ in $\overline{\overline{D}}$ and $d(\beta, \rho) = 1$ in D, row β has at most $l-2$ nonzero entries to the right of column ρ in matrix D. Since $d(\alpha, \rho) = 1$ in $\overline{\overline{D}}$, row α has at least $l-1$ nonzero entries to the right of column ρ in matrix D. Thus there exists $\eta > \rho$ such that $d(\alpha, \eta) = 1$ and $d(\beta, \eta) = 0$ in D. See Figure 5(c). This implies that D is not Γ-free which is a contradiction.

Since $\overline{\overline{D}}$ is Γ-free,$\overline{\overline{G}}$ is strongly chordal [12, 13]. ■

Observation 4 *The cardinality of a minimum clique transversal of $\overline{\overline{G}}$ is the same as that of a minimum K_l-clique transversal of G.*

Proof. Suppose α and β denote the cardinalities of a minimum clique transversal of $\overline{\overline{G}}$ and a minimum K_l-clique transversal of G respectively. Let S denote a minimum clique transversal of $\overline{\overline{G}}$. For every column j (vertex v_j) of S, let $\overline{j}$ be the row of $\overline{\overline{D}}$ which contains the uppermost nonzero entry of column j of $\overline{\overline{D}}$. Say Y_i is the l rightmost columns of D which have nonzero entries in row i of D. Define $S_l = \{Y_{\overline{j}} \mid v_j \in S\}$. Since D is Γ-free, $Y_{\overline{j}}$ is in all those rows i of D which satisfy the conditions that $i \geq \overline{j}$ and $d(i,j) = 1$. That is, $Y_{\overline{j}}$ of S_l covers all those maximal cliques which are covered by v_j of S. Thus S_l is a K_l-clique transversal of G. Therefore, $\beta \leq \alpha$.

To prove the converse, consider a minimum K_l-clique transversal S_l of G. For any Y of S_l, let l_Y be the leftmost column of Y in D. That is, if $Y = \{v_{y_1}, v_{y_2}, ..., v_{y_\mu}\}$ such that $y_1 < y_2 < ... < y_\mu$, then $l_Y = y_1$. Define $S = \{v_{l_Y} \mid Y \in S_l\}$. Since S_l covers all the maximal cliques of size $\geq l$, S is clique transversal of $\overline{\overline{G}}$. Therefore, $\alpha \leq \beta$. ■

Here we present a parallel algorithm for the K_l-clique transversal problem on strongly chordal graphs:

Algorithm 2

1. Input clique matrix D of G in the Γ-free ordering
2. Prune D to get the matrix $\overline{\overline{D}}$ and $\overline{\overline{G}}$.
3. Find a minimum clique transversal S of $\overline{\overline{G}}$.
4. For every j of S, find $\overline{j}$ where $\overline{j}$ is the row of $\overline{\overline{D}}$ which contains the uppermost nonzero entry of column j of $\overline{\overline{D}}$.
5. Design $S_l = \{Y_{\overline{j}} \mid j \in S\}$ where Y_i is the l rightmost columns of D which have nonzero entries in row i of D.
6. Output S_l, a minimum K_l-clique transversal of G.

Lemma 3 *The output S_l of Algorithm 2 is a minimum K_l-clique transversal of G.*

Proof. First we claim that S_l is K_l-clique transversal of G. Suppose C_α is a maximal clique of G of size $\geq l$. That is, row α has nonzero entries in $\overline{\overline{D}}$. Since S is a clique transversal of $\overline{\overline{G}}$, $\exists$ a $v_j \in S$ such that column j has nonzero entry in row α. By the definition of $\overline{j}$, we have $\overline{j} \leq \alpha$. By the construction, Y_j of D is in S_l. Since D is Γ-free, $Y_{\overline{j}}$ is in row α of D. Thus the maximal clique C_α has member of S_l. Hence S_l is a K_l-clique transversal of G.

The cardinality of S_l is the same as that of a minimum clique transversal of $\overline{\overline{G}}$. By observation 4, the cardinality is minimum. ■

Lemma 4 *Algorithm 2 runs in $O(log^3 n)$ time with $O(n^4)$ processors.*

Proof. Dahlhaus and Karpinski [8] have given a parallel algorithm to construct a strong elimination ordering of a strongly chordal graph which runs in $O(log^2 n)$ time with $O(n^2)$ processors. Given a strong elimination ordering, one can achieve a Γ-free clique matrix D of G by the method explained in Section 2 (one has to sort the maximal cliques by lexicographic ordering based on the strong elimination ordering). This sorting can be done in $O(log^2 n)$ time with $O(n^2)$ processors [10]. To prune matrix D to get $\overline{\overline{D}}$, it is enough to find the $l-1$ rightmost nonzero entries of row i in D. This can be done by parallel prefix computation [11] in $O(log\ n)$ time with $O(n^2)$ processors. A minimum clique transversal of a strongly chordal graph can be located in $O(log^3 n)$ time with $O(n+m)$ processors [7]. The $\overline{j}$ is the uppermost row of $\overline{\overline{D}}$ with nonzero entry at column j and it can be found using parallel prefix computation method. The $Y_{\overline{j}}$ is the l rightmost columns which have nonzero entries in row i of D and again it can be computed by parallel prefix computation method. ■

Thus we conclude that

Theorem 6 *Given a strong elimination ordering of a strongly chordal graph G, the K_l-clique transversal problem can be solved in $O(log^3 n)$ time with $O(n^4)$ processors.*

4 Conclusion

We have shown that the K_l-clique transversal problem has a polynomial time algorithm on the class of strongly chordal graphs. The immediate question is:

> Does this problem have a polynomial solution in balanced graphs?

Before probing this question, it may be useful to analyze possible extensions of Theorem 3.

> If $H(V, \bar{E})$ is a balanced hypergraph, is $B(H, \Im)$ balanced where $\Im = \{S_1, S_2, \ldots, S_r\}$ is a family of subsets of V?

A positive answer would allow us to apply integer programming on balanced matrices and to solve the K_l-clique transversal problem for balanced graphs.

A polynomial algorithm for the k-fold clique problem on balanced graphs has been discussed in [9]. An NC algorithm for this problem in strongly chordal graphs does not seem to be straightforward. The technique we use to solve the K_l-clique transversal problem on strongly chordal graphs does not work for the k-fold transversal problem. It would be interesting to explore the k-fold clique transversal problem to design an NC algorithm on strongly chordal graphs.

References

[1] T. Andreae, M. Schughart and Zs. Tuza, Clique-transversal sets of line graphs and complements of line graphs, *Discrete Math.* **88** (1991), 11–20.

[2] M. S. Chang, Y. H. Chen, G. J. Chang and J. H. Yan, Algorithmic aspects of the generalized clique transversal problem on chordal graphs, *Discrete Appl. Math.* **66** (1996), 189–203.

[3] M. Conforti, D. G. Corneil and A. R. Mahjoub, k_i-covers i: Complexity and polytopes, *Discrete Math.* **58** (1986), 121–142.

[4] M. Conforti, D. G. Corneil and A. R. Mahjoub, k_i-covers ii: k_i-perfect graphs, *J. Graph Theory* **4** (1987), 569–584.

[5] D. G. Corneil and J. Fonlupt, The complexity of generalized clique covering, *Discrete Appl. Math.* **22** (1989), 109–118.

[6] G. J. Chang, M. Farber and Zs. Tuza, Algorithmic aspects of neighborhood numbers, *SIAM J. Discrete Math.* **6** (1993), 24–29.

[7] E. Dahlhaus and P. Damaschke, The parallel solution of domination problems on chordal graphs and strongly chordal graphs, *Discrete Appl. Math.* **52** (1994), 261–273.

[8] E. Dahlhaus and M. Karpinski, Fast parallel computation of perfect and strongly perfect elimination schemes, *Technical Report 8513-CS, Institut für informatik, Universität Bonn*, Nov. 1987.

[9] E. Dahlhaus, P. D. Manuel and M. Miller, Maximum h-colourable subgraph problem in balanced graphs, *Inf. Processing Letters* **65** (6) (1998), 301–303.

[10] R. Karp and V. Ramachandran, Parallel algorithms for shared memory machines, *Handbook of Theoretical Computer Science: Algorithms and Complexity* Vol. A (ed. J. Van Leeuwen) (1990), 869–932.

[11] R. Ladner and M. Fisher, Parallel prefix computation, *J. ACM* **27** (1980), 831–838.

[12] A. Lubiw, γ-free matrices, *M.Sc. Thesis, University of Waterloo*, 1982.

[13] A. Lubiw, Doubly lexical orderings of matrices, *SIAM J. Computing* **16** (1987), 854–879.

[14] M. V. Marathe, R. Ravi and C. Pandurangan, Generalized vertex covering in interval graphs, *Discrete Appl. Math.* **39** (1992), 87–90.

[15] R. Paige and R. E. Tarjan, Three partition refinement algorithms, *SIAM J. Computing* **16** (1987), 973–989.

[16] J. Spinrad, Doubly lexical ordering of dense 0-1 matrices, *Inf. Processing Letters* **45** (1993), 229–235.

[17] Zs. Tuza, Covering all cliques of a graph, *Discrete Math.* **86** (1990), 117–126.

[18] G. Venkatesan and C. Pandurangan, Algorithmic aspects of clique-transversal and clique-independent sets, *Discrete Appl. Math.* **100** (2000), 182–202.

(Received 3 July 1998; revised 5 July 2005)