Tendências em Matemática Aplicada e Computacional, **3**, No. 1 (2002), 147-155.
© *Uma Publicação da Sociedade Brasileira de Matemática Aplicada e Computacional.*

Particionamento de Grafos Cordais em Conjuntos Independentes e Cliques

P. HELL[1], School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada, V5A1S6.

S. KLEIN[2], IM e COPPE/Sistemas, Universidade Federal do Rio de Janeiro, RJ, 21945-970, Brasil.

L. T. NOGUEIRA[3], COPPE/Sistemas, Universidade Federal do Rio de Janeiro, RJ, 21945-970, Brasil.

F. PROTTI[4], NCE, Universidade Federal do Rio de Janeiro, RJ, 20001-970, Brasil.

Resumo. Neste trabalho consideramos a seguinte generalização de grafos split: Um grafo G é um *grafo-*(k,l) se seu conjunto de vértices pode ser particionado em k conjuntos independentes e l cliques (uma clique é um subgrafo completo não necessariamente maximal). Portanto, os grafos-(k,l) são uma generalização dos grafos split, que correspondem aos grafos-$(1,1)$. Grafos split podem ser eficientemente reconhecidos [4]. Além disso, os problemas clássicos de otimização combinatória são também resolvidos eficientemente nessa classe. Na verdade, nosso resultado principal é uma caracterização de grafos cordais-(k,l). Também apresentamos um algoritmo para reconhecer grafos cordais-(k,l) cuja complexidade é $O(n(m+n))$. Quando $k=l=1$, nosso algoritmo possui complexidade $O(m+n)$.

Em particular, obtivemos um algoritmo mais simples e eficiente apra reconhecer grafos split, do qual torna-se fácil derivar a conhecida caracterização de grafos split por subgrafos proibidos.

1. Introdução

Um grafo G é um *grafo-*(k,l) [1] se seus vértices podem ser particionados em k conjuntos independentes e l cliques. (Uma clique é um subgrafo completo não necessariamente maximal.) Portanto, os grafos-(k,l) são uma generalização dos grafos split [7], que correspondem aos grafos-$(1,1)$. Grafos split podem ser eficientemente

[1]pavol@cs.sfu.ca
[2]sula@cos.ufrj.br
[3]loana@cos.ufrj.br
[4]fabiop@cos.ufrj.br

reconhecidos [7]. Além disso, os problemas clássicos de otimização combinatória são também resolvidos eficientemente nessa classe. Quando $k > 1$ ou $l > 1$, os grafos-(k,l) não são em geral cordais. (Um grafo é dito *cordal* se não possui C_k, um ciclo de k vértices sem cordas, para $k \geq 4$, como subgrafo induzido.) Em [2], algoritmos polinomiais de reconhecimento para as classes $(2,1)$, $(1,2)$ e $(2,2)$ foram propostos. Feder *et al.* [3] também propuseram algoritmos polinomiais de reconhecimento para essas classes, que surgiram como sub-produto de algoritmos de partição em subgrafos densos e esparsos. Por outro lado, sabe-se que reconhecer grafos-(k,l) para $k \geq 3$ ou $l \geq 3$ é NP-completo [1]. (Por exemplo, a classe dos grafos-$(k,0)$ corresponde ao problema de reconhecer se um dado grafo é k-colorível.).

Neste trabalho apresentamos uma caracterização para a classe dos grafos cordais-(k,l), assim como um algoritmo polinomial para o reconhecimento desta classe de grafos. Mais especificamente, provamos que um grafo cordal é um grafo-(k,l) se, e somente se, não contém $l+1$ cópias independentes de K_{k+1}. (Um conjunto de subgrafos é independente se eles são disjuntos e dois a dois não adjacentes, i.e., não ligados por nenhuma aresta. A notação K_r designa uma clique com r vértices.)

Um caso especial deste resultado para grafos cordais-$(2,1)$ foi inicialmente tratado em [5].

Uma forma alternativa de enunciar nosso resultado baseia-se no seguinte fato: o número máximo de $K_r's$ independentes num grafo cordal é igual ao número mínimo de cliques necessárias para tocar todos os $K_r's$ de G. Em outras palavras, se denotarmos por $f(G,r)$ o número máximo de cópias independentes de $K_r's$ em G, e por $g(G,r)$ o número mínimo de cliques que interceptam todos os $K_r's$ de G, então podemos mostrar que para um grafo cordal $f(G,r) = g(G,r)$. (Observe que quando $r = 1$, $f(G,r)$ é o número de independência de G, e $g(G,r)$ é o número de cobertura por cliques de G.). Nosso algoritmo de tempo $O(n(m+n))$ identifica $f(G,r)$ cópias independentes de $K_r's$ e o mesmo número de cliques que interceptam todos os $K_r's$. Nosso algoritmo de reconhecimento na verdade encontra o menor valor de l para o qual G é um grafo-(k,l). O algoritmo se torna mais eficiente quando $k = 1$, i.e., quando procuramos uma partição do grafo em *um* conjunto independente e um conjunto de cliques. Quando ambos k e l são iguais a um, especializamos o algoritmo para gerar um algoritmo mais simples e eficiente para grafos split. (Observe que neste caso nós não precisamos da restrição de grafo cordais.)

Seja G um grafo. Se $S, S' \subseteq V(G)$, denotamos por $N_S(S')$ a vizinhança de S' em S, i.e., o conjunto de vértices de S que estão em S' ou que são adjacentes a algum vértice de S'. Além disso, se $N_S(S') \neq \emptyset$, então nós dizemos que S e S' são adjacentes.

Escreveremos $N_S(v)$ ao invés de $N_S(\{v\})$, observe que esta *vizinhança* de v em S contém v se $v \in S$.

2. Teoremas

Nesta seção apresentamos nossa caracterização de grafos cordais-(k,l) em termos de subgrafos proibidos. Os seguintes lemas serão úteis:

Lema 1. *Sejam C e C' duas cliques (adjacentes ou não) num grafo cordal G. Então algum vértice de C' é adjacente a todos os vértices de $N_C(C')$.*

Prova. Devemos provar que, na verdade, os vizinhos de C' em C são linearmente ordenados por inclusão. Suponha que dois vértices distintos $v_1, v_2 \in C'$ tem vizinhos incomparáveis em C, i.e., nenhum dos conjuntos $N_C(v_1)$, $N_C(v_2)$ contém o outro. Então existem dois vértices distintos $u_1, u_2 \in C$ tal que u_1 é adjacente a v_1 mas não o é a v_2, e u_2 é adjacente a v_2 mas não o é a v_1. Isto é impossível, já que u_1, u_2, v_2, v_1 induziriam um ciclo de tamanho 4 sem corda, como mostra a Figura 1. O lema segue considerando o vértice $v \in C'$ com $N_C(v)$ máximo.

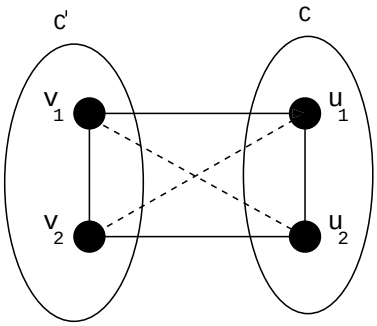

Figura 1: Exemplo para o Lema 1

Lema 2. *Sejam C e K duas cliques disjuntas (adjacentes ou não) de um grafo cordal G. Então existe uma clique C' com a seguinte propriedade: C' intercepta K, e também intercepta todas as cliques adjacentes a K que são interceptadas por C.*

Prova. Seja $L = N_C(K)$. Pelo Lema 1 algum vértice de K é adjacente a todos os vértices de L, e portanto pode ser adicionado a L para obtermos uma clique que intercepta K, assim como todas as cliques de G interceptadas por L. Considere agora uma clique K' de G que intercepta C mas é disjunta de L. Segue, da definição de L, que tal clique não intercepta K. Precisamos considerar tal K' se ele contiver um vértice a adjacente a algum vértice de K. Denote por A o conjunto de tais vértices, i.e., vértices que são adjacentes a K e pertencem a alguma clique que intercepta C mas não L. Afirmamos que cada $a \in A$ é adjacente a todos os vértices de L. Na verdade, se $b \in L$ não é adjacente a a, então existem vértices $c \in C \setminus L$, e $s, t \in K$ (possivelmente $s = t$) tal que , b, c, a, s, t, b é um ciclo sem corda. Similarmente, provamos que A é uma clique, i.e., para quaisquer $a, a' \in A$, a e a' são adjacentes em G. Caso contrário, existem vértices $c, c' \in C \setminus L$ (possivelmente $c = c'$), e $s, s' \in K$ (possivelmente $s = s'$), tal que $a, c, c', a', s', s'a$ é um ciclo sem corda. Logo, o conjunto $L \cup A$ induz uma clique, e Lema 1 garante que existe um vértice $u \in K$ adjacente a todos os vértices de $L \cup A$. Agora, a clique induzida por $L \cup A \cup u$ intercepta K e, por definição de A, intercepta, também, todas as cliques interceptadas por V que são adjacentes a K.

Observe que o lema acima também é válido quando C e K não são disjuntos (com $C' = C$).

Lema 3. *Sejam $C_1, C_2, \ldots, C_p$ uma coleção de cliques duas a duas adjacentes num grafo cordal G. Então existe uma clique C em G que intercepta cada $C_i, i = 1, 2, \ldots, p$.*

Prova. O resultado segue facilmente quando $p \leq 2$. Assuma agora que $p > 2$. Por indução, existe uma clique C que intercepta C_i para cada $i \in \{1, \ldots, p-1\}$. Se C intercepta C_p, nada resta provar. Caso contrário, aplique o Lema 2 a C e C_p.

Uma condição simples e necessária para que um grafo G seja um grafo-(k, l) é que ele não contenha $l + 1$ cópias independentes de K_{k+1}. Na verdade, considere qualquer partição de G em k conjuntos independentes e l cliques. Qualquer K_{k+1} em G teria de conter um vértice de uma das cliques na partição, e portanto, dentre as $(l + 1)$ cópias dos K_{k+1}, duas delas deveriam interceptar a mesma clique, e logo possuiriam uma aresta ligando-as (o que significa que não eram independentes.) Verifica-se que para grafos cordais a condição acima é também suficiente. (Observe que a condição diz apenas que $(l + 1)K_{k+1}$ não é um subgrafo induzido de G.) Derivamos este fato do seguinte resultado:

Teorema 1. *Seja G um grafo cordal, e seja $r \geq 1$ um inteiro. Então $f(G, r) = g(G, r)$.*

É claro que $f(G, r) \leq g(G, r)$ para qualquer G e qualquer $r \geq 1$. Para provarmos a igualdade para grafos cordais, vamos proceder como segue.

Seja G um grafo. Vamos definir $K^r(G)$ como o grafo com um vértice correspondendo a cada K_r em G, e dois vértices adjacentes em $K^r(G)$ se, e somente se, os K_r's correspondentes não são independentes em G.

Lema 4. *Para qualquer grafo G, $f(G, r)$ é o número de independência de $K^r(G)$. Para um grafo cordal G, $g(G, r)$ é o número de cobertura por cliques de $K^r(G)$.*

Prova. A primeira sentença é óbvia. A segunda segue da observação de que podemos modificar qualquer cobertura por clique C de $K^r(G)$, para construir uma coleção de (com o mesmo número) cliques que interceptam todos os K_r's de G, aplicando o Lema 3 a cada clique em C.

Lema 5. *Se G é cordal então $K^r(G)$ também o é.*

Prova. Assuma que $W_1, W_2, \ldots, W_q, W_1$, $(q \geq 4)$ é um ciclo sem cordas em $K^r(G)$. Isto significa que W_i e W_j são consecutivos no ciclo se, e somente se, os K_r's correspondentes em G são adjacentes. Considere uma sequência S de vértices de G, $S = (u_1, w_1, u_2, w_2, \ldots, u_q, w_q)$ (índices são tomados circularmente no intervalo $1 \ldots q$). É claro que se i e j não são índices consecutivos, então os subconjuntos $\{u_i, w_i\}$ e $\{u_j, w_j\}$ não são adjacentes. Ocasionalmente, pode ocorrer $u_i = w_i$ ou $w_i = u_{i+1}$ para algum i, mas estas igualdades não podem ocorrer simultaneamente.

Isto significa que todo vértice ocorrendo em S aparece no máximo duas vezes, e duas ocorrências de um mesmo vértice usam, necessariamente, posições consecutivas em S. Estas observações mostram que podemos construir um ciclo C_0 em G a partir de S removendo ocorrências repetidas de vértices. Esta construção assegura que pelo menos um vértice de $\{u_i, w_i\}$ é escolhido para todo $i \in \{1, \ldots, q\}$. Logo, C_0 contém pelo menos 4 vértices. Além disso, C_0 é claramente um ciclo sem corda, uma contradição.

O Teorema 1 segue naturalmente dos Lemas 4 e 5.

Prova do Teorema 1. Pelo Lema 5 $K^r(G)$ é cordal, e logo perfeito. Portanto, o número de independência de $K^r(G)$ é igual a número de cobertura por cliques. O Lema 4 completa a prova.

A caracterização de grafos cordais-(k, l) por subgrafos proibidos segue como uma consequência do teorema anterior.

Teorema 2. *Um grafo cordal é um grafo-(k, l) se, e somente se, não contém $(l + 1)K_{k+1}$ como um subgrafo proibido.*

Prova. Mostramos que um grafo cordal-(k, l) não pode conter $l + 1$ cópias independentes de K_{k+1}, i.e., não pode conter $(l + 1)K_{k+1}$ como um subgrafo proibido. Por outro lado, o Teorema 1 implica que se um grafo cordal G não contém $l + 1$ cópias independentes de K_{k+1}, então $g(G, k + 1) \leq l$. Isto significa que G contém l cliques cuja remoção deixa um subgrafo G' sem K_{k+1}. Como G é perfeito, G' é K-colorível, portanto G admite uma partição em k conjuntos independentes e l cliques.

3. Os Algoritmos

Como k e l são fixos, existe um número polinomial de subgrafos de G com $(l + 1)(k + 1)$ vértices, e portanto o Teorema 1 nos fornece um algoritmo polinomial para reconhecer grafos cordais-(k, l). Existem, contudo, algoritmos mais eficientes. O algoritmo que apresentamos aqui possui complexidade $O(n(m+n))$ e nos fornece uma segunda prova para o Teorema 1.

Inicialmente revisamos o algoritmo padrão de coloração para grafos cordais. (Observe que testar a existência de uma k-coloração é equivalente a reconhecer grafos-$(k, 0)$.) Suponha que os vértices de G obedecem a um esquema de eliminação perfeita (EEP) $1, 2, \ldots, n$ [7]. O Algoritmo Guloso Reverso procede na ordem $n, n - 1, \ldots, 1$, associando a cada vértice a menor cor disponível. Em outra palavras, para colorir G com as cores $s_1, s_2, \ldots$, colorimos o vértice n com a cor s_1, e tendo colorido os vértices $n, n - 1, \ldots, i + 1$, colorimos o vértice i com a cor s_d onde d é o menor índice tal que nenhum vizinho de i dentre $i + 1, i + 2, \ldots, n$ foi colorido com a cor s_d. Observe que até este ponto, i faz parte de um K_d, já que i tem um vizinho de cada uma das cores $s_1, s_2, \ldots, s_{d-1}$, que são mutualmente adjacentes. (Quaisquer

dois vizinhos de i dentre $i+1, i+2, \ldots, n$ são adjacentes, já que $1, 2, \ldots, n$ é um EEP.) Segue que o Algoritmo Guloso Reverso fornece, em tempo $O(m+n)$, uma coloração mínima e uma clique máxima.

Estamos prontos para descrever nosso algoritmo. Seja $k \geq 0$ um inteiro. O algoritmo encontra o menor valor de l (possivelmente $l = 0$) para o qual G é um grafo-(k, l). Estaremos colorindo os vértices do grafo cordal de entrada G com as cores $s_1, s_2, \ldots, s_k$ e $c_1, c_2, \ldots, c_l$. Através da execução do algoritmo, os vértices coloridos com a cor s_d formarão um conjunto independente, e os vértices coloridos com a cor c_a formarão uma clique. Denotaremos por S_i o conjunto que consiste de i, e mais todos os vértices dentre $1, 2, \ldots, i-1$ coloridos com $s_1, s_2, \ldots, s_k$.

O seguinte fato é facilmente obtido das definições, usando as propriedades do esquema de eliminação perfeita:

Lema 6. *Se o vértice i é adjacente ao primeiro vértice j colorido com a cor c_a e $j < i$, então i é adjacente a todos os vértices $x < i$ coloridos com a cor c_a.*

Este Lema nos permitirá facilmente testar se um dado vértice i pode ser adicionado ou não a clique formada pelos vértices coloridos com a cor c_a.

Algoritmo para Reconhecer Grafos Cordais-(k, l)

Assuma que G é um grafo cordal com um esquema eliminação perfeita $1, 2, \ldots, n$.

- Colora o vértice 1 com a cor s_1
- Tendo colorido os vértices $1, 2, \ldots, i-1$ sem usar a cor c_1:
  - remova as cores de $1, 2, \ldots, i-1$ e colora $1, 2, \ldots, i$ com as cores $s_1, s_2, \ldots, s_k$ (usando o Algoritmo Guloso Reverso), se possível, ou
  - mantenha a coloração dos vértices $1, 2, \ldots, i-1$, e colora i com a cor c_1.
- Tendo colorido os vértices $1, 2, \ldots, i-1$ e tendo usado as cores $c_1, c_2, \ldots, c_a$:
  - colora i com a cor c_b, onde $b \leq a$ é o menor índice tal que i é adjacente ao primeiro vértice colorido com a cor c_b, se tal índice existe, ou
  - remova as cores dos vértices de $S_i \setminus i$ e colora S_i com as cores $s_1, s_2, \ldots, s_k$ (usando o Algoritmo Guloso Reverso), se possível, ou
  - mantenha a coloração de $1, 2, \ldots, i-1$, e colora i com a cor c_{a+1}.

No algoritmo acima, l é o maior valor de a tal que existe um vértice colorido c_a, ou $l = 0$ se todos os vértices estão coloridos com as cores $s_1, s_2, \ldots, s_k$.

Como o funcionamento do algoritmo é dominado por no máximo n aplicações do Algoritmo Guloso Reverso, segue que o tempo de execução é $O(n(m+n))$.

Proposição 1. *Se o algoritmo usa a cor c_p, então G contém um pK_{k+1} como subgrafo induzido.*

Prova. Seja v_a o primeiro vértice (no EEP) colorido com a cor c_a. O subgrafo de G induzido por S_{v_a-1} é k-colorível, mas nosso algoritmo achou impossível adicionar v_a de forma que S_{v_a} continuasse k-colorível. Logo existe um subgrafo isomorfo a K_{k+1} contendo v_a e alguns k vértices de S_{v_a-1}. Resta apenas mostrar que os subgrafos $X_1, X_2, \ldots, X_p$ são independentes. Suponha que um vértice x do subgrafo X_a seja adjacente ou igual a um vértice x' de um subgrafo $X_{a'}$. Assuma que $a < a'$.

Se $x' \leq x$, então devido ao fato de que x' é adjacente ou igual a $v_{a'}$, concluímos que x é adjacente ou igual a $v_{a'}$. Agora v_a e $v_{a'}$ devem ser adjacentes, já que $x \leq v_a$ e $x \leq v_{a'}$. Isto significa que $v_{a'}$ é adjacente ao primeiro vértice colorido por c_a contradizendo o fato de que nosso algoritmo pode colorir $v_{a'}$ com a cor c_a.

Se $x' > x$, então x é adjacente ou igual a v_a por um argumento similar. Se $x' \leq v_a$ então $v_{a'}$ e v_a devem ser adjacentes e $v_{a'}$ deveria ter sido colorido com a cor c_a, como no caso anterior. Por outro lado, se $x' > v_a$, então x' é adjacente ao primeiro vértice colorido com a cor c_a. Portanto, nosso algoritmo deveria ter colorido x' com a cor c_a, contradizendo o fato dele ter sido colorido com alguma cor c_d (no caso de x ser um elemento de $S_{v_{d'}-1}$) ou com a cor $c_{d'}$ (no caso $x' = v_{d'}$.)

Corolário 1. *As seguintes afirmações são equivalentes:*

1 - O algoritmo particiona G em k conjuntos independentes e l cliques;
2 - O grafo G é um grafo-(k, l);
3 - O grafo G não contém $(l+1)K_{k+1}$ como subgrafo induzido.

Prova: As implicações $1 \Rightarrow 2$ e $2 \Rightarrow 3$ são óbvias, e a Proposição 1 prova que $3 \Rightarrow 1$.

Observe que a equivalência entre 1 e 2 prova a corretude do algoritmo, enquanto que a equivalência entre 1 e 3 nos fornece uma segunda prova do Teorema 2.

Encerramos esta seção observando que o algoritmo encontra, para qualquer k e para qualquer grafo cordal G, o menor valor l tal que G é um grafo-(k, l).

4. O Caso de um Conjunto Independente, Enfatizando Grafos Split

Quando $k = 1$, podemos de alguma forma simplificar o algoritmo, já que não precisamos do Algoritmo Guloso Reverso para testar se um vértice pode ser adicionado a um conjunto independente, mantendo a propriedade de independência.

Algoritmo para Reconhecer Grafos Cordais-$(1, l)$

Assuma que G é um grafo cordal com um EEP $1, 2, \ldots, n$.

- Colora o vértice 1 com a cor s_1,
- e continue colorindo os vértices $i = 2, 3, \ldots$ com a cor $s_1$enquanto for possível (i não tem aresta para $1, 2, \ldots, i-1$),
- e então colora o primeiro j que não pode ser colorido com a cor c_1.
- Tendo colorido os vértices $1, 2, \ldots, i-1$ usando as cores $s_1, c_1, c_2, \ldots, c_a$,
  - colora i com a cor c_b, onde $b \leq a$ é o primeiro índice tal que i é adjacente ao primeiro vértice colorido com a cor c_b, se tal índice existir, ou
  - colora i com a cor s_1 se i é não adjacente a todos os vértices coloridos com a cor s_1, ou
  - colora i com a cor c_{a+1}.

É claro que este algoritmo pode ser implementado em tempo $O(m+n)$.

A situação é mais simples quando $k = l = 1$, e neste caso não precisamos explicitamente assumir cordalidade. (Grafos split são automaticamente cordais.) Como grafo split são de grande interesse [7, 4], nós alteramos o algoritmo uma vez mais, de forma a reconhecer grafos split:

Algoritmo para Reconhecer Grafos Split

Seja G um grafo qualquer.

- Encontre um EEP $1, 2, \ldots, n$ de G [7].
- Colora 1 com a cor s, e continue colorindo $i = 2, 3, \ldots$ com a cor s enquanto for possível (i é não adjacente aos vértices já coloridos), então colora o próximo vértice j com a cor c.
- Se todos os vértices $1, 2, \ldots, i-1$ tiverem sido coloridos, e ambas as cores s e c já tiverem sido usadas, então colora i com a cor c se for adjacente ao primeiro vértice j colorido com a cor c; caso contrário, colora i com a cor s se for não adjacente a todos os vértices já coloridos com a cor s.

Se o algoritmo falhar devido a não existência de um EEP, então o algoritmo dado em [7] exibe um subgrafo induzido isomorfo a um C_4, C_5, ou C_k, $k \geq 6$. Se ele falhar colorindo todos os vértices, então de acordo com a Proposição 1, G contém um subgrafo induzido isomorfo a um $2K_2$. Apresentamos abaixo um pequena versão da prova, que será usada numa generalização mais abaixo.

Se, num certo momento, um vértice i não pode ser colorido com a cor s ou com a cor c, então ele é não adjacente ao primeiro vértice j colorido com a cor c, e é

adjacente a algum vértice k que foi anteriormente colorido com a cor s. Afirmamos que j e k não podem ser adjacentes. Se $k < j$, segue das propriedades de EEP. Se $k > j$, então se k fosse adjacente a j o algoritmo teria colorido com a cor c.

O vértice j foi colorido com a cor c, porque ele era adjacente a um vértice l previamente colorido com a cor s. Como $l < j < i$, e i, j são não adjacentes, l deve ser não adjacente a i. Além disso, os vértices k, l são não adjacentes, já que eles foram ambos coloridos com a cor s. Portanto, i, j, k, l formam um subgrafo induzido isomorfo a um $2K_2$ em G.

Como cada $C_k, k \geq 6$ também contém um subgrafo induzido isomorfo a um $2K_2$, obtivemos a seguinte caracterização já conhecida de grafos split [7]:

Corolário 2. *Um grafo G é um grafo split se, e somente se, não contém $2K_2$, C_4 ou C_5 como subgrafos induzidos.*

Referências

[1] A. Brandstädt, Partitions of graphs into one or two independent sets and cliques, *Discrete Mathematics*, **152** (1996), 47-54.

[2] A. Brandstädt, The complexity of some problems related to graph 3-colorability, *Discrete Applied Mathematics*, **89** (1998), 59-73.

[3] T. Feder, P. Hell, S. Klein, and R. Motwani, Complexity of graph partition problems, em "The 31st Annual ACM Symposium on Theory of Computing - STOC'99" (F. W. Thatcher and R. E. Miller, eds.), pp. 464-472, Plenum Press, New York, 1999.

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

[5] L. T. Nogueira, " Grafos Split e Grafos Split Generalizados", Tese de Mestrado, COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil, 1999.

[6] M. R. Garey, D. S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science*, **1** (1976), 237-267.

[7] M. C. Golumbic, " Algorithmic Graph Theory and Perfect Graphs", Academic Press, New York, 1980.