The Parallel and Sequential Complexity of Matching on Chordal and Strongly Chordal Graphs

Elias Dahlhaus
Basser Department of Computer Science, University of Sydney, Australia
Marek Karpinski *
Department of Computer Science
University of Bonn, Bonn, Germany

*

Abstract

0 Introduction

Chordal graphs became interesting as a generalization of interval graphs (see for example [9]). We call a graph chordal if every cycle of length greater than three has a chord, i.e. an edge that joins two non consecutive vertices of the cycle. Note that interval graphs are not only chordal but strongly chordal as defined in [3]. Strongly chordal graphs are just those chordal graphs having a so called strongly perfect elimination ordering.

In this paper we consider the sequential and parallel complexity of the maximum matching problem in chordal and strongly chordal graphs. Note that in general a linear time algorithm for perfect matching is not known. Here we shall show that, provided a strongly perfect elimination ordering is known, a maximum matching in a strongly chordal graph can be found in linear time by a simple greedy algorithm. This algorithm can be turned into a (non optimal) parallel algorithm. The random bits in the algorithm of [10] can be eliminated in the special case of strongly chordal graphs. We also will see that these techniques can be extended to multidimensional matching that is NP-complete in general (see for example [5]).

On the other hand, we shall find out that matching restricted to chordal graphs (also restricted to path graphs) is of the same parallel complexity degree as bipartite matching.

In section 1, we shall introduce the basic notation. In section 2 we consider the sequential and parallel complexity of maximum matching restricted to strongly chordal graphs. In section 5 we discuss the parallel complexity of matching restricted to path graphs.

1 Notation and Basic Definitions

A *graph* $G = (V, E)$ consists of a *vertex set* V and an *edge set* E. Multiple edges and loops are not allowed. The edge joining x and y is denoted by xy.

We say that x is a *neighbor* of y iff $xy \in E$. The *full neighborhood* of x is the set $\{x\} \cup \{y : xy \in E\}$ consisting of x and all neighbors of x and is denoted by $N(x)$.

A *path* is a sequence $(x_1 \dots x_k)$ of distinct vertices such that $x_i x_{i+1} \in E$.

A *cycle* is a closed path, that means a sequence $(x_0 \dots x_{k-1} x_0)$ such that $x_i x_{i+1 \pmod k} \in E$.

A *subgraph* of (V, E) is a graph (V', E') such that $V' \subset V$, $E' \subset E$.

An *induced subgraph* is an edge-preserving subgraph, that means (V', E') is an induced subgraph of (V, E) iff $V' \subset V$ and $E' = \{xy \in E : x, y \in V'\}$.

A graph (V, E) is *chordal* iff each cycle $(x_0 \dots x_{k-1} x_0)$ of length greater than 3 has an edge $x_i x_j \in E, j - i \neq \pm 1 \bmod k$ (which joins vertices which are not neighbors in the cycle). Sometimes they are also called *triangulated* or *rigid circuit* graphs. We remark that this notion is equivalent to the nonexistence of an induced cycle of length greater than 3.

Independently Gavril [6] and Buneman [1] proved the following:

Theorem 1 *A graph is chordal iff it is the intersection graph of vertices of subtrees of a tree, i.e. the vertices of the chordal graph correspond to subtrees of a fixed tree and two vertices of the chordal graph are joined by an edge iff the corresponding subtrees share a vertex of the tree.*

A *path graph* is the intersection graph of a collection of paths of a tree.

We also can define chordal graphs by characteristic orderings.

Theorem 2 *[4] A graph $G = (V, E)$ is chordal iff there is an ordering $<$ of V, such that with $x < y$, $x < z$, $xy \in E$, and $xz \in E$, we have $yz \in E$. Such an ordering is called a* perfect elimination ordering.

A graph $G = (V, E)$ is called *strongly chordal* [3] iff there is an ordering $<$ on the vertices of V such that

1. for $xy, xz \in E$, such that $x < y$ and $x < z$, also $yz \in E$,
2. for $x_1y_2, x_2y_1, x_1x_2 \in E$, such that $x_1 < y_1$ and $x_2 < y_2$, we have $y_1y_2 \in E$.

Such an ordering is called a *strongly perfect elimination ordering.*

A *matching* of $G = (V, E)$ is a subset M of E such that no two edges share a vertex. A matching of maximal size is called a *maximum matching.* If all vertices of G belong to an edge of the matching M then M is called a *perfect matching.*

2 Maximum Matching Algorithms for Strongly Chordal Graphs

We assume that a strongly perfect elimination ordering $<$ of the vertex set of the graph $G = (V, E)$ i.e. the corresponding enumeration $(v_1, \ldots v_n)$ is given.

We claim that the following algorithm computes a maximum matching in a strongly chordal graph.

```
1. $V' := V$; $M := \emptyset$;
2. Repeat
      $uv$ is an edge in $E$ with $u, v \in V'$, $u$ is minimal with respect to $<$, $v$
         is the $<$-smallest vertex in $V'$ than is adjacent to $u$;
```

$M := M \cup \{uv\}$; $V' := V' \setminus \{u, v\}$

until there are no edges in E with both incident vertices in V'.

It is easily seen that this algorithm has a time bound of $O(n + m)$.

We have to show the correctness.

For a matching M of G, we call a pair of edges u_1u_2 and w_1w_2 in M a *defect* of M if

1. $u_1w_1 \in E$,
2. $u_1 < w_2$, and $w_1 < u_2$.

Lemma 1 *If there is a matching of cardinality k then there is a defect free matching of the same cardinality k.*

Proof: We label the edge v_iv_j with $l_{v_iv_j} := (i - j)^2$. Suppose there is a defect consisting of the pair u_1u_2 and w_1w_2. Then, by definition $u_1w_1 \in E$. Since $u_1 < w_2$ and $w_1 < u_2$ and $<$ is a strongly perfect elimination ordering, $u_2w_2 \in E$. Therefore we get a matching M' where the edges u_1u_2 and w_1w_2 are replaced by the edges u_1w_1 and u_2w_2.

Claim: $\Sigma_{e \in M'} l_e < \Sigma_{e \in M} l_e$.

Proof of Claim: For simplicity, we identify the vertices with their indices v_i.

We consider the following subcases:

First case $u_1 < w_2 < w_1 < u_2$:

$$\begin{aligned}(u_2 - w_2)^2 + (w_1 - u_1)^2 &= (u_2 - w_2)^2 + ((w_1 - w_2) + (w_2 - u_2))^2 \\ &= (u_2 - w_2)^2 + (w_1 - w_2)^2 + 2(w_1 - w_2)(w_2 - u_1) + (w_2 - u_1)^2 \\ &< (w_1 - w_2)^2 + (u_2 - w_2)^2 + 2(u_2 - u_1)(w_2 - u_1) + (w_2 - u_1)^2 \\ &= (w_1 - w_2)^2 + (u_2 - u_1)^2.\end{aligned}$$

Second case $u_1 < w_1 < w_2 < u_2$:

$$(w_1 - u_1)^2 + (u_2 - w_2)^2 < (u_2 - u_1)^2 < (u_2 - u_1)^2 + (w_2 - w_1)^2$$

Third case $u_1 < w_1 < u_2 < w_2$: Then the inequality $(w_1 - u_1)^2 - (w_2 - u_2)^2 < (u_2 - u_1)^2 + (w_2 - w_1)^2$ follows immediately.

All other possible cases are permutations of the cases as considered.

□(Claim)

Clearly after the removal of several defects, we find a matching of the same cardinality with a minimum sum of labels l_e. This matching is free of defects.

□(Lemma)

Lemma 2 *The matching obtained by above algorithm is defect free.*

Proof: Suppose there is a defect u_1w_1, u_2w_2 with $u_1 < w_2$, $u_2 < w_1$ and $u_1u_2 \in E$. Suppose $u_1 < u_2$. Then w_1 is not the minimal choice of a neighbor as required by the algorithm.

□(Lemma)

Theorem 3 *The matching computed by the above algorithm is a maximum matching.*

Proof: We consider any defect free maximum matching M and the matching M' computed by the above algorithm.

Let x be the smallest vertex y such that M restricted to $\{u|u \leq y\}$ and M' restricted to $\{u|u \leq y\}$ are different. Then M restricted to $\{u|u < x\}$ and M' restricted to $\{u|u < x\}$ coincide. It cannot be that x is covered by an edge of M but not by an edge of M', because necessarily x has a neighbor that is not covered by M restricted to $\{u|u < x\}$ and the edge joining x with the minimum neighbor t must be in M'(if $t < x$ then x is chosen as the smallest neighbor of t not covered by the matching considered before. If $x < t$ then t is chosen as the smallest neighbor of x by above algorithm). Suppose x is in an edge of M' but does not appear in an edge of M. Note that x has a neighbor t that is not in an edge of M restricted to $\{u|u < x\}$. We add xt to M and delete the edge $ty \in M$ if such an edge exists.

Therefore we may assume that there are edges xt of M and an edge xt' of M' that are incident with x. Note that $t' < t < x$ and t' does not belong to an edge of M restricted to $\{u|u < x\}$. Moreover, it cannot belong to an edge of M. Otherwise there is an edge $t'y \in M$ with $y > x$ and $t'y$ and xt forms a defect. Therefore in M, we can replace xt by xt' and the new matching M coincides with M' in $\{u|u \leq x\}$. By induction, we get a maximum matching that coincides with M'.

□ (theorem)

Corollary: For strongly chordal graphs, a maximum matching can be computed in linear time

Theorem 4 *In strongly chordal graphs, one can find a perfect matching by a CREW-PRAM in $O(\log^2 n)$ time with a polynomial processor bound if a perfect matching exists.*

Proof: We prove that there is at most one defect free perfect matching. Since this is the perfect matching with the minimum sum of labels $l_{uv} = (u-v)^2$, we get a perfect matching by the minimum perfect matching algorithm of [10] in $O(\log^2 n)$ time with a polynomial processor bound.

Lemma 3 *There exists at most one defect free perfect matching.*

Proof: Assume there are defect-free perfect matchings M and M'. Assume M and M' coincide in $\{u|u < x\}$ but not in $\{u|u \leq x\}$. Suppose $xt \in M$ and $xt' \in M'$ are the edges in M and M' respectively that are incident with

x. Without loss of generality, we assume that $t' < t$. Both t and t' do not appear in any edge of M and M' with both incident vertices in $\{u|u < x\}$. Therefore the edge $t'u$ in M that is incident with t' must have the property that $u > x$. But then $t'u$ and xt form a defect in M. This is a contradiction.

□(lemma)

□(theorem)

Remark: A strongly perfect elimination ordering of a strongly chordal graph can be computed in $O(\log^4 n)$ time with a linear processor number [2]. Therefore it is possible to get an NC-algorithms to compute a perfect matching in strongly chordal graphs also without the knowledge of a strongly perfect elimination ordering.

3 Multidimensional Matching in Strongly Chordal Graphs

The problem of *multidimensional matching* is to find, for given graph $G = (V, E)$ and natural number k, a maximum number of pairwise disjoint complete sets of cardinality k. In general, even for $k = 3$, the problem is NP-complete (see for example [5], [Exact Cover by Triangles]). For strongly chordal graphs, the following generalization of the perfect matching algorithm computes a multidimensional matching.

Input $G = (V, E)$, k, output: a multidimensional matching M.

```
1. $V' := V$; $M := \emptyset$; $l := 0$; $d := \emptyset$;
2. Repeat
     if $l \neq k$ and $u$ is the $<$-minimum element in $V'$ that is adjacent to all
        vertices in $d$ then
         • $d := d \cup \{u\}$
         • $V' := V' \setminus \{u\}$
        if such a $u$ does not exit the set $k := 0$; If such a $u$ does not exist
        then output:
    If $l = k$ then
        • $M := M \cup \{d\}$
        • $k = 0$
until $V' = \emptyset$
```

It is easily seen that this algorithm can be implemented in linear time. It remains to show that this algorithm computes a maximum multidimensional matching.

We only have to show that if there is a k-matching with l complete sets of cardinality k then there is a k-matching of the same cardinality such that one of the complete sets consists of the smallest element x of V and its $k-1$ smallest neighbors.

Note that for all (greater) neighbors u and v of x with $u < v$, $N(u) \cup \{u\} \subseteq N(v) \cup \{v\}$.(*)

Let M be a k-matching and $c_1, \ldots c_q$ be the complete sets in M that intersect $N(x)$ but do not contain x. Let $c_1, \ldots, c_q$ be sorted with respect of their smallest elements in $N(x)$. By (*), we can replace each c_i by a complete set c'_i such that

1. $c'_i \setminus N(x) = c_i \setminus N(x)$
2. if $u, v, w \in N(x)$, $u, w \in c'_i$ then $v \in c'_i$, and
3. if $i < j$ then all vertices in c'_i are smaller than all vertices in c'_j.

Moreover, we can find c'_i with this property such that $\bigcup_{i=1}^{q} = \{v \in N(x) | v > u\}$, for some $u \in N(x)$. If u is the $k - 1^{th}$ smallest neighbor of x, we only have to add the complete set consisting of x and its $k-1$ smallest neighbors to M and we have a k-matching of cardinality $l+1$. Otherwise the $k-1$ smallest neighbors of x intersect only c'_1 and we can replace c'_1 by the complete set c''_1 consisting of x and its $k-1$ smallest neighbors.

Theorem 5 *Multidimensional Matching for strongly chordal graphs can be done in linear time.*

4 Multidimensional Matching in Chordal Graphs

Here we consider only the problem whether there is an exact cover of a given chordal graph by mutually disjoint complete sets of a certain cardinality k. We assume that $G = (V, E)$ and a perfect elimination ordering $<$ is given.

Let M be an exact cover of G by complete sets of cardinality k. We call the $<$-smallest vertices of any $c \in M$ *vertices of first kind* and all other vertices vertices of second kind.

Lemma 4 *If there is an exact cover of G by complete sets of cardinality k then there is an exact cover M by complete sets of cardinality k with the following property.*
If x and y are adjacent vertices of first kind and $x < y$ then for all vertices x' of the $c \in M$ x belongs to, $x' < y$.

Proof: Suppose x is the smallest vertex of $c \in M$ and y is the smallest vertex of $d \in M$, $x < y$, and $z \in c$ but $x < z$. Since $<$ is a perfect elimination ordering, $yz \in E$, and therefore also, since $y < z$ and $yw \in E$, for all $w \in d$, z is adjacent to all vertices in d. Therefore we can interchange the memberships of y and z in c and d, and the resulting collection of sets of cardinality k is still an exact cover of G by complete sets. After a finite number of applications of this procedure, we get an exact cover satisfying the requirements of the lemma. □

Theorem 6 *The problem to get an exact cover by complete sets of cardinality k in a chordal graph can be solved in polynomial time.*

Proof: We reduce the problem to the following variation of bipartite perfect matching.

Bipartite Multimatching: Given a bipartite graph $B = (V \cup W, E)$ and a natural number k, is there a subset M' of E such that each $x \in V$ belongs to exactly k edges of M and each $y \in W$ belongs to at most one edge of M.

Replacing each $x \in V$ by k copies, we get a reduction to the well known marriage problem.

A partial bipartite k-matching is a subset M of E such that each $x \in V$ belongs to at most k edges of M and each $y \in W$ belongs to at most one edge of M.

Note that, with V' as the set of vertices of first kind, W' as the set of vertices of the second kind, and $xy \in E'$ if $x \in V'$, $y \in W'$, $xy \in E$, and $x < y$, an exact cover by complete sets of size k translates into a bipartite $k-1$-matching in $B' = (V' \cup W', E')$ and vice versa. The only problem is to determine the vertices of first kind.

The following algorithm determines an exact cover by complete sets of cardinality k if it exists.

Input: a chordal graph $G = (V, E)$, a perfect elimination ordering $<$ of G, say $V = \{v_1, \ldots v_n\}$ with $v_i < v_j$ if $i < j$,and a natural number k.

Output: An exact covering M of G by complete sets of cardinality k.

1. $M' := \emptyset$; $V' := E' := \emptyset$;
2. for $i = 1, \ldots, n$ do
    - if v_i has no smaller neighbors then add v_i to V' and add all edges incident to v_i to E' else
    - Compute a maximum partial bipartite $k-1$-matching M'' of $B'[\{v_1, \ldots, v_i\}] = (V' \cup (\{v_1, \ldots, v_i\} \setminus V', E')$ by applying augmenting path techniques on M', i.e. all vertices $< v_i$ that are

```
    covered by $M'$ remain covered by $M''$.
    endif
    endfor
  • If $v_i$ does not belong to $M''$ then add $v_i$ to $V'$ and all edges $v_i y$
    with $v_i < y$ to $E'$;
  • $M' := M''$
3. For each $x \in V'$, $c_x := \{y | xy \in M'\}$; $M := \{c_x | x \in V'\}$.
```

In principle, the algorithm computes, for each v_i, a maximum $k-1$-matching M' for the bipartite graph B' restricted to $v_1, \ldots, v_i$ and if v_i is not covered by M' then v_i is made a vertex of first kind. That means vertices are made vertices of first kind only if there is no possibility to cover them by a maximum $k-1$-matching in such a way that all smaller vertices remain covered, i.e. the maximum partial $k-1$-matching of B' restricted to $\{v_1, \ldots v_{i-1}\}$ is also a maximum partial $k-1$-matching of B' restricted to $\{v_1, \ldots, v_i\}$. Any covering by complete sets of cardinality k satisfying the requirements of lemma 4 corresponds to a maximum $k-1$-matching of B'.

All single steps can be done in polynomial time. Therefore the whole algorithm works in polynomial time. □

5 The Parallel Complexity of Maximum Matching in Path Graphs

Theorem 7 *Suppose the we can find a perfect matching of a path graph in polylogarithmic time with a polynomial processor bound. Then we can find a perfect matching in a bipartite graph in polylogarithmic time with a polynomial processor bound, i.e. the marriage problem is in NC.*

Proof: We construct a reduction from the the bipartite perfect matching problem into the perfect matching problem restricted to path graphs that can be computed in logarithmic time with $O(n^2)$ processors.

Given a bipartite graph $B = (V \cup W, E)$ with all edges incident with exactly one vertex in V and exactly one vertex in W. Note that B has only a perfect matching if V and W have the same size.

We construct an interval representation as follows.

The tree T is consists of a main node c, vertices t_v, for each $v \in V$, and vertices $s_{w,i}$, $1 \leq i < deg(w)$, $w \in W$. The parent of each t_v and each $s_{w,1}$ is c and the the parent of each $s_{w,i}$ is $s_{w,i-1}$, for $i \neq 1$.

The collection $\mathcal{P}$ of paths is constructed as follows. For each node $t \neq c$ of T, we provide a one node path p_t containing exactly t, and for each $vw \in E$, we have a path q_{vw} containing t_v, c, and all nodes $s_{w,i}$.

It is easily seen that this path representation and therefore also the resulting path graph $G = (\mathcal{P}, E_G)$ can be constructed in $O(\log n)$ time with $O(n^2)$ processors by a CREW-PRAM.

It remains to show that each perfect matching in G induces a perfect matching in B and vice versa.

Suppose a perfect matching M og G is given. Note that there are as many paths p_t as paths q_{vw}. Note that each path p_t shares a node only with a path q_{vw}. Therefore a perfect matching of G consists only of edges of the form $p_t q_{vw}$. Since there are $deg(w) - 1$ nodes $s_{w,i}$, exactly and $deg(w)$ many paths $q_{v,w}$, exactly one path $q_{v,w}$ is matched with $p_{t,v}$, say $q_{v_w,w}$. Then $M' = \{v_w w | w \in W\}$ defines a perfect matching in B.

Vice versa, we assume that a perfect matching M' of B is given. For each $vw \in M'$, let $p_{t_v} q_{vw} \in M$ and for each $v'w \in E$ with $v' \neq v$, choose a distinguished number $i_{v'} < deg(w)$ and let $s_{w,i_{v'}} q_{v'w} \in M$. M defines a perfect matching of G.

□ (theorem)

6 Conclusions

We would like to mention that the parallel perfect matching algorithm for strongly chordal graphs is not optimal. It remains an intersting problem to find an optimal parallel perfect elimination algorithm for strongly chordal graphs.

Finally we would like to remark that strongly chordal graphs are exactly the chordal graphs that are complements of comparability graphs [7]. It is known that the perfect matching problem restricted to complements of co-comparability graphs is equivalent to 2-processor scheduling, and this can be done in $\log^2 n)$ time with a polynomial processor bound [8]. It might be interesting to find a reasonable upper class of strongly chordal graphs and complements of comparability graphs such that the perfect matching problem can still be parallelized.

7 Acknowledgements

Recently Stephan Olariu mentioned that his student Lin got similar results related to perfect matching on strongly chordal graphs. We are grateful to Avi Wigderson, Joseph Naor, and Alejandro Schaeffer for a number of intersting conversations.

References

[1] P. Bunemann, *A Characterization of Rigid Circuit Graphs*, Discrete Mathematics 9 (1974), pp. 205-212.

[2] E. Dahlhaus, *Chordale Graphen im besonderen Hinblick auf parallele Algorithmen*, Habilitation Thesis, University of Bonn, 1991.

[3] M. Farber, *Characterizations of Strongly Chordal Graphs*, Discrete Mathematics 43 (1983), pp. 173-189.

[4] D. Fulkerson, O. Gross, *Incidence Matrices and Interval Graphs*, Pacific Journal of Mathematics 15 (1965), pp.835-855.

[5] M. Garey, D. S. Johnson, *Computers and Intractability*, Freeman and Company, 1979.

[6] F. Gavril, *The Intersection Graphs of Subtrees in Trees Are Exactly the Chordal Graphs*, Journal of Combinatorial Theory Series B, vol. 16(1974), pp. 47-56.

[7] P. Gilmore, A. Hoffman, *A Characterization of Cocomparability Graphs and of Interval Graphs*, Canadian Journal of Mathematics 16 (1964), pp. 539-548.

[8] D. Helmbold, E. Mayr, *Two Processor Scheduling is in NC*, in *VLSI Algorithms and Architectures* (F. Makedon et al. ed.), LNCS 227 (1986), pp. 12-15.

[9] C. Lekkerkerker, J. Boland, *Representation of a Finite Graph by a Set of Intervals on the Real Line*, Fundamenta Mathematicae 51.

[10] K. Mulmuley, U. Vazirani, V. Vazirani, *Matching is as easy as matrix inversion*, Combinatorica 7 (1987), pp. 105-113.