

# O Problema da T-Coloração: Modelo e Algoritmo

*Nair Maria Maia de Abreu\**  
*Pedro Soares da Silva Neto\*\**

O Problema de Alocação de Canais de Rádio visando a minimizar interferência entre transmissores a serem instalados numa dada região, vem sendo tratado como uma generalização do Problema de Coloração de vértices em grafos, conhecido como T-coloração.

Neste trabalho apresentamos um algoritmo para o problema de Coloração de vértices com desempenho superior ao de Welsh e Powell e compatível com o do algoritmo Cosine. Além disso, apresentamos um modelo para o problema de T-coloração de modo a melhor atender as restrições mais freqüentes impostas na prática pela alocação de canais de rádio. Um algoritmo baseado na modelagem proposta é apresentado com razão de performance média superior ao do guloso de Cozzens e Robert (1982) para T-coloração e desempenho computacional compatível.

---

\* Engenheira Química, Mestre em Ciências e Doutorada em Química.

\*\* Graduado em Bacharelado e Licenciatura em Química; Mestre em Ciência em Síntese Orgânica.

---

Palavras-chave: T-Coloração; Alocação de Canais de Rádio; Algoritmos Heurísticos; Otimização Combinatória.

## O PROBLEMA DE T-COLORAÇÃO

O Problema de T-coloração surgiu em conexão com o Problema de Alocação de Canais de Rádio, onde dados  $n$  transmissores  $x_i$ ,  $i = 1, \dots, n$ , em uma região, deseja-se alocar a cada transmissor  $x_i$ , uma freqüência  $A(x_i)$  sob a qual o transmissor  $x_i$  deve operar sem interferência. Assumimos que  $A(x_i)$  é um inteiro positivo.

Um grafo  $G = (V, E, T)$  para  $V = \{ x_i \mid x_i \text{ é um transmissor de rádio} \}$ ,  $E = \{ (x_i, x_j) \mid x_i, x_j \in V, \text{ e se interferem} \}$  e  $T \subset \mathbb{Z}^+$ , cujos elementos representam separações de canais não permitidas a serem alocados a transmissores que se interferem modela o problema.

A equação

$$(x_i, y_j) \in E \Rightarrow |A(x_i) - A(x_j)| \notin T \quad (1)$$

expressa a restrição imposta por T e a função  $A: V \rightarrow Z_{0+}$  satisfazendo a equação (1) é chamada **T-coloração**.

Assumimos que  $0 \in T$  e a solução do problema para  $T=\{0\}$  resulta na atribuição de canais distintos a transmissores adjacentes. É o caso do Problema de Alocação de Co-canal e coincide com a Coloração de Vértices de um Grafo G. Para  $T = \{0, 1\}$ , os transmissores não podem receber frequências iguais ou adjacentes, senão a equação (1) deixa de ser respeitada. O conjunto T foi introduzido no modelo para limitar ou eliminar possíveis interferências entre os transmissores pelo afastamento dos canais (frequências) a serem alocados aos mesmos.

As principais medidas de eficiência para a T-coloração são dadas pela **ordem**  $|A(x_i)|$  que é o número de diferentes cores alocadas, e pelo **span**  $\max |A(x_i) - A(x_j)|_i$ , a diferença entre a maior e a menor cor alocadas, o que representa a largura do espectro de frequências ocupado. O que se deseja

é determinar a menor ordem e o menor *span* de uma T-coloração, conhecidos, respectivamente, por **número T-cromático**  $\chi_t(G)$  e **número T-span**  $sp_t(G)$ .

Em geral, os problemas para obter  $\chi_t$  e  $sp_t$  são NP-completo, pois para  $T = \{0\}$  o problema de T-coloração é exatamente o problema de coloração. Geralmente, os problemas são árduos, mas um grande progresso tem sido obtido em problemas sobre certas classes de grafos e classes específicas de conjuntos T. Para maiores detalhes ver Cozzens e Roberts (1982).

Um outro aspecto importante relaciona-se ao fato de que nem sempre se obtém *span* ótimo quando a alocação é de ordem ótima e vice-versa. Na figura 1 ilustramos tal fato com o grafo  $C_5$  e  $T=\{0, 1, 4, 5\}$ . Em 1.a. temos uma alocação de ordem mínima com as cores  $\{1, 3, 7\}$ ; em 1b temos uma alocação *span* mínimo, com as cores  $\{1, 2, 3, 4, 5\}$ .

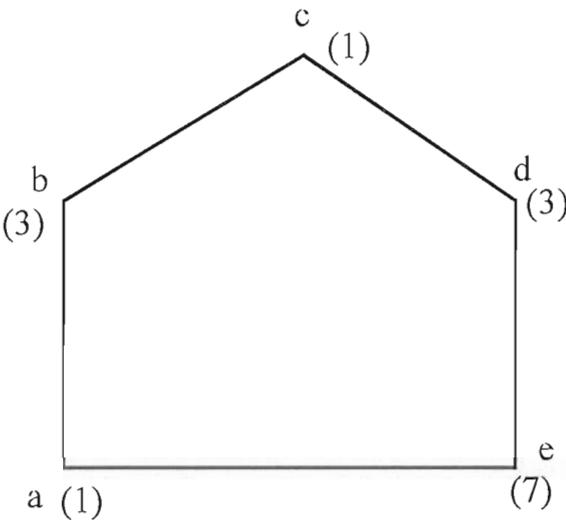


FIGURA 1.a.: ordem mínima

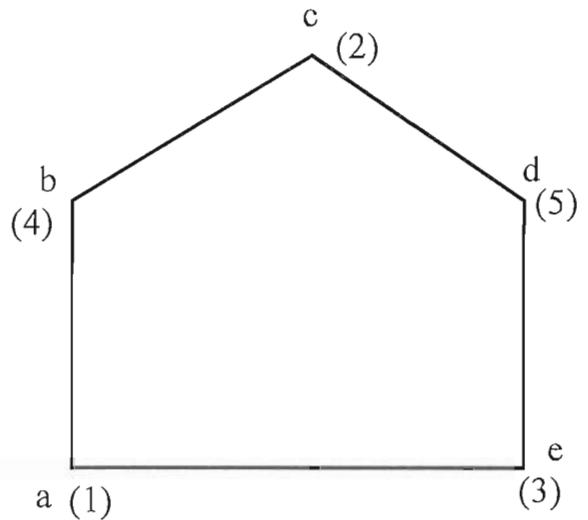


FIGURA 1.b.: span mínimo

## UM NOVO MODELO

O modelo apresentado anteriormente não atende os requisitos de um típico Problema de Alocação de Canal, pois apesar de considerar transmissores como vértices unidos por arestas quando há potencial de interferência entre eles, não leva em consideração diferentes níveis de interferência entre cada par de transmissores. Assim, o uso de apenas um conjunto  $T$  para todo o grafo nos impõe um único nível de interferência para todos os pares de transmissores, o que não corresponde à realidade. A literatura utiliza o conceito de grafos aninhados para contornar essa questão o que dificulta a resolução do problema que agora requer a necessidade de se obter uma  $T$ -coloração para  $k$  grafos distintos e compatíveis entre si.

Propomos, então, um modelo mais abrangente que utiliza um único grafo, onde cada aresta recebe um rótulo  $k$  associado a um conjunto  $T_k$  de inteiros não-negativos que representam separações de canais não-permitidas para alocações a transmissores que interferem entre si. Neste caso  $\mathcal{S}$  passa a ser uma classe de conjunto  $\mathcal{S} = \{T_1, T_2,$

$\dots, T_t\}$  e o grafo  $G = (V, E, \mathcal{S})$  um  $\mathcal{S}$ -grafo, onde  $V = \{x_1, \dots, x_n\}$  representa o conjunto de transmissores.  $E$  um conjunto de arestas rotuladas  $[(x_i, x_j), k]$ ,  $x_i$  e  $x_j \in V$  e  $k$  é o rótulo da aresta  $(x_i, x_j)$  correspondente a  $T_k \in T$ . A equação a seguir resume a restrição imposta por  $\mathcal{S}$ :

$$[(x_i, x_j), k] \in E \Rightarrow |A(x_i) - A(x_j)| \notin T_k, 1 \leq k \leq t \quad (2)$$

A função  $A: V \rightarrow Z_{0+}$  satisfazendo a equação (2) é aqui chamada uma  **$\mathcal{S}$ -coloração generalizada**.

Seja  $G = (V, E, \mathcal{S})$  um grafo, onde  $V = \{1, 2, 3, 4, 5\}$ ;  $E = \{[(1,2),0], [(1,3),0], [(1,4),0], [(2,3),3], [(2,4),3], [(2,5),2], [(3,4),4], [(3,5),1], [(4,5),1]\}$  e  $\mathcal{S} = \{T_k \mid k = 0, \dots, t\}$ , tal que  $T_0 = \{0\}$ ,  $T_1 = \{0, 15\}$ ,  $T_2 = \{0, 7, 14, 15\}$ ,  $T_3 = \{0, 1, 7, 14, 15\}$ ,  $T_4 = \{0, 1, 2, 3, 4, 5, 7, 8, 14, 15\}$ .

O  $\mathcal{S}$ -grafo é apresentado na figura 2. Ao lado de cada vértice está o respectivo rótulo e no meio de cada aresta o rótulo que identifica o conjunto  $T_k$  associado. Encontra-se entre parênteses, ao lado de cada vértice, os valores obtidos numa  $\mathcal{S}$ -coloração de *span* mínimo.

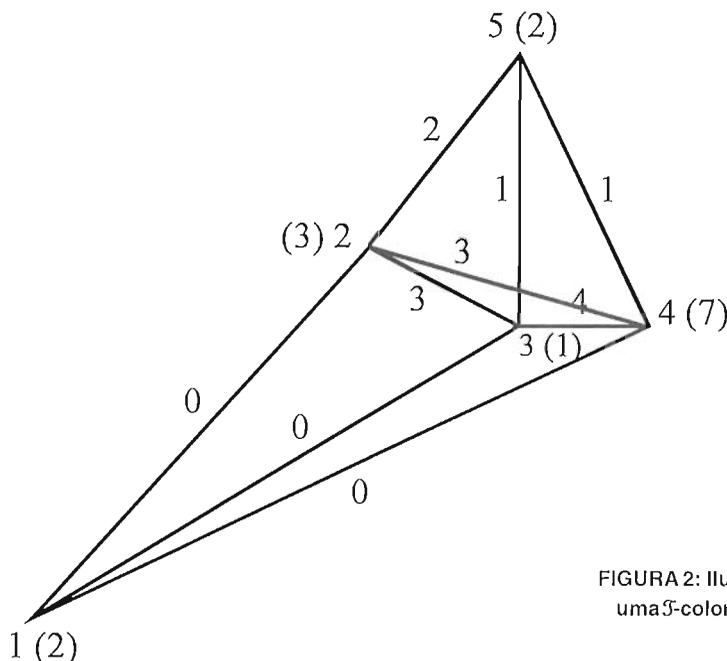


FIGURA 2: Ilustração do  $\mathcal{S}$ -grafo e uma  $\mathcal{S}$ -coloração generalizada

## O ALGORITMO COLORCLIQUE

O algoritmo mais explorado na literatura para resolver o Problema da T-coloração é um do tipo guloso, que pode ser encontrado em Cozzens e Roberts (1982) que trabalha sobre um grafo simples e um conjunto T comum a todas as arestas do grafo. Ao adaptarmos o referido algoritmo para trabalhar sobre o nosso modelo, não ficamos satisfeitos com sua razão de performance média e nos dirigimos em busca de um algoritmo melhor. Assim, surgiu o algoritmo ColorClique que no item posterior será adaptado à resolução do Problema de T-coloração. Agora trataremos do problema simples de coloração em grafos.

Seja  $G(V, E)$  um grafo simples não-orientado. Uma  $k$ -coloração de  $G$  é um mapeamento  $c: V \rightarrow \{1, 2, \dots, k\}$ , onde  $k$  é o número de cores usadas para colorir o grafo  $G$  e  $c$  é uma  $k$ -coloração própria, se  $c(u) \neq c(v)$ , para todas arestas  $\{u, v\} \in E$ . O número cromático de  $G$ , denotado por  $\chi(G)$ , é o menor inteiro positivo  $k$  para o qual existe uma  $k$ -coloração própria. O

problema de coloração de um grafo arbitrário é um problema de decisão NP-completo Garey e Johnson (1979).

Baseado no procedimento Clique de Turner (1988), mostrado a seguir, o algoritmo colore a clique obtida com cardinalidade  $k$ , e assume dispor pelo menos dessas  $k$  cores para colorir os demais vértices de  $G$ . A cada vértice não colorido, associa-se o conjunto de cores disponíveis  $\{1, \dots, k\}$ . E para aqueles adjacentes a um dado vértice  $u$  da clique colorida, o conjunto correspondente de cores disponíveis é atualizado. O passo seguinte consiste em colorir todos os demais vértices não coloridos, usando como critério de seleção aquele com o menor número de cores disponíveis. Se não houver mais cores disponíveis, a cor  $k+1$  é usada para colori-lo introduzindo-a em cada conjunto de cores disponíveis dos vértices não coloridos e não adjacentes a  $u$ . Caso contrário, atribui-se a  $u$ , a menor cor disponível, atualizando-se o conjunto de cores disponíveis correspondentes aos seus adjacentes. O algoritmo encerra quando todos os vértices estiverem coloridos.

### O Algoritmo ColorClique

Dado o grafo  $G = (V, E)$

$K := \text{Clique}(G)$ ; {Encontrar a Clique  $K$  pelo procedimento de Turner com  $k=|K|$ }

Inicializar os conjuntos CoresDisponíveis( $v$ ) com  $\{1, \dots, k\}$ ,  $\forall v \in V$ .

Colorir a cada vértice  $v$  da Clique  $K$  com uma das  $k$  cores, excluindo a cor( $v$ ) dos conjuntos CoresDisponíveis(Adj( $v$ )).

Enquanto houver  $v \in V$  não colorido faça

    Procurar  $u \in V$  é tal que  $|\text{CoresDisponíveis}(u)| = \text{Mín} \{|\text{CoresDisponíveis}(v)|, v \in V \text{ e } v \text{ não colorido}\}$

    Se  $\text{CoresDisponíveis}(u) = \emptyset$   
    então

$k := k+1$       {inserir mais uma cor}

$\text{CoresDisponíveis}(v) := \text{CoresDisponíveis} \cup \{k\}$ , tq  $v \in V$  e  $v \notin \text{Adj}(u)$

    senão

$c := \text{Mín} \{ \text{CoresDisponíveis}(u) \}$

$\text{cor}(u) := c$  {colorir o vértice  $u$  com a cor  $c$ }

$\text{CoresDisponíveis}(v) := \text{CoresDisponíveis}(v) \setminus \{c\}$ ,  $v \in \text{Adj}(u)$

fim enquanto

retornar  $k$  e cor      { número de cores e coloração obtida }

Apresentamos abaixo a função Clique proposta por Turner (1988) a qual retoma uma clique do grafo, não necessariamente maximal, mas segundo Turner, na maioria das vezes maximal.

função Clique( $G(V,E)$ ) {de Turner}

var  $K, S$ : conjunto de vértices;

$S := V$ ;  $K := \phi$ ;

enquanto  $S \neq \phi$  faça

  selecionar aleatoriamente  $x \in S$  de maior grau;

$K := K \cup \{x\}$ ;

$S := S \cap \text{Adj}(x)$ ;

clique :=  $K$ ;

fim

## OUTRAS CONSIDERAÇÕES

O algoritmo ColorClique usa a estratégia de escolher o vértice com menor número de cores disponíveis para coloração, o correspondente ao vértice com o maior grau de saturação no algoritmo de Brélaz (1979).

O ColorClique foi testado em grafos aleatórios e, para efeitos de comparação, aplicamos os algoritmos Guloso e Cosine; este, apresentado por Hertz (1991) nos mesmos grafos. Os resultados obtidos para 30 instâncias geradas aleatoriamente com distribuição uniforme, com  $n$  vértices,  $20 \leq n \leq 100$ , e probabilidade 0.5 de ocorrência de aresta entre pares de vértices, é mostrado na tabela 1.

Vértices	Guloso	ColorClique	Cosine
10	4.1333	3.9000	3.9667
20	6.9000	5.9000	6.1333
30	9.2000	7.6667	7.8667
40	11.0667	9.3000	9.4667
50	13.1333	11.3000	10.7667
60	14.8000	12.6333	12.2000
70	16.5667	14.4667	13.6667
80	18.0333	15.6000	15.0000
90	19.6000	17.2000	16.3667
100	21.5667	18.4667	17.6000

TABELA 1: Média do número de cores obtidas

Como podemos observar na tabela 1, o número de cores médio obtido pelo ColorClique é muito próximo do obtido pelo Cosine. Para  $n > 30$ , o Cosine passa a ser ligeiramente superior, mas mantendo a proximidade do número de cores médio obtido pelo ColorClique.

Avaliamos também o número médio de cores obtido pelo ColorClique frente ao Cosine para grafos aleatórios com 100 vértices, e  $0,1 \leq p \leq 0,9$ , cujos resultados estão na tabela 2. Como pode ser observado, o ColorClique mantém praticamente a mesma razão de performance do Cosine nas diversas densidades.

p	ColorClique	Cosine
0.10	5.8000	5.5000
0.20	8.7667	8.5000
0.30	11.866	11.3667
0.40	14.9667	14.2000
0.50	18.5000	17.4333
0.60	22.1333	21.3000
0.70	27.1000	25.9667
0.80	33.1333	31.9667
10.90	43.0333	41.5000

TABELA 2: Média de cores obtidas por Cosine e ColorClique variando p

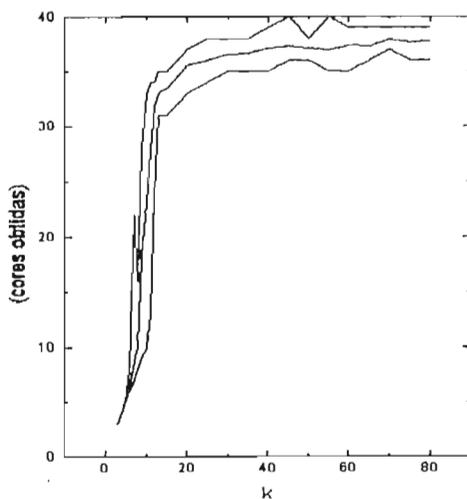


FIGURA 3: Cores obtidas com o algoritmo ColorClique em grafos K-coloríveis

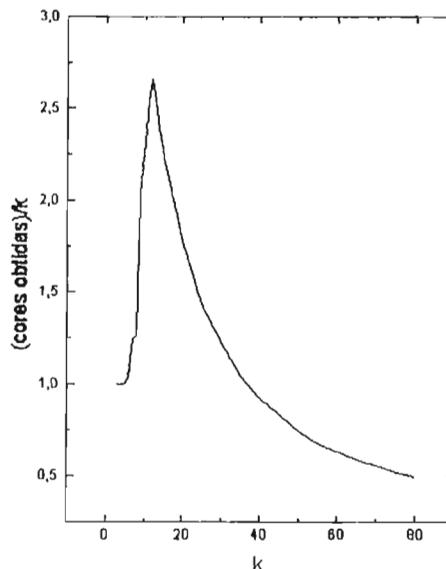


FIGURA 4: Média da razão [(cores obtidas)/k] pelo ColorClique em grafos K-coloríveis

Na figura 3 mostramos o número de cores obtidas (mínimo, médio, máximo) pelo algoritmo ColorClique sobre grafos K-coloríveis  $X_{255}(k,0.5)$ , com 255 vértices,  $k$  cores e probabilidade de geração de arestas 0,5. Na figura 4, mostramos os mesmos resultados na forma de média da razão entre cores obtidas e  $k$ . Observamos que o algoritmo ColorClique sobre os grafos K-coloríveis utilizados estabiliza o número de cores obtidas para valores em torno de 37 em média quando  $k$  cresce. Baseado no estudo de Turner (1988), este algoritmo apresenta excelente desempenho para valores de  $k \leq 10g_2n$ .

O ColorClique tem complexidade  $O(n^2)$ , conforme apresentado em Silva Neto(1995), superando o Cosine  $O(k.n^2)$ . No entanto, o Cosine tende a melhorar a sua eficiência para grafos mais densos, aproximando-se do desempenho do ColorClique. Isto se justifica, pois o Cosine trabalha com contração de vértices e quanto mais

denso for o grafo, menos operações de contração são requeridas para se chegar a um clique, condição de parada do algoritmo.

### ALGORITMO $\mathcal{F}$ -COLORCLIQUE

Baseado no algoritmo ColorClique, desenvolvemos o  $\mathcal{F}$ -ColorClique para resolver o problema realmente proposto. Este algoritmo procura uma clique K com o procedimento **clique** de Turner (1988) e inicia a T-Coloração, pelos vértices da clique. Como  $k$  cores normalmente não são suficientes para T-colorir e a quantidade de cores não é facilmente determinada previamente, preferimos usar como parâmetro de entrada do algoritmo o maior número de cores a serem usadas (MaxCor). Todos os vértices do grafo possuem MaxCor cores disponíveis.

### O algoritmo T-ColorClique

Dados o  $\mathcal{F}$ -grafo  $G(V,E,\mathcal{F})$ ;  $\{ O(t.n^2)$

T-Coloração - vetor  $[1..n]$  para armazenar a  $\mathcal{F}$ -coloração, onde  $n=|V|$

MaxCor, a maior cor disponível no conjunto  $\{1, \dots, \text{MaxCor}\}$  de cores

```

início
K:= Clique(G); { maior clique, procedimento de Turner –O(m + n) }
k:= |K|;
CoresDisponiveis(u):= { 1..MaxCor },  $\forall u \in K$ 
CoresUsadas:=  $\emptyset$ ;
CoresNaoUsadas:= { 1..MaxCor };
Para u  $\in K$  faça { colorir a clique K }
  início
  Se não T-colorir(u,G,T-Coloração, CoresUsadas, MaxCor) então
    Encerrar o algoritmo com falha, MaxCor pequeno;
  AtualizarCoresDisponiveis(u,G,T-CoresDisponiveis,MaxCor);
  fim
nc:=k+1; { controla o numero de vértices coloridos }
enquanto nc  $\leq$  n faça
  início
  u:= vértice não colorido com menor conjunto CoresDisponiveis
  Se não T-colorir(u,G,cores,CoresUsadas,MaxCor) então
    Encerrar o algoritmo com falha; { MaxCor pequeno }
  AtualizarCoresDisponiveis(u,G,CoresDisponiveis,MaxCor);
  nc:= nc + 1;
  fim { enquanto
fim- { ColorClique }

```

A função **T-Colorir** atribui a menor cor disponível e já usada ao vértice **u**. Esta estratégia evita a introdução de uma nova cor enquanto houver cor usada que atenda. Caso contrário, a procura é estendida às demais cores até “MaxCor”.

```

Função T-Colorir(u:vertice;Tcoloração:veto):lógico; { O(MaxCor)
início
Se CoresDisponiveis(u) =  $\emptyset$  então
  Retornar T-Colorir:=Falso;
cor:= Min { CoresDisponiveis(u)  $\cap$  CoresUsadas }
Se cor < 1 { nao T-coloriu } então
  início
  cor:= Min { CoresDisponiveis(u)  $\cap$  CoresNaoUsadas };
  CoresUsadas:= CoresUsadas  $\cup$  { cor };
  CoresNaoUsadas :=CoresNao Usadas  $\setminus$  { cor };
  fim
T-Coloração(u) := cor
T-colorir:= Verdadeiro;
fim;

```

A função **AtualizarCoresDisponíveis** atualiza os conjuntos **CoresDisponíveis** dos vértices não  $\mathcal{F}$ -coloridos e adjacentes a **u**, já  $\mathcal{F}$ -colorido, retirando-se as cores proibidas impostas pelo conjunto  $T_i$  associado à aresta  $(u, v, i)$ .

```

Função AtualizarCoresDisponíveis(u,G,CoresDisponíveis);
{ atualiza os conjuntos de cores disponíveis Adjacentes a u }
início
Para todo  $v \in \text{Adj}(u)$  faça
    início
    Para todo  $t \in T_i$  associado à aresta  $(u,v,i)$  faça
        início
         $\text{CorProibida} := T\text{-coloração}(u) + t$ 
        Se  $\text{CorProibida} \in \text{CoresDisponíveis}(v)$  então
             $\text{CoresDisponíveis}(v) := \text{CoresDisponíveis}(v) \setminus \{\text{CorProibida}\}$ ;
         $\text{CorProibida} := T\text{-coloração}(u) - t$ 
        Se  $\text{CorProibida} \in \text{CoresDisponíveis}(v)$  então
             $\text{CoresDisponíveis}(v) := \text{CoresDisponíveis}(v) \setminus \{\text{CorProibida}\}$ ;
        fim;
    fim;
fim;

```

Para o grafo  $G(V, E, T)$  da figura 5, consideramos  $T(0) = \{0\}$ ,  $T(1) = \{0, 1\}$  e  $\text{MaxCores} = 5$ . Deseja-se obter uma  $\mathcal{F}$ -coloração para  $G$  aplicando-se o algoritmo  $\beta$ -ColorClique.

A figura 6 ilustra os passos do algoritmo. Inicialmente, o procedimento *Clique* retorna  $K$ , pelos vértices  $\{1, 3, 2\}$ ; cada vértice de  $G$  recebe um conjunto de cores disponíveis  $\{1, 2, 3, 4, 5\}$  e a  $\mathcal{F}$ -coloração de  $K$  é iniciada pelo vértice 1 que recebe a cor 1, a menor dentre as disponíveis; a seguir, os vértices 2 e 3 perdem a cor 1 de seus respectivos conjuntos **CORESDISPONÍVEIS** e o vértice 4 perde as cores 1 e 2; então o vértice 3, escolhido pela ordem obtida em  $K$ , recebe a cor 2 e atualizando-se o conjunto **CORESDISPONÍVEIS** do vértice 2, são retiradas as cores 2 e 3, sobram então  $\{4,5\}$ ; do vértice 4 é retirada a cor 3, RESTANDO-SE  $\{4, 5\}$ ; o vértice 2 recebe a cor 4 e não possui adjacentes não coloridos a serem atualizados;  $\mathcal{F}$ -colorindo o restante do grafo, o vértice 4 recebe a cor 4 e não havendo mais vértices, o algoritmo termina.

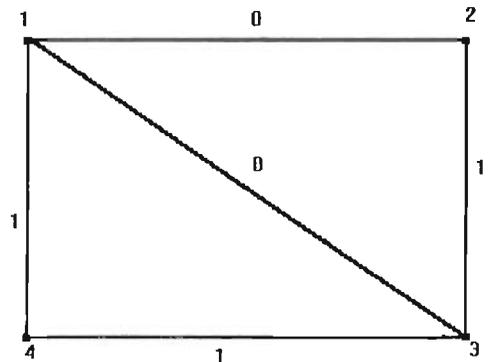


FIGURA 5: T-grafo  $G(V, E, T)$  usado pelo exemplo 5

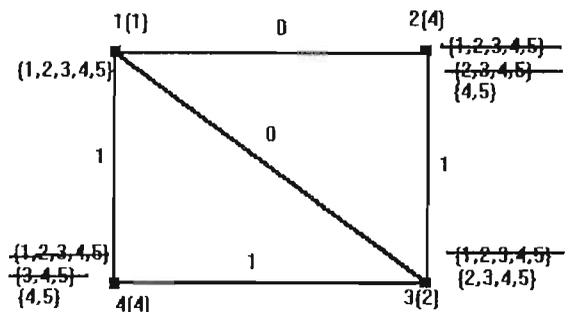


FIGURA 6: Ilustração das etapas de  $\mathcal{F}$ -coloração do exemplo 5

A complexidade do algoritmo  $\mathcal{F}$ -ColorClique é dominada pelas tarefas *AtualizarCores-Disponíveis* que requer  $O(m.t)$ , onde  $m = |E|$  e  $t = \text{Max}\{T\}$ , e *Colorir* que requer  $O(n.\text{MaxCor})$ . Assim o algoritmo  $\mathcal{F}$ -ColorClique terá complexidade  $O(t.n^2)$ . Para maiores detalhes, confira em Silva Neto (1995).

## CONCLUSÕES

Apresentamos um modelo para o problema de Alocação de Canais de Rádio por meio de  $\mathcal{F}$ -grafos. Este modelo se mostra mais adequado

aos problemas práticos, pois considera os diferentes níveis de interferência entre os diferentes pares de transmissores pela associação de conjuntos distintos às arestas do grafo proposto.

Para resolver o problema, propusemos o algoritmo  $\mathcal{F}$ -ColorClique de complexidade  $O(t.n^2)$ , para  $t = \text{Max}\{|T_j|$ , baseado neste novo modelo e, no algoritmo ColorClique que também apresentamos. Este último trata-se de uma heurística polinomial de  $O(n^2)$  para o clássico problema de Coloração de Vértices de um grafo simples  $G$ , mostrando-se mais adequado à adaptação ao caso mais geral da T-coloração do que o Cosine proposto por Hertz (1991).  $\square$

## BIBLIOGRAFIA

- COZZENS, M. B. & ROBERTS, F. S., T-Colorings of graph and the channel assignement problem, Congr. Numer. 35, 1982, 191-208.
- BRÉLAZ, D., New methods to color the vertices of a graph, Comm. ACM, 22, 1979, 251-256.
- GAREY, M. R. & JOHNSON, D. S., Computers and intractability: A Guide to the Theory of NP-completeness, W. H. Freeman and Co, 1979.
- BERTZ, A., Cosine: A new graph coloring algorithm, Operations Research Letters 10, 1991, 411-415.
- SILVA NETO, P. S., Problema de Alocação de Canal e T-Coloração, Tese de Mestrado, 1995.
- TURNER, J. S., Almost All k-Colorable Graphs Are Easy to Color, Journal of Algorithms, 9, 1988, 63-82.