FAST PARALLEL DECOMPOSITION BY CLIQUE SEPARATORS

Elias Dahlhaus

and

Marek Karpinski*

Dept. of Computer Science
University of Bonn

Abstract.

We design a fast parallel algorithm for decomposing an arbitrtary graph by the clique separators. The algorithm works in $O(log^2 n)$ parallel time and $O(n^4)$ processors on a CREW-PRAM. It is the first sublinear parallel time (and therefore sequential sublinear space) algorithm for this problem.

1. Introduction

For some time the problem of clique separator decomposition of a graph was considered to be inherently sequential and very difficult to parallelize. The fastest known sequential algorithm of Tarjan [Ta] works in $O(nm)$ time and is based on a 'highly sequential' subroutine for computing a minimal ordering of a graph.

In this paper we develop the first fast (polylogarithmic time) parallel (NC-) algorithm for decomposing an arbitrary graph by clique separators. The algorithm works in $O(log^2 n)$ parallel time and $O(n^4)$ processors on a CREW-PRAM.

For the background of the clique separator theory and its applications we refer to [Ta]. Since the other special problems mentioned in [Ta] of finding a maximum weight clique, maximum weight independent set and graph coloring are NC^1-

*Supported in part by Leibniz Center for Research in Computer Science and the DFG Grant KA 673/2-1

reducible to the clique separator decomposition, all of them are also proven be in NC.

We refer to [Ta] for general terminology on clique separators and chordal extensions, and to [Co] for the basic models of parallel computation.

2. Parallel Clique Decomposition

We shall look for an NC-algorithm satisfying the following I/O property:

Input: A graph $G = (V, E)$

Output: The set of all minimal separating complete vertex sets, the so-called *clique separators.*

Tarjan's [Ta] technique was an inclusion minimal extension of the edge set to an edge set E', so that $G' = (V, E')$ is a chordal graph. Then each clique separator of G is also a clique separator of G'. Moreover, each clique separator is the intersection of the neighbourhoods of two vertices.

The main step of Tarjan's algorithm which is not easily parallelizable is the construction of the chordal extension G'. We shall give an alternative construction of the edge set of G, such that each clique separator is the intersection of two neighbourhoods. The idea is similar to the chordality test of Tarjan and Yannakakis [TY] or Naor, Naor and Schäffer [NNS].

Algorithm *Fill:*

Input: a graph $G = (V, E)$. Let $n = \#V$

Begin: Let $E'_0 := E$.
For each x let $N(x) := \{y | xy \in E\}$;

Repeat $log_2\ n$ times: all edges of E'_i are also in E'_{i+1}. For each x, y, z such that $xy \in E'_i$ and $yz \in E'_i$ and each connected component of $G \restriction (V \setminus N(y))$, set $xz \in E'$ if there are vertices $x', z' \in C$, such that $xx', zz' \in E$.

End Repeat.

Output $G' := (V, E') := (V, E'_{\log_2 n})$.

We have to prove the following

Lemma 1: Each clique separator of G is also a clique separator of G'.

PROOF: Consider a clique separator c of G which separates vertices x and y. We shall prove the lemma for each (V, E'_m) by induction on m.
For $i = 0$ it is true by definition. Assume x and y are not separated by c in (V, E'_{m+1}). Then there is a path p in (V, E'_{m+1}) from x to y not passing any vertex of c, p contains an edge $x_{j-1}x_j$, such that x_{j-1} is in the E'_i-component of x and x_j is in the E'_i-component of y, $x_{j-1}\, x_j \in E'_{i+1} \backslash E'_i$. The common E'_m neighbourhood of x_{j-1} and x_j must be in c. Otherwise one can reach x_j by a path in (V, E'_m) without passing c. Since $x_{j-1}x_j \in E'_{m+1} \backslash E'_m$, there is a vertex u, such that $x_{j-1}u, x_j u \in E'_m$ and a path $q = (x_{j-1} = y_0, y_1, \cdots, y_{k-1}, y_k = x_j)$, such that no $y_i, i = 1, \cdots, k-1$ is adjacent to u in $G = (V, E)$. The vertex u must be in c, because u is in the common neighbourhood of x_{j-1} and x_j. Therefore no y_i, $i = 0, \cdots, k$ is in c. But then c does not separate x_{j-1} and x_j in (V, E) and therefore also not in (V, E'_m). Therefore x and x_j are not separated by c in (V, E'_m). This is a contradiction. ∎

A second useful result is the following

Lemma 2: Let $d := (x_1, \cdots, x_k, x_{k+1} = x_1)$ be a chordless cycle of G; that means $x_i x_j \in E$ iff $|i - j| = 1 \bmod k$. Then for all $i \neq j$ $x_i x_j \in E'$. This means chordless cycles are filled to a clique by the algorithm *Fill*.

PROOF: We shall prove the following by induction over m.

Claim: For all i and all $j \leq 2^m$ $x_i x_{i+j} \in E'_m$:
For $m = 0$ the claim is true by definition.
Let $j \leq 2^{m+1}$. Then we find $j_1, j_2 \leq 2^m$, such that $j = j_1 + j_2$. By the assumption of the induction $x_i x_{i+j_1}$, $x_{i+j_1} x_{i+j_1+j_2} \in E'_m$. Consider the path $q = (x_{i+j} = x_{i+j_1+j_2}, x_{i+j+1(mod k+1)}, x_{i+j+2(mod k+1)}, \cdots, x_{i-1(mod k+1)}, x_i)$. All elements of q with the exception of x_i and x_{i+j} are not in the neighbourhood of x_{i+j_1} in G. Therefore x_i and x_{i+j} are adjacent to the same neighbourhood of $G \backslash N(x_{i+j_1})$. Since $x_i x_{i+j_1}$ and $x_{i+j1} x_{i+j} \in E'_m$ $x_i x_{i+j} \in E'_{m+1}$.
END OF THE PROOF OF THE CLAIM

Since $k \leq n =$ number of vertices, after $\log_2 n$ steps all vertices of the cycle d are pairwise adjacent in $(V, E') = (V, E'_{\log_2 n})$. ∎

The key for the construction of all clique separators is the following

Theorem 1: All clique separators of G are intersections of two neighbourhoods or a neighbourhood of one vertex of $G' = (V, E')$, where E' is defined as in the algorithm *Fill*. For the proof of this theorem we consider two auxiliary results.

Lemma 3 (Tarjan): Let E'' be a minimal extension of E, such that $G'' = (V, E'')$ is a chordal graph. Then each clique separator of (V, E) is also a clique separator of G'', and each clique separator of G'' is a neighbourhood of one vertex or the intersection of the neighbourhood of two vertices.

The second result is the following

Lemma 4: Let $G'' = (V, E'')$ be a minimal chordal extension of $G = (V, E =)$. Then each edge $e \in E'' \backslash E$ joins two vertices of the same chordless cycle.

PROOF: We consider for G'' a tree T and a family $\mathcal{S}$ of subtrees of T, such that the vertex intersection graph of $\mathcal{S}$ is isomorphic to G''. For each vertex v of G'' let T_v be the corresponding subtree of T. We may also assume that each vertex t of T corresponds to a maximal clique, namely $c_t := \{v | t \in T_v\}$.

Consider an edge $uv \in E'' \backslash E$. Then there is a cycle of G'' which has no chord except uv. But such a cycle must be of length=4. Therefore u and v have a common neighbourhood which is not complete. Hence u and v are in more than one maximal clique of G''.

Let us remark that each maximal clique of G'' is of the form c_t. Let t_1 and t_2 be leaves of $T_0 := T_u \cap T_v$. Since c_{t_1} and c_{t_2} are maximal cliques, we find vertices v_1, v_2 which are in c_{t_1} or in c_{t_2}, respectively, but in no other c_t, such that $t \in T_u \cap T_v$. (see Fig. 1)

Figure 1

We consider the cycle $C_0 := (u, v_2, v, v_1, u)$. uv is the only chord of C_0. $uv \in E'' \backslash E$.

i) All chords of C_0 are in $E'' \backslash E$.

ii) Let x and y be neighbours in C_0. Then they have only one clique c_t in $T_0 = T_u \cap T_v$ in common.

iii) Suppose x and y are no neighbours in C_0. Then $T_x \cap T_y \subseteq T_0$.

We construct stepwise a sequence (C_i, T_i), such that $T_i \subseteq T_{i+1}$ and $V(C_i) \subseteq V(C_{i+1})$ and i), ii), and iii) are satisfied where C_0, T_0 is replaced by T_i, C_i, and for the last member (T_i, C_i) of the sequence, C_i is a cycle of G. The sequence is constructed as follows:

Let C_i and T_i be just constructed. Let x, y be neighbours in C_i and $xy \notin E$. Let c_t be the common clique of x and y in T_i. $T_{i+1} = T_i \cup (T_x \cap T_y)$. Let t' be a leaf of $T_x \cap T_y$, but $t' \neq t$. This exists since x and y have more than one common clique. Then t' is a leaf of T_{i+1}. Since also $c_{t'}$ is a maximal clique, we find a vertex z, such that $T_{i+1} \cap T_z = \{t'\}$. let $C_i = (x, y, u_1, \cdots, u_i, x)$. Then $C_{i+1} := (x, y, z, u_1, \cdots, u_i, x)$. It is easily seen that also $(T_{i+1}C_{i+1})$ satisfies i)-iii), if (T_i, C_i) does.

¿From the last lemma we can conclude the following

Lemma 5: Let $G' := (V, E')$ be the output of $Fill(G) = Fill(V, E)$. Then for any minimal chordal extension (V, E'') of G, $E'' \subseteq E'$.

Since by the algorithm *Fill* all chordless cycles are filled to complete sets and each clique separator of G is also a clique separator of the output G' of $Fill(G)$, the theorem is proved.

Now we change the algorithm *Fill* to an equivalent algorithm with an acceptable time and processor number.

Algorithm *Fill′:*

Input: a graph $G = (V, E)$, such that $n = \#V$.

Begin

1) for each $x \in V$, let $\mathcal{C}_x$ be the set of connected components of $G \upharpoonright (V \backslash N(x)), (O(\log^2 n)$ time and $O(n^3)$ processors).
2) for each $x \in V, y \in V$ and each $q \in C_x$, let $(x, y, q) \in Z$ iff $\exists v \in q, yv \in E, (O(\log n)$ time, $O(n^4)$ processors).
3) **Initialize:** $E_0 := E$

Repeat log n **times:** $E_{i+1} = \{y_1y_2 | \exists x \in V, \exists q \in C_x(x, y_1, q), (x, y_2, q) \in Z \cup xy_1, xy_2 \in E_i\} \cup E_i$. (Each step of the repeat loop needs $O(\log n)$ time and $O(n^4)$ processors). Output $G := (V, E') := (V, E_{\log n})$.

End

Therefore

Theorem 2: The algorithm *Fill′* needs $O(n^4)$ processors and $O(\log^2 n)$ time.

Now we have to complete the set of clique separators of G.

Algorithm *Clique Sep*:

Input $G := (V, E), \#V = n$

1) Execute *Fill′* with output $G' := (V, E')$

2) For each $x, y \in V, xy \notin E'$
   Let $c_{xy} := \{v | xy \in E' \wedge yv \in E'\}$($O(n^3)$ processors and $O(\log n)$ time).
3) For each $xy \notin E'$ check whether c_{xy} separates x and y in G, i.e. whether x and y are in different connected components of $G \restriction V \backslash c_{xy}$ ($O(n^4)$ processors and $O(\log^2 n)$ time).

   Output all x and y separating c_{xy}.

   Therefore, we formulate our main Theorem:

Theorem 3: The set of all clique separators of an arbitrary graph can be computed in $O(\log^2 n)$ time by $O(n^4)$ processors on a CREW-PRAM. $\square$

References.

[CL] Chin-Wen Ho, and Lee, R.C.T., *Efficient Parallel Algorithms for Finding Maximal Cliques, Clique Trees, and Minimum Coloring on Chordal Graphs*, Preprint National Tsing Hua University, Hsinchu, Taiwan

[Cu] Cunningham, W.H., *Decomposition of Directed Graphs*, SIAM Journal of Algebraic and Discrete Methods, Vol. 3, No. 2(1982), pp. 214-228

[Co] Cook, S.A., *A Taxonomy of Problems with Fast Parallel Algorithms*, Information and Control 64(1985), pp. 2-22

[DK 1] Dahlhaus, E., and Karpinski, M., *A Fast Parallel Algorithm for Computing All Maximal Cliques in a Graph and the Related Problems*, to appear in SWAT

[DK 2] Dahlhaus, E., and Karpinski, M., *Fast Parallel Computation of Perfect and Strongly Perfect Elimination Schemes*, submitted for publication

[Ga] Gavril, F., *Algorithms for Minimum Coloring, Maximum Clique, Minimum Coloring by Cliques, and Maximum Independent Sets of a Chordal Graph*, SIAM J. Comput. (1972), pp. 180-187

[GS] Goldberg, M., and Spencer, T., *A New Parallel Algorithm for the Maximal Independent Set Problem*, Proc. 28^{th} IEEE FOCS (1987), pp. 161-165

[Lu] Luby, M., *A Simple Parallel Algorithm for the Maximal Independent Set Problem*, Proc. 17^{th} ACM STOC (1985), pp. 1-9

[NNS] Naor, J., Naor, M., and Schäffer, A., *Fast Parallel Algorithms for Chordal Graphs*, Proc. 19^{th} ACM STOC (1987), pp. 355-364

[TNS] Takamizawa, K., Nishizeki, T., and Saito, N., *Combinatorial Problems on Series Parallel Graphs*, Journal of the ACM, Vol. 29 (1982), pp. 623-641

[Ta] Tarjan, R., *Decomposition by Clique Separations*, Discrete Mathamatics 55 (1985), pp. 221-232

[TY] Tarjan, R., and Yannakakis, M., *Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Hypergraphs; Addendum*, SIAM J. Comput. 14 (1985), pp. 254-255

[Wh] Whitesides, F.H., *An Algorithm for Finding Clique Cutsets*, Information Processing Letters 12 (No 1) (1981), pp. 31-32