

Algoritmos aproximados para o Problema do Maior Conjunto Controlado Generalizado

Ivairton Monteiro Santos*, Carlos Alberto Martinhon, Luiz Satoru Ochi

Universidade Federal Fluminense, Departamento de Ciência da Computação
Rua Passo da Pátria, 156 - Bloco E - 3 andar - Boa Viagem
24210-240, Niterói, RJ, Brasil
isantos@ic.uff.br, {mart, satoru}@dcc.ic.uff.br

Resumo Dado um grafo $G = (V, E)$ e um conjunto de vértices $M \subseteq V$, sendo $|V| = n$, dizemos que o vértice $v \in V$ é *controlado* por M se a maioria dos seus vértices adjacentes (incluindo ele mesmo) pertencem ao conjunto M . O conjunto M define um *monopólio* em G se M controla todos os vértices de V . Dado um conjunto $M \subseteq V$ e dois grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$, onde $E_1 \subseteq E_2$, temos o PROBLEMA DE VERIFICAÇÃO DE MONOPÓLIO - PVM, que consiste em encontrar um grafo sanduíche $G = (V, E)$ (i.e., um grafo onde $E_1 \subseteq E \subseteq E_2$), tal que M defina um monopólio em G . Caso a resposta do PVM seja “NÃO”, temos então o PROBLEMA DO MAIOR CONJUNTO CONTROLADO - PMCC, cujo objetivo é encontrar um grafo sanduíche $G = (V, E)$, tal que o número de vértices de G controlados por M seja maximizado. O PVM pode ser resolvido em tempo polinomial, o PMCC, entretanto, só será resolvido em tempo polinomial se $P = NP$. Neste trabalho apresentamos a noção de *f-control* e introduzimos ainda o PROBLEMA DO MAIOR CONJUNTO CONTROLADO GENERALIZADO (PMCCG), que associa pesos e folgas mínimas a cada vértice de V . Nesse caso, o objetivo será maximizar o somatório dos pesos dos vértices *f-controlados* por M . Apresentamos um algoritmo 0,5-aproximado para o PMCCG e um procedimento para a geração de soluções viáveis baseado na solução de uma relaxação linear para o problema. Essas soluções são utilizadas posteriormente em um método de busca local (Busca Tabu) visando a determinação de soluções de melhor qualidade para o PMCCG. Finalmente, apresentamos alguns resultados computacionais e comparamos os resultados obtidos com o valor ótimo encontrado para pequenas instâncias do problema.

Palavras Chaves: Grafo sanduíche, Busca Tabu, PMCC, Heurísticas.

Abstract Given a graph $G = (V, E)$ and a set of vertices $M \subseteq V$, with $|V| = n$, we say that $v \in V$ is *controlled* by M , if the majority of v neighbors (including itself) belongs to M . The set M defines a *monopoly* in G if M controls all vertices of V . In the *Monopoly Verification Problem* - MVP, we are given a set $M \subseteq V$ and two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, with $E_1 \subseteq E_2$. The objective is to find a sandwich graph $G = (V, E)$ (i.e., a graph where $E_1 \subseteq E \subseteq E_2$), such that M defines a monopoly in G . However, if the answer to the MVP is “NO”, we have the

Max-Controlled Set Problem - MCSP, whose objective is to find a sandwich graph $G = (V, E)$, such that the total number of controlled vertices is maximized. The MVP can be solved in polynomial time, the MCSP, however is NP-hard. In this work we describe the notion of *f-controlled* vertices and introduce the *Generalized Max-Controlled Set Problem* - GMCSP, where weights and gaps are associated to all vertices of V . In this case, the objective is to maximize the summation of the weights of all vertices *f-controlled* by M . We present a 0.5-approximation algorithm for the GMCSP and a new procedure for finding feasible solutions based on the solutions of a linear programming relaxation. These solutions are then used in a local search procedure (Tabu Search) looking for solutions of better quality. Finally, we present some computational results and we compare these results with the optimum solutions values obtained for small instances of the problem.

Key words: Sandwich Problems, Tabu Search, MCSP, Heuristics.

1 Introdução

Dado dois grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$ tal que $E_1 \subseteq E_2$, dizemos que $G = (V, E)$, onde $E_1 \subseteq E \subseteq E_2$, é um *grafo sanduíche* com propriedade Π se, e somente se, $G = (V, E)$ satisfaz Π . Um problema de decisão envolvendo grafo sanduíche consiste em decidir se há algum grafo G para o par G_1, G_2 que satisfaça Π . Problemas utilizando grafos sanduíche podem ser vistos em diferentes áreas de pesquisa, como em mapeamento físico de DNA, raciocínio temporal, sincronização de processos paralelos, árvores evolutivas, sistemas esparsos de equações lineares, entre outros [10].

No mapeamento físico de DNA [3], por exemplo, as informações de interseções e não-interseções de pares de segmentos, são obtidas a partir da cadeia de DNA de maneira experimental. O problema consiste em como arranjar os vários segmentos, de modo que seus pares de interseções combinem com os dados experimentais. Na representação utilizando grafos os vértices correspondem aos segmentos, e dois vértices são conectados por uma aresta se os segmentos correspondentes possuem interseções, dessa maneira, define-se o conjunto de arestas E_2 . Na prática as informações de interseções são conhecidas parcialmente, por causa de experimentos incompletos ou resultados não conclusivos. A ambigüidade nas informações de interseções introduz o conjunto de arestas $E_2 \setminus E_1$ (arestas optativas). Assim, decidir sobre esse problema, é equivalente a encontrar um grafo sanduíche com arestas E , tal que $E_1 \subseteq E \subseteq E_2$ [9]. Neste trabalho abordamos um tipo especial de problema em grafo sanduíche, o PROBLEMA DO MAIOR CONJUNTO CONTROLADO GENERALIZADO discutido adiante.

Dado um grafo não direcionado $G = (V, E)$ e um conjunto de vértices $M \subseteq V$, um vértice $i \in V$ é *controlado* por M se $|N_G[i] \cap M| \geq |N_G[i]|/2$, onde $N_G[i] = \{i\} \cup \{j \in V | (i, j) \in E\}$ é vizinhança fechada de i .

O conjunto $M \subseteq V$ define um *monopólio* em G se todo vértice $i \in V$ é controlado por M . O PROBLEMA DE VERIFICAÇÃO DE MONOPÓLIO - PVM é um problema de decisão que retorna “SIM” caso seja encontrando um grafo

sanduíche (se existir), tal que todos os vértices de G estejam controlados por M . Makino *et. al.*[13] demonstram que o grafo sanduíche para o PVM pode ser obtido em tempo polinomial por meio da solução de um Problema de Fluxo Máximo definido convenientemente.

Quando o conjunto $M \subseteq V$ não define um monopólio, ou seja, o problema de decisão PVM retorna “NÃO”, tem-se então o PROBLEMA DO MAIOR CONJUNTO CONTROLADO - PMCC, cujo objetivo é encontrar um grafo sanduíche $G = (V, E)$, tal que o conjunto de vértices controlados por M seja maximizado. O PMCC é NP-difícil, mesmo que G_1 seja um grafo vazio ou G_2 seja um grafo completo [13].

Makino *et. al.*[13] apresentam uma proposta de extensão do PMCC que iremos denominar de PROBLEMA DO MAIOR CONJUNTO CONTROLADO COM FOLGA MÍNIMA - PMCCFM, que introduz a definição de vértice *f-controlado*. Dessa maneira, diremos que um vértice $i \in V$ será *f-controlado* por M em um grafo sanduíche G se, e somente se, $|N_G[i] \cap M| - |N_G[i] \setminus M| \geq f_i$ onde $f_i \in \mathbb{Z}$. A constante f_i representa, de certa forma, o “grau de controle” necessário para que o vértice $i \in V$ seja *f-controlado* por $M \subseteq V$. Chamaremos de *folga mínima* a constante $f_i \in \mathbb{Z}$ associada ao vértice $i \in V$.

Neste trabalho apresentamos uma proposta de extensão para o PMCC e o PMCCFM denominada PROBLEMA DO MAIOR CONJUNTO CONTROLADO GENERALIZADO - PMCCG, que incorpora o conceito de *f-controle* introduzido por Makino *et. al.*[13] e considera pesos associados a cada vértice $i \in V$. No PMCCG, o objetivo será maximizar o somatório dos pesos dos vértices *f-controlados* por M .

Na Seção 2, iremos apresentar as regras de redução, discutidas em [13,14], eliminando informações redundantes nos dados de entrada do PMCC. Como veremos adiante, estas regras serão aplicadas igualmente ao PMCCG, bastando substituir a definição de vértice controlado por *f-controlado*. O PMCCG será tratado na Seção 3. Apresentamos um modelo matemático para o problema (Seção 3.2), um algoritmo determinístico 0,5-aproximado (Seção 3.1) e um algoritmo para obtenção de soluções viáveis a partir da solução da relaxação linear (Seção 3.3). Na Seção 4 introduzimos uma busca local baseada em Busca Tabu e na Seção 5 exibimos alguns dos resultados computacionais obtidos e a comparação com o exato. Finalmente, na Seção 6, apresentamos as conclusões e considerações finais.

2 Regras de redução

As regras de redução colaboram reduzindo ou eliminando informações redundantes nos dados de entrada, sem interferir no resultado (conjunto de soluções ótimas) do problema original. Essas regras se aproveitam de características presentes na estrutura dos grafos G_1 e G_2 , adicionando ou removendo permanentemente arestas pertencentes a $E_2 \setminus E_1$ (arestas optativas), de tal forma que o conjunto de soluções ótimas continue o mesmo.

Descreveremos agora as regras de redução apresentadas em [13,14] para o PMCC. Primeiramente, considere a seguinte notação auxiliar: dados dois grafos G_1 e G_2 como definidos anteriormente e dois conjuntos $A, B \subseteq V$, representamos por $D(A, B) = \{(i, j) \in E_2 \setminus E_1 \mid i \in A, j \in B\}$, o conjunto de arestas optativas com uma extremidade em A e outra em B . Temos então as seguintes regras de redução descritas em [13]: adicionar a E_1 as arestas de $D(M, M)$ (*Regra de redução 1*) e remover as arestas $D(U, U)$ de E_2 (*Regra de redução 2*), onde $U = V \setminus M$. O novo conjunto E deverá satisfazer a:

$$E \supseteq E_1 \cup D(M, M) \quad (1)$$

$$E \subseteq E_1 \cup D(M, M) \cup D(U, M) \quad (2)$$

Visando a introdução das regras de redução descritas em [14], considere inicialmente a seguinte partição de V :

- M_{SC} e U_{SC} , vértices *sempre controlados*, consistem dos vértices pertencentes a M e U , respectivamente, que estão controlados qualquer que seja o grafo sanduíche $G = (V, E)$. Em outras palavras, um vértice $i \in (M_{SC} \cup U_{SC})$ será sempre controlado por M , independentemente de quais arestas optativas $E_2 \setminus E_1$ sejam adicionadas ou removidas em G ;
- M_{NC} e U_{NC} , vértices *nunca controlados*, consistem dos vértices pertencentes a M e U , respectivamente, que são sempre não-controlados qualquer que seja o grafo sanduíche $G = (V, E)$. Ou seja, um vértice $i \in (M_{NC} \cup U_{NC})$ nunca será controlado por M , independentemente de quais arestas optativas $E_2 \setminus E_1$ sejam adicionadas ou removidas em G ;
- M_R e U_R , definidos por $M_R = M \setminus (M_{SC} \cup M_{NC})$ e $U_R = U \setminus (U_{SC} \cup U_{NC})$, são vértices que poderão estar controlados ou não de acordo com a adição ou remoção de arestas optativas.

A partição de V descrita acima pode ser realizada, facilmente, após a adição e remoção de todas as arestas optativas de G . Assim, após a adição das arestas de $E_2 \setminus E_1$, identificamos diretamente aqueles vértices em M_{SC} que continuam controlados e os vértices de U_{NC} que continuam não-controlados. Da mesma forma, após a remoção de todas as arestas optativas, identificamos diretamente os vértices de U_{SC} e M_{NC} que continuam controlados e não-controlados, respectivamente.

Considere então as regras de redução (esquematizadas na Figura 1) apresentadas em [14]:

- Adicionar a E_1 as arestas $D(M_{SC} \cup M_{NC}, U_R)$ (*Regra de redução 3*);
- Remover de E_2 as arestas $D(M_R, U_{SC} \cup U_{NC})$ (*Regra de redução 4*);
- Aleatoriamente adicionar ou remover as aresta $D(M_{SC} \cup M_{NC}, U_{SC} \cup U_{NC})$ (*Regra de redução 5*).

Na Regra de redução 1, a adição das arestas optativas irá somente colaborar para o controle de novos vértices em M . O oposto ocorre na Regra 2, onde as arestas optativas que prejudicariam o controle de novos vértices em U são eliminadas. Na Regra 3 são adicionadas as arestas optativas incidentes àqueles

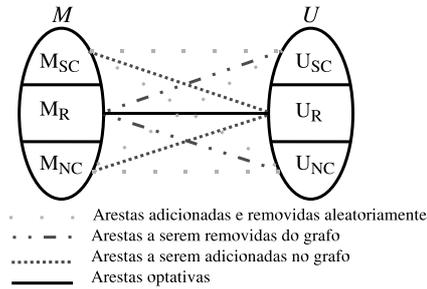


Figura 1. Relação entre os sub-conjuntos de vértices do grafo. Aplicação das regras de redução 3,4 e 5.

vértices de M , sempre ou nunca controlados, facilitando o controle dos outros vértices adjacentes em U_R . Na Regra 4, se um vértice pertence a $U_{SC} \cup U_{NC}$, removemos todas as suas arestas optativas incidentes visando aumentar as chances de controle de vértices adjacentes em M_R . Por fim, a Regra 5 visa somente reduzir o espaço de soluções, já que a adição ou remoção de arestas entre vértices sempre ou nunca controlados não influencia na solução do problema. É fácil ver que, após a aplicação das regras de redução descritas acima, tem-se apenas arestas optativas com extremidades em vértices pertencentes M_R e U_R .

A ordem em que a terceira e quarta regras de redução são aplicadas pode interferir na remoção ou adição de uma aresta optativa. É possível que a aplicação das regras ocorra de maneira que uma regra colabore para a outra e vice-versa. Esse fato leva a aplicação consecutiva das regras 3 e 4 até que não seja mais possível adicionar ou remover arestas optativas. A Figura 2 ilustra um exemplo de aplicações sucessivas das regras 3 e 4. Na Figura 2(a) tem-se um grafo resultante após a aplicação da primeira e segunda regra de redução. A Figura 2(b) demonstra a aplicação da terceira regra de redução. Note que o vértice 1 pertence a M_{SC} , logo todas as suas arestas optativas incidentes serão adicionadas ao grafo sanduíche G . Na Figura 2(c) tem-se a aplicação da quarta regra de redução. Agora o vértice 7 irá pertencer a U_{SC} por consequência da aplicação da terceira regra de redução. A aplicação em seqüência da terceira e quarta regras gera o grafo ilustrado na Figura 2(d).

As regras de redução serão aplicadas ao PMCCG da mesma maneira que são empregadas no PMCC, bastando substituir a definição de vértice controlado por M , para vértice f -controlado por M .

3 Problema do Maior Conjunto Controlado Generalizado - PMCCG

Imagine a seguinte situação, cada vértice no grafo G representa um cidadão. No dia de amanhã, os cidadãos de G irão votar entre “SIM/NÃO” para uma questão

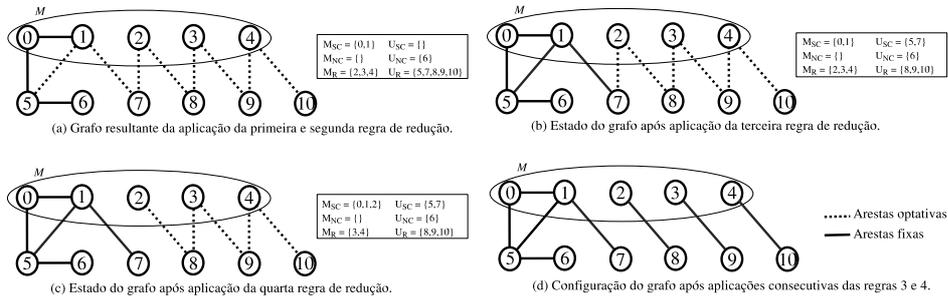


Figura 2. Exemplo de aplicação da terceira e quarta regra consecutivamente

polêmica. Curiosamente, cada cidadão decidiu por fazer uma pesquisa entre seus vizinhos, incluindo seu próprio voto, para ter um censo particular sobre a eleição do dia seguinte. Para a surpresa, ou desapontamento dos cidadãos que votavam “SIM”, o resultado encontrado mostrava uma vantagem de 2:1 para votos “NÃO” em sua vizinhança.

Como exemplo concreto, considere um gráfico completo e bipartido com dois votos para “NÃO” de um lado e $n - 2$ votos “SIM” no outro, veja na Figura 3.

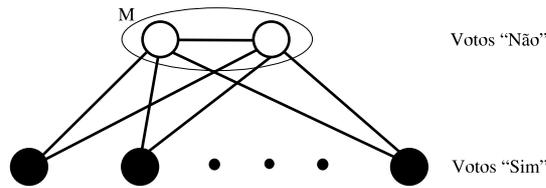


Figura 3. Dois vértices “NÃO” dão a impressão de ser a maioria na pesquisa local de todos os outros vértices. Na figura os vértices brancos representam a coalizão (conjunto M) e os vértices pretos os vértices controlados pela coalizão.

Se cada eleitor tiver seu voto influenciado pela sua pesquisa particular, na eleição hipotética tem-se uma vitória dos votos “NÃO”, mesmo sendo a minoria no momento em que as pesquisas foram realizadas. Uma série de problemas práticos importantes envolvendo coalizões em grafos, maiorias locais e monopólios se enquadram nesse contexto. Maiores detalhes sobre esse assunto podem ser encontrados em [2,15].

A situação apresentada acima caracteriza, de certa forma, o PMCC apresentado em Makino *et. al.*[13], cujo interesse é promover um “controle” de uma maioria de votantes dado um sub-conjunto M de vértices, representando por exemplo uma “coalizão” ou “partido político”. Poderíamos imaginar, por exemplo,

que um projeto importante precisa ser votado em uma câmara legislativa e os vértices controlados por M são aqueles que concordam ou votam favoravelmente a um projeto. É de interesse do partido (coalizão M), promover o controle da maioria dos representantes da câmara, ou seja, que o maior número possível de pessoas, dentro ou fora do partido, apóie um determinado projeto. Entretanto, como estabelecer parcerias e/ou contatos para que o partido consiga a maioria das opiniões a favor do seu projeto na câmara?

Imagine ainda que cada representante eleitor considere um número mínimo de votos (diferença entre os membros de dentro e fora da coalizão) para que sua escolha seja efetivada, teríamos então uma “folga mínima” ou “grau de convencimento” para esse eleitor, caracterizando portanto o PMCCFM. Por último, imagine também que existem eleitores que tenham maior “influência” ou maior “peso político” (seja um indivíduo com cargo importante ou de uma classe econômica privilegiada, por exemplo) e que tivéssemos o interesse de maximizar o “controle” desses eleitores importantes. Essa situação corresponde à idéia dos *pesos* associados aos vértices do grafo, caracterizando agora o PMCCG.

O PMCCG agrega todas as extensões propostas para PMCC. Dessa maneira para cada vértice $i \in V$ teremos um *peso* $p_i \in \mathbb{Z}$ associado, que corresponde à “importância” ou “peso do voto”, e $f_i \in \mathbb{Z}$, o valor da folga mínima ou “grau de convencimento” necessário ao vértice i para que ele seja f -controlado ou concorde com um determinado projeto.

Em seguida, apresentamos um algoritmo polinomial com razão de aproximação igual a 0,5, o modelo matemático para o PMCCG e um algoritmo para encontrar soluções viáveis iniciais por meio de uma relaxação linear do problema.

3.1 Algoritmo 0,5-aproximado para o PMCCG

O PMCC pode ser resolvido por um algoritmo determinístico, de tempo polinomial e razão de aproximação igual a 0,5 (vide Makino *et. al.*[13]). Veremos nessa seção que esse algoritmo pode ser estendido facilmente ao PMCCG, garantindo-se a mesma razão de aproximação.

O Algoritmo 1 pode ser empregado após a aplicação das regras de redução 1 e 2 descritas anteriormente. Além disso, suponha que W_1, W_2 representam o somatório dos pesos dos vértices f -controlados por M em $G = (V, E)$ para $E = E_1$, e $E = E_2$, respectivamente. A variável z_{H1} representa a melhor solução obtida em ambos os casos.

Algoritmo 1 Algoritmo 0,5-aproximado para PMCCG

- 1: $W_1 \leftarrow$ Somatório dos pesos dos vértices f -controlados por M , obtido após a remoção das arestas $D(U, M)$ de G_2 ;
 - 2: $W_2 \leftarrow$ Somatório dos pesos dos vértices f -controlados por M , obtido após a inserção em G_1 das arestas $D(U, M)$;
 - 3: $z_{H1} \leftarrow \max\{W_1, W_2\}$;
-

Teorema 1 *Seja z_{max} o valor da solução ótima do PMCCG. O valor de z_{H1} fornecido pelo Algoritmo 1 satisfaz $z_{H1} \geq \frac{1}{2}z_{max}$, qualquer que seja a instância do problema.*

Prova: É fácil ver que: $z_{max} \leq W_1 + W_2 \leq 2 \times \max\{W_1, W_2\}$. Como $z_{H1} = \max\{W_1, W_2\}$ então $z_{H1} \geq \frac{1}{2}z_{max}$ c.q.d. ■

3.2 Modelo matemático para o PMCCG

A fim de introduzir uma formulação de programação linear inteira para o PMCCG, definimos inicialmente as variáveis binárias z_i , que determinam quais vértices i de V serão f -controlados ou não por M . As variáveis binárias x_{ij} são usadas para decidir quais arestas pertencentes a $E_2 \setminus E_1$ serão consideradas ou não no grafo sanduíche. As constantes $p_i, f_i \in \mathbb{Z}$ correspondem, respectivamente, ao peso e folga mínima do vértice $i \in V$. As constantes binárias $a_{ij} \in \{0, 1\}$ são associadas às arestas $(i, j) \in E_2$, sendo $a_{ij} = 1$, se e somente se, $i = j$ ou $(i, j) \in E_2$. Assumimos ainda que $a_{ij} = a_{ji}, \forall i, j \in V$.

Considere os conjuntos $M_R, M_{SC}, M_{NC}, U_R, U_{SC}, U_{NC}$ como definidos nas regras de redução (Seção 2) obtidos substituindo-se a definição de vértice controlado por f -controlado. Considere ainda uma constante b_i , que possui o valor correspondente à pior folga que um vértice $i \in M_R \cup U_R$ pode assumir. Assim, b_i será calculado com o auxílio da equação (3) a seguir:

$$b_i = \left| \sum_{j \in M} a_{ij}x_{ij} - \sum_{j \in U} a_{ij}x_{ij} - f_i \right| \text{ para } i \in M_R \cup U_R \quad (3)$$

A determinação de b_i é realizada então da seguinte forma, se $i \in M_R$ fazemos $x_{ij} = 1, \forall (i, j) \in E_2 \setminus E_1$. Analogamente, se $i \in U_R$ fazemos $x_{ij} = 0, \forall (i, j) \in E_2 \setminus E_1$. Obviamente, em ambos os casos teremos $x_{ij} = 1, \forall (i, j) \in E_1$.

Temos então o seguinte modelo de programação linear inteira para o PMCCG, obtido após a aplicação da regras de redução:

Maximizar:

$$z_{max} = \sum_{i \in V} p_i z_i \quad (4)$$

sujeito a:

$$\sum_{j \in M} \left(\frac{a_{ij}}{b_i} \right) x_{ij} - \sum_{j \in U} \left(\frac{a_{ij}}{b_i} \right) x_{ij} - \frac{f_i}{b_i} + 1 \geq z_i, \forall i \in M_R \cup U_R \quad (5)$$

$$z_i = 1, \forall i \in M_{SC} \cup U_{SC} \quad (6)$$

$$z_i = 0, \forall i \in M_{NC} \cup U_{NC} \quad (7)$$

$$x_{ij} = 1, \forall (i, j) \in E_1 \quad (8)$$

$$x_{ii} = 1, \forall i \in V \quad (9)$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in E_2 \setminus E_1 \quad (10)$$

$$z_i \in \{0, 1\}, \forall i \in V \quad (11)$$

Na formulação apresentada temos a função objetivo (4) que calcula o somatório dos pesos dos vértices f -controlados por M . A desigualdade (5) garante o f -controle do vértice i por M , sempre que o primeiro membro da inequação for maior ou igual a 1. Caso contrário, o vértice i não será f -controlado por M e z_i será fixado em 0. As divisões por b_i são realizadas para manter a diferença entre os dois somatórios e $\frac{f_i}{b_i}$ sempre maior que -1 , garantindo sempre $z_i \geq 0$. Isso garante também uma relaxação linear de melhor qualidade para o PMCCG, o que não ocorreria por exemplo, se substituíssemos b_i por $|V|$ (já que $|V| > b_i, \forall i \in V$). A igualdade (6) indica os vértices sempre f -controlados no grafo e na igualdade (7) os vértices nunca f -controlados. As igualdades (8) e (9) definem o conjunto de arestas fixas em G_1 . A relaxação linear do PMCCG, representada simplesmente por \bar{P} , é obtida substituindo-se as restrições (10) e (11) (associadas às variáveis de decisão) por $x_{ij} \in [0, 1]$ e $z_i \in [0, 1]$, respectivamente. Representaremos por $\bar{x}_{ij} \in [0, 1], \forall (i, j) \in E_2$, e $\bar{z}_i \in [0, 1], \forall i \in V$, ou simplesmente (\bar{x}, \bar{z}) , uma solução ótima obtida após resolvermos o modelo relaxado do PMCCG. Como discutido em [16], essa relaxação pode ser resolvida em tempo polinomial através dos métodos de pontos interiores. Observe finalmente que, se $f_i = 0$ e $p_i = 1, \forall i \in V$ temos o PMCC conforme descrito em [13]. Entretanto, se apenas $p_i = 1, \forall i \in V$ temos o PMCCFM.

3.3 Solução para o PMCCG baseada na relaxação linear

Mostraremos nesta seção como obter uma solução viável para o PMCCG a partir da relaxação das restrições de integralidade (10) e (11). Assim, um procedimento bastante natural para o problema (que chamaremos Algoritmo 2) pode ser obtido da seguinte forma: dada uma solução (\bar{x}, \bar{z}) de \bar{P} com coordenadas $\bar{x}_{ij} \in [0, 1], \forall (i, j) \in E_2$ e $\bar{z}_i \in [0, 1], \forall i \in V$, defina como f -controlado, todos os vértices $i \in V$ onde $\bar{z}_i = 1$, e como não f -controlado os demais vértices de V , onde $\bar{z}_i < 1$.

Como veremos adiante, os valores $\bar{x}_{ij} \in [0, 1]$ obtidos após a solução de \bar{P} poderão ser sempre inteiros, significando portanto, que ao escolhermos os vértices f -controlados ou não por M através de \bar{z} , teremos automaticamente um mecanismo para decidir quais arestas optativas serão ou não selecionadas ao final do procedimento. Formalmente, temos a seguinte proposição demonstrada em anexo.

Teorema 2 *Considere uma solução ótima (\bar{x}, \bar{z}) de \bar{P} com coordenadas $\bar{x}_{ij} \in [0, 1], \forall (i, j) \in E_2$, $\bar{z}_i \in [0, 1], \forall i \in V$, e valor ótimo $\bar{z}_{max} = \sum_{i \in V} \bar{z}_i$. Se $\bar{x}_{ij} \in (0, 1)$ para alguma aresta $(i, j) \in E_2 \setminus E_1$, existirá então, uma nova solução relaxada (\tilde{x}, \tilde{z}) de \bar{P} onde $\bar{z}_{max} = \sum_{i \in V} \tilde{z}_i$ sendo $\tilde{x}_{ij} \in \{0, 1\}, \forall (i, j) \in E_2$ e $\tilde{z}_i \in [0, 1], \forall i \in V$.*

Note no exemplo da Figura (4.a), que para $f_i = 0, \forall i \in V$, uma solução ótima do problema relaxado (associado a (4)-(11)), pode ser obtida fazendo-se $\bar{x}_{ij} = 0, \forall (i, j) \in E_2 \setminus E_1$. Na Figura (4.b) entretanto, observamos uma outra solução ótima onde $\bar{x}_{ij} \in \{0, 1\}, \forall (i, j) \in E_2 \setminus E_1$. Em ambos os casos, 4 vértices são controlados.

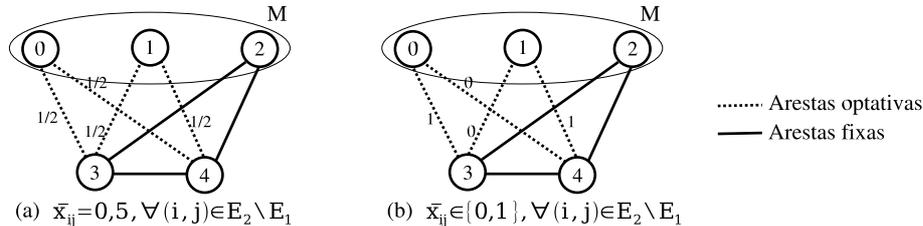


Figura 4. Exemplo de grafos equivalentes com solução fracionária e solução inteira para \bar{x}_{ij} .

Um novo procedimento, que chamaremos de Algoritmo 3, consiste em simplesmente selecionar a melhor solução obtida nos Algoritmos 1 e 2. Como demonstrado em [14], o Algoritmo 3 possui uma razão de performance superior para o PMCC (situação particular do PMCCG onde $f_i = 0$ e $p_i = 1, \forall i \in V$). Aquelas instâncias onde $n \leq 4$, podem ser resolvidas facilmente, de maneira exata e em tempo polinomial. Para maiores detalhes sobre esse procedimento e sua análise de aproximação vide [14].

Teorema 3 *O Algoritmo 3 garante, em tempo polinomial, uma razão de performance para o PMCC igual a $\frac{1}{2} + \frac{1+\sqrt{n}}{2(n-1)}, \forall n > 4$.*

Visando incrementar ainda mais as soluções obtidas pelos Algoritmos 1 e 3, descrevemos a seguir um procedimento de busca local para o PMCCG baseado no procedimento de Busca Tabu.

4 Busca Tabu aplicada ao PMCCG

O método de Busca Tabu (*Tabu Search*) - BT, proposto independentemente por [6] e [11] em 1986, é um procedimento iterativo para a solução de problemas de otimização combinatória e que aceita movimentos de piora da solução corrente para tentar escapar de ótimos locais evitando, dessa forma, retornar a regiões previamente pesquisadas.

Dentre as metaheurísticas existentes na literatura, a BT tem se mostrado competitiva na resolução de diferentes problemas de otimização combinatória. Na BT, em cada iteração uma busca local é executada vasculhando-se uma vizinhança $N(S)$ da solução corrente S . Soluções ou movimentos presentes na *lista tabu* são proibidos temporariamente. Após um determinado número de iterações, a lista é atualizada de forma a permitir que novos movimentos ou soluções assumam o *status* de tabu. A lista tabu consiste de um conjunto de *soluções* ou *movimentos proibidos*, determinados por informações históricas das iterações precedentes no procedimento de busca local.

A BT apresentada no Algoritmo 4 contém as etapas normalmente encontradas nos modelos tradicionais da literatura. Dessa forma, o método inclui construção da solução inicial, fase de busca local intensiva e diversificação (fuga de ótimos locais).

Considere S (com custo $c(S)$ associado) a solução corrente no algoritmo e S^* a melhor solução encontrada (*solução global*). A lista tabu será representada por T e irá conter os atributos de cada movimento, dessa maneira, uma série de movimentos proibidos será definida implicitamente. O critério de aspiração permite que sejam exploradas regiões ainda não visitadas no espaço de soluções, garantindo que movimentos presentes na lista Tabu sejam executados, desde que promovam uma solução melhor que a solução global encontrada.

Algoritmo 4 Pseudo código do BT aplicado a um problema de maximização

```

1: Contadores de iterações  $i, j \leftarrow 0$ 
2:  $S, S^* \leftarrow \text{Solução inicial}(), c(S'') \leftarrow -\infty$ 
3: enquanto  $i < i_{max}$  faça
4:    $T \leftarrow \emptyset$ 
5:   enquanto  $j < j_{max}$  faça
6:      $S' \leftarrow \text{BuscaLocal}(N(S) \setminus T)$ 
7:      $S'' \leftarrow \text{BuscaLocal}(N(S) \cap T)$  satisfazendo o critério de aspiração
8:     se  $c(S'') > c(S')$  então
9:        $S' \leftarrow S''$ 
10:    fim se
11:    se  $c(S') > c(S^*)$  então
12:       $S^* \leftarrow S'$ 
13:       $j \leftarrow 0$ 
14:    fim se
15:    se  $c(S') < c(S)$  então
16:      Insira o movimento  $(S', S)$  na lista tabu  $T$ 
17:    fim se
18:     $S \leftarrow S'$  e  $j \leftarrow j + 1$ 
19:  fim enquanto
20:   $\text{Diversificação}(S)$ 
21:   $i \leftarrow i + 1$ 
22: fim enquanto
23: retorna  $S^*$ 

```

No caso do PMCCG, o procedimento de construção da solução inicial (linha 2) utilizado, consiste do Algoritmo 3 apresentado na Seção 3.3. Naquelas instâncias onde o número de restrições é muito grande para serem resolvidas por Programação Linear, geramos a solução inicial utilizando o Algoritmo 1 como descrito na Seção 3.1. Os contadores i e j , denotam o número de iterações do algoritmo e da busca local respectivamente, sendo que a busca local é interrompida sempre que j_{max} iterações ocorram sem melhoria da solução global. Na linha 6, a busca local em $N(S)$, é executada observando-se os movimentos proibidos presentes na lista tabu. Esses movimentos são verificados e realizados na linha

7, caso promovam uma melhora na solução global, ou seja, caso o critério de aspiração seja satisfeito. Para escapar de máximos locais, a busca local permite também movimentos de piora da solução corrente. Nesse caso deve-se garantir que uma busca conseguinte não retorne novamente a uma região já pesquisada. Essa garantia é dada pela inserção do movimento de “volta” na lista tabu, representada por um ou mais atributos do movimento que levaria a uma solução já encontrada (linhas 15 e 16). A lista tabu se baseia normalmente no conceito de fila. Assim, em cada atualização um movimento proibido é incluído no final da lista (linha 16), ficando presente na lista tabu por um número de iterações equivalente à capacidade dessa lista. A função de diversificação, executada na linha 20, efetua uma perturbação na solução corrente, com o objetivo de fugir de ótimos locais, indo para regiões ainda não pesquisadas pela busca local.

As etapas de busca local e diversificação da solução corrente são descritas detalhadamente a seguir.

4.1 Fase de busca local intensiva no PMCCG

Uma das etapas mais importante da BT, se refere à etapa de busca local intensiva onde a vizinhança de uma solução corrente S é pesquisada. A qualidade dessa busca local pode ser medida em função do tamanho dessa vizinhança e do número de vezes em que ela é efetuada numa dada região. A idéia, a princípio, é investigar uma mesma região enquanto ela se mostrar promissora.

A busca local (linhas 6 e 7 do Algoritmo 4) analisa a partir de uma solução inicial (semente) soluções vizinhas, utilizando um critério normalmente guloso, ou seja, priorizando o controle dos vértices com maior peso associado. Na busca local aqui realizada, buscamos o f -controle de novos vértices do grafo sem que outros vértices sejam descontrolados (ou não- f -controlados). Representaremos esta estrutura de vizinhança por $N_0(S)$.

O procedimento de busca local proposto para o PMCCG é ilustrado sucintamente no Algoritmo 5. Dado um grafo sanduíche G (solução corrente S), teremos um conjunto de arestas optativas associadas e um conjunto $M_S \subseteq M_R$ (respectivamente $U_S \subseteq U_R$) dos vértices f -controlados (temporariamente) por M em G . Considere também $L_S = M_S \cup U_S$, $L_D = (M_R \cup U_R \setminus L_S)$ e $L_v = \{x \in V \mid (v, x) \in E_2 \setminus E_1\}$.

Nesta estrutura de vizinhança esperamos que os vértices $v \in L_D$ sejam f -controlados desde que nenhum outro vértice temporariamente f -controlado seja *descontrolado*. Note que, neste caso, uma nova solução de melhor custo, pertencente a $N_0(S)$ será gerada já que todos os pesos associados aos vértices de V são positivos. Ainda, dado um grafo sanduíche G (solução corrente S), chamaremos de *folga corrente* de um vértice $i \in V$ o valor $f_G(i) = |N_G[i] \cap M| - |N_G[i] \cap U| - f_i$. Assim, dado um grafo sanduíche G , dizemos que $i \in V$ é f -controlado se, e somente se, $f_G(i) \geq 0$, caso contrário i será descontrolado (ou não f -controlado).

A busca local tenta controlar vértices $v \in L_D$ (linha 2), desde que o vértice selecionado não esteja presente nas soluções definidas implicitamente pela lista tabu (linha 4). Note por exemplo que $f_G(v) < 0, \forall v \in L_D$. Representaremos os vizinhos de v que podem auxiliar em seu f -controle por $\bar{L}_v = \{w \in L_v \mid f_G(w) \neq$