Annals of Operations Research 138, 179–187, 2005

Packing r-Cliques in Weighted Chordal Graphs

P. HELL pavol@cs.sfu.ca
School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

S. KLEIN sula@cos.ufrj.br
IM and COPPE/Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, CEP 21945-970, Rio de Janeiro, RJ, Brazil

L.T. NOGUEIRA loana@cos.ufrj.br
COPPE/Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, CEP 21945-970, Rio de Janeiro, RJ, Brazil

F. PROTTI fabiop@cos.ufrj.br
IM and NCE, Universidade Federal do Rio de Janeiro, Caixa Postal 2324, CEP 20001-970, Rio de Janeiro, RJ, Brazil

Abstract. In Hell et al. (2004), we have previously observed that, in a chordal graph G, the maximum number of independent r-cliques (i.e., of vertex disjoint subgraphs of G, each isomorphic to K_r, with no edges joining any two of the subgraphs) equals the minimum number of cliques of G that meet all the r-cliques of G. When $r = 1$, this says that chordal graphs have independence number equal to the clique covering number. When $r = 2$, this is equivalent to a result of Cameron (1989), and it implies a well known forbidden subgraph characterization of split graphs. In this paper we consider a weighted version of the above min-max optimization problem. Given a chordal graph G, with a nonnegative weight for each r-clique in G, we seek a set of independent r-cliques with maximum total weight. We present two algorithms to solve this problem, based on the principle of complementary slackness. The first one operates on a graph derived from G, and is an adaptation of an algorithm of Farber (1982). The second one improves the performance by reducing the number of constraints of the linear programs. Both results produce a min-max relation. When the algorithms are specialized to the situation in which all the r-cliques have the same weight, we simplify the algorithms reported in Hell et al. (2004), although these simpler algorithms are not as efficient. As a byproduct, we also obtain new elementary proofs of the above min-max result.

Keywords: chordal graphs, min-max theorems, greedy algorithms, linear programming duality, complementary slackness

Introduction

In Hell et al. (2004) we studied the problem of partitioning a given chordal graph into k independent sets and l cliques. Graphs which admit such a partition have been called (k, l)-graphs, e.g., Brandstädt (1996) and Feder et al. (1999). (A clique is a complete subgraph, not necessarily maximal.) The case of (1, 1)-graphs is best known–they are called split graphs, a subclass of chordal graphs, and can be efficiently recognized, e.g.,

Foldes and Hammer (1977), Golumbic (1980), and Hell et al. (2004). On the other hand, $(k, 0)$-graphs are precisely the k-colourable graphs, thus they are not necessarily chordal. The recognition of $(k, 0)$-graphs is NP-complete in general, but polynomial time solvable for chordal graphs. This lead us to seek algorithms for the recognition of chordal (k, l)-graphs. We obtained in Hell et al. (2004) an $O(n(m + n))$ algorithm for this problem ($O(m + n)$ in case $k \leq 1$), and the following forbidden subgraph characterization of chordal (k, l)-graphs:

Theorem 1 (Hell et al., 2004)**.** A chordal graph is a (k, l)-graph if and only if it contains no induced subgraph isomorphic to $(l + 1)K_{k+1}$.

The graph $(l + 1)K_{k+1}$ consists of $l + 1$ disjoint copies of K_{k+1} with no edges between the copies.

In the process of proving this theorem, we observed that to partition a graph G into k independent sets and l cliques amounts to finding l cliques whose removal leaves a k-colourable subgraph, which, in the case of chordal graphs, means a subgraph without K_{k+1}. Thus if we let $g(G, r)$ denote the minimum number of cliques which meet all r-cliques of G, then a chordal graph G is a (k, l)-graph if and only if $g(G, k+1) \leq l$. Let $f(G, r)$ denote the maximum number of independent r-cliques in G. Note that $f(G, r)$ is always at most $g(G, r)$, and that $f(G, k + 1) \leq l$ if and only if G contains no induced $(l + 1)K_{k+1}$. Thus an equivalent way to state the above theorem is the following:

Theorem 2 (Hell et al., 2004)**.** For any chordal graph G and positive integer r, we have $g(G, r) = f(G, r)$.

In words, in a chordal graph, the maximum number of independent r-cliques equals the minimum number of cliques that meet all r-cliques. Our algorithms for recognizing chordal (k, l)-graphs can also be used to find both a maximum set of independent r-cliques, and a minimum set of cliques that meet all r-cliques.

In this paper we consider a weighted version of the above optimization problem. Given a chordal graph G, with a nonnegative weight for each r-clique in G, we seek in G a set of independent r-cliques with maximum total weight.

We present two algorithms based on complementary slackness, which solve the problem (and the corresponding duals). The first one is an adaptation of an algorithm of Farber (1982). The second one improves the efficiency by reducing the number of constraints of the linear programs. When the algorithms are specialized to the situation in which all the r-cliques have the same weight, we simplify the proofs and algorithms reported in Hell et al. (2004), although these simpler algorithms are not as efficient.

The goal of the original set packing problem, see Karp (1972), is to find the maximum number of mutually disjoint sets of a given collection C. We remark that many interesting packing problems may arise by considering other types of relationships involving the sets (here, for instance, we replace "mutually disjoint" by "independent", where C is the collection of r-cliques of a chordal graph G).

1. The derived graph

Let G be a graph, and r a positive integer. We define the derived graph $\mathcal{K}^r(G)$ as follows: the vertices of $\mathcal{K}^r(G)$ are the r-cliques of G, and two such vertices are adjacent in $\mathcal{K}^r(G)$ whenever the two r-cliques have an edge joining them. (Note that, according to this definition, two distinct intersecting r-cliques in G are adjacent in $\mathcal{K}^r(G)$.) Clearly, the size of $\mathcal{K}^r(G)$ is bounded by $O(n^r)$.

In Hell et al. (2004) we proved that, if G is chordal then, for each positive integer r, the derived graph $\mathcal{K}^r(G)$ is also chordal. Since we are going to solve our problem on G by manipulating $\mathcal{K}^r(G)$, we will give a more effective proof of this fact, by explanining how to obtain a perfect elimination ordering of $\mathcal{K}^r(G)$ from a perfect elimination ordering of G.

Thus consider a fixed chordal graph G, with a perfect elimination ordering $1, 2, \ldots, n$ of its vertices, and fix a positive integer r. (Recall that, in a perfect elimination ordering, the neighborhood of vertex i in the subgraph induced by $i+1, \ldots, n$ is a clique, for every $i = 1, \ldots, n$; see Golumbic (1980).) Let R be the vertex set of the derived graph $\mathcal{K}^r(G)$, i.e., the set of all subgraphs of G isomorphic to K_r. Elements of R are totally ordered by the following lexicographic order: for $S, S' \in R$, we let $S < S'$ if $\max S < \max S'$, or if $\max S = \max S' = m_1$ and $\max(S - m_1) < \max(S' - m_1)$, or if $\max S = \max S' = m_1$ and $\max(S-m_1) = \max(S'-m_1) = m_2$ and $\max(S-m_1-m_2) < \max(S' - m_1 - m_2)$, etc. In other words, $S < S'$ if and only if $s < s'$, where s is the largest element of $S - S'$ and s' the largest element of $S' - S$.

Lemma 2.1. Suppose S, S' are two adjacent vertices of $\mathcal{K}^r(G)$, and $S < S'$. Then $maxS$ is adjacent to some element of S' in G.

Proof. Assume that $\max S$ is not in S', for otherwise there is nothing to prove. According to the definition of $\mathcal{K}^r(G)$, we know that some vertex $a \in S$ is adjacent to some vertex $b \in S'$. We may assume that $a < \max S$, for otherwise we are done. If also $a < b$, then $\max S$ and b are adjacent by the properties of a perfect elimination ordering. Otherwise we have $b < a < \max S \leq \max S'$. It again follows from the properties of a perfect elimination ordering that a is adjacent to $\max S'$. If $\max S = \max S'$, then $\max S$ is adjacent to b. Thus we may assume that $\max S < \max S'$, and hence $\max S$ is adjacent to $\max S'$ by a last application of the properties of a perfect elimination ordering. □

Theorem 3. The total order $<$ on R is a perfect elimination ordering on $\mathcal{K}^r(G)$.

Proof. Suppose that S is adjacent in $\mathcal{K}^r(G)$ to both S' and S'', and $S < S'$, $S < S''$. We shall prove that S' is adjacent to S''. Since $S < S'$ and $S < S''$, then, by Lemma 2.1, $\max S$ is adjacent to some vertex, say a, of S', and adjacent to some vertex, say b, of S''. We may assume that $\max S$ does not belong to $S' \cup S''$, and that S' and S'' are disjoint, for otherwise we are done. In particular, $\max S$, a and b are distinct. In the remainder of this proof, let us denote the perfect elimination ordering on G by $<_G$ to avoid ambiguity. If $\max S <_G a$ and $\max S <_G b$ then, by the usual property of a perfect elimination ordering, a and b are

adjacent. Thus, without loss of generality, we assume that $a <_G \max S$. Therefore, $\max S$ is adjacent to $\max S'$ by the property of a perfect elimination ordering. If $\max S <_G b$ then, again, by the property of perfect elimination ordering, $\max S$ is adjacent to $\max S''$. Otherwise, we must have $\max S > b$, and similarly $\max S$ and $\max S''$ are adjacent. The conclusion follows by a last application of the property of perfect elimination ordering to the vertices $\max S$, $\max S'$ and $\max S''$. □

With these results, we may assume that we have access to a perfect elimination ordering of $\mathcal{K}^r(G)$. The weights on the r-cliques in G induce weights on the vertices of $\mathcal{K}^r(G)$, and finding a maximum weight independent set of r-cliques in G corresponds exactly to finding a maximum weight independent set of vertices in $\mathcal{K}^r(G)$. The algorithms of Frank (1976) and Farber (1982) can be used for this purpose. In what follows we are going to write $u \sim v$ to denote that u and v are equal or adjacent, and we are going to write $u \lesssim v$ ($u \gtrsim v$) if $u \sim v$ and $u \leq v$ ($u \sim v$ and $u \geq v$). To apply these algorithms to $\mathcal{K}^r(G)$, we state the following two linear programs:

$$P: \text{maximize} \sum_{S \in R} w_S x_S$$

$$\text{subject to} \sum_{S \lesssim S'} x_{S'} \leq 1, \ \forall S \in R, \tag{1}$$

$$x_S \geq 0, \ \forall S \in R. \tag{2}$$

The dual D is stated as follows:

$$D: \text{minimize} \sum_{S \in R} y_S$$

$$\text{subject to} \sum_{S' \lesssim S} y_{S'} \geq w_S, \ \forall S \in R, \tag{3}$$

$$y_S \geq 0, \ \forall S \in R. \tag{4}$$

The algorithm proceeds as follows:

Algorithm 1

- **Initialization:**
  For all $S \in R$ do

  $$y_S := 0 \quad \text{and} \quad x_S := 2.$$

- **First Step:**
  For all $S \in R$, in the perfect elimination ordering, do

  $$\text{If } w_S > \sum_{S' \lesssim S} y_{S'} \text{ then}$$

  $$y_S := w_S - \textstyle\sum_{S' \lesssim S} y_{S'}.$$

- **Second Step:**
  For $S \in R$, in the reverse perfect elimination ordering, do

$$\begin{aligned}&\text{If } x_S = 2 \text{ and } y_S > 0 \text{ then}\\ &\qquad x_S := 1\\ &\qquad \text{For each } S' \text{ adjacent to } S \text{ do } x_{S'} := 0.\end{aligned}$$

As explained in Farber (1982), Algorithm 1 clearly computes feasible integer solutions for P and D. In addition, these solutions have the same value. By the principle of complementary slackness, P and D have integer optima.

The r-cliques can be generated during the initialization step by traversing a perfect elimination ordering $v_1, \ldots, v_n$ of G as follows. Assume that the neighborhood of each v_i is an adjacency list $v_{i_1}, \ldots, v_{i_{k_i}}$ where $i < i_1 < i_2 < \cdots < i_{k_i}$ (note that only neighbors with indices greater than i are stored). If v_1 has at least r neighbors, let $p_1 < \cdots < p_r$ be r indices ("pointers") such that $p_1 = 1$ and $p_2, \ldots, p_r$ are the indices of the first $r-1$ neighbors of v_1 in its adjacency list. Clearly, $v_{p_1}, \ldots, v_{p_r}$ is an r-clique $S \in R$. By moving the pointers to the right conveniently, one at a time, we obtain the subset of r-cliques contaning v_1, according to the ordering discussed in Theorem 3. We proceed analogously for v_2, and so on. The running time to generate all the r-cliques during the initialization step is $O(n' + m')$, where n' and m' are, respectively, the number of vertices and edges of $\mathcal{K}^r(G)$.

Step 1 can be done in time $O(n' + m')$, since each pair S', S satisfying $S' \lesssim S$ is considered only once, when computing $\sum_{S' \lesssim S} y_{S'}$.

Clearly, Step 2 can also be done in time $O(n'+m')$. Therefore, the overall complexity of Algorithm 1 is $O(n' + m')$, linear on the size of $\mathcal{K}^r(G)$.

2. A more efficient algorithm

We can save some of the work of the above algorithm by simplifying the linear programs P, D as follows:

$$P' : \text{maximize} \quad \sum_{S \in R} w_S x_S$$

$$\text{subject to} \sum_{j \gtrsim i,\ j \in S} x_S \leq 1, \quad \forall i = 1, 2, \ldots, n, \tag{5}$$

$$x_S \geq 0, \quad \forall S \in R. \tag{6}$$

Note that P' has, in general, fewer constraints than P, since we have now one constraint for each vertex of G, rather than for each r-clique of G. Nevertheless, any integer solution x_S, $S \in R$, has the chosen r-cliques (those $S \in R$ for which $x_S = 1$) independent, since no maximal clique of G contains vertices of more than one such S.

Thus P' computes the fractional relaxation of P.

The new dual now has fewer variables:

$$D': \text{ minimize } \sum_{1\le i\le n} y_i$$

$$\text{subject to } \sum_{i\lesssim j,\ j\in S} y_i \ge w_S, \quad \forall S \in R, \tag{7}$$

$$y_i \ge 0, \quad \forall i = 1, 2, \ldots, n. \tag{8}$$

Observe that constraint (2) was replaced by constraint (5), which still guarantees that if two r-cliques are adjacent then only one of them can be chosen. In order to check this, consider the cliques $F_i = G[\{j \mid j \gtrsim i\}]$, $i = 1, \ldots, n$. (The notation $G[X]$ stands for the subgraph induced by the subset of vertices X.) It is clear that any two adjacent r-cliques are intersected by a same F_i. Constraint (5) therefore says that we can select at most one r-clique from the collection of mutually adjacent r-cliques intersected by F_i.

A careful analysis of the previous algorithm reveals that we can replace all variables y_S with the same $\max S$ by one variable y_i, where $i = \max S$, by performing the following algorithm:

Algorithm 2

- **Initialization:**
  For all $S \in R$ do $x_S := 2$
  For all $i = 1, \ldots, n$ do $y_i := 0$
- **First Step:**
  For all $i = 1, \ldots, n$ do

  $$\text{Let } M_i := \max_{S\in R,\ i=\max S} \left\{ w_S - \sum_{j\ne i \text{ and } j\lesssim k,\ k\in S} y_j \right\}$$

  If $M_i > 0$ then $y_i := M_i$
- **Second Step:**
  For $i = n, \ldots, 1$ do
  If $i = \max S$, $x_S = 2$ and $y_i > 0$ then
  $x_S := 1$, where S is a r-clique which caused setting $y_i > 0$
  For each S' adjacent to S do $x_{S'} := 0$.

Again, Algorithm 2 computes feasible integer solutions for P' and D', both having the same value. Thus P' and D' have integer optima.

Recall that, in Algorithm 1, we use the fact that if G is chordal, then $\mathcal{K}^r(G)$ is also chordal. But Algorithm 1 also works for a graph G (possibly not chordal) such that $\mathcal{K}^r(G)$ is chordal, as long as we have a perfect elimination ordering of $\mathcal{K}^r(G)$.

2.1. Implications for the unweighted problem

In this section we propose a new algorithm for computing $f(G, r)$ and $g(G, r)$, based on the programs P' and D'. The algorithm is not as efficient as the one given in Hell et al. (2004), but it is simpler to describe, and yields a new proof of Theorem 2.

It is easy to see that when all the weights of the r-cliques are 1, (5) and (6) compute the relaxation of $f(G, r)$, and (7) and (8) compute the relaxation of $g(G, r)$. Thus Theorem 2 follows from the principle of complementary slackness. To make this subsection independent (and to provide an elementary and selfcontained proof of Theorem) we give a direct proof.

The algorithm below computes both $f(G, r)$ and $g(G, r)$ for a fixed integer $r \geq 1$ and any chordal graph G. This is accomplished by constructing a set of s cliques that cover all r-cliques and a set of s r-cliques that are independent. Since $f(G, r) \leq g(G, r)$, this proves the result.

The cliques that cover all r-cliques are going to be of the type $F_i = G[\{j \mid i \lesssim j\}]$. Thus we will be choosing a set A of vertices of G so that the collection of cliques F_i, $i \in A$, will cover all r-cliques. We will also be selecting a set B of r-cliques which are independent. The algorithm is presented below.

Algorithm 3

Assume G is a chordal graph with a perfect elimination ordering $1, 2, \ldots, n$.

Initialize $A = \emptyset$.

In the order $i = 1, 2, \ldots, n$,

- if $i = \max S$ for some $S \in R$ such that A does not contain any vertex $j \lesssim k$, for some vertex $k \in S$, then
  - add i to A, and
  - add S to B.

    (If there are several possible S we choose any one such S).

For $r = 1$, Algorithm 3 bears some resemblance with Gavril's algorithm to find both a maximum independent set and a minimum clique covering in a chordal graph. (see Golumbic (1980), pp. 99–100).

Lemma 3.1. The cliques F_i, $i \in A$, cover all r-cliques of G, and the r-cliques in the set B are independent. Moreover, $|A| = |B|$.

Proof. It is clear that $|A| = |B|$, and that the cliques F_i corresponding to A cover all r-cliques in G. Thus it remains to verify only that the r-cliques in B are independent. Suppose that some S which was selectedis adjacent to some S' selected previously. Let $s = \max S$ and $s' = \max S'$. Thus both s and s' are in A, and $s' < s$. We now show,

using the properties of perfect elimination ordering, that if any $j' \in S'$ is adjacent to any $j \in S$, then $F_{s'}$ contains a vertex of S, contradicting the fact that$s$ was chosen for A.

We have to consider the following cases:

Case I. $j > s'$.

In this case both j and s' are in $F_{j'}$ and hence must be adjacent. Thus $F_{s'}$ contains the vertex j of S.

Case II. $j' < j < s'$.

It is still the case that both j and s' are in $F_{j'}$ and hence are adjacent. However, in this case F_j contains s' instead of the situation in the above case. Nevertheless, F_j also contains s and so $F_{s'}$ now contains $s \in S$.

Case III. $j < j'$.

Vertices j' and s are adjacent, since F_j contains both. Thus vertices s and s' are adjacent, as $F_{j'}$ contains both. Therefore $F_{s'}$ contains $s \in S$. □

Theorem 2 follows directly from Lemma 3.1.

Acknowledgments

We are grateful to Tomás Feder for helpful discussions concerning the complexity of the algorithms presented in this work. We also thank the referees for valuable suggestions that improved this work.

References

Brandstädt, A. (1996). "Partitions of Graphs Into One or Two Independent Sets and Cliques." *Discrete Mathematics* 152, 47–54.

Brandstädt, A. (1998). "The Complexity of Some Problems Related to Graph 3-Colorability." *Discrete Applied Mathematics* 89, 59–73.

Cameron, K. (1989). "Induced Matchings." *Discrete Applied Mathematics* 24, 97–102.

Farber, M. (1982). "Applications of Linear Programming Duality to Problems Involving Independence and Domination." Ph.D. Thesis, Rutgers University.

Feder, T., P. Hell, S. Klein, and R. Motwani. (1999). "Complexity of Graph Partition Problems." In F.W. Thatcher and R. E. Miller (eds.), *Proceedings of the 31st Annual ACM Symposium on Theory of Computing—STOC'99*. New York: Plenum Press, pp. 464–472.

Foldes, S. and P. Hammer. (1977). "Split Graphs." F. Hoffman et al. (eds.). *Proc. 8th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Louisiana State Univ., Baton Rouge, Louisiana, pp. 311–315.

Frank, A. (1976). "Some Polynomial Algorithms for Certain Graphs and Hypergraphs." In *Proceeding 5th British Combin. Conf. Congressus Numerantium* No. XV, Utilitas Math., Winnipeg.

Garey, M.R., D.S. Johnson, and L. Stockmeyer. (1976). "Some Simplified NP-Complete Graph Problems." *Theoretical Computer Science* 1, 237–267.

Golumbic, M.C. (1980). *Algorithmic Graph Theory and Perfect Graphs.* New York: Academic Press.

Hell, P., S. Klein, L.T. Nogueira, and F. Protti. (2004). "Partitioning Chordal Graphs into Independent Sets and Cliques." *Discrete Applied Mathematics* 141, 185–194.

Karp, R.M. (1972). "Reducibility Among Combinatorial Problems." In R.E. Milner and J.W. Thatcher (eds.), *Complexity of Computer Computations*. New York: Plenum Press, pp. 85–103.

Nogueira, L.T. (1999). "Grafos Split e Grafos Split Generalizados." Master Thesis, COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Brazil, (In Portuguese).