

MÉTODOS EXATOS E HEURÍSTICOS PARA O PROBLEMA DO FLUXO COM ROTULAÇÃO MÍNIMA

KERVEN MACIEL MONTEIRO DE ALBUQUERQUE (IFPB, Campus João Pessoa), THIAGO GOUVEIA (IFPB, Campus João Pessoa)

E-mails: albuquerque.kerven@academico.ifpb.edu.br, thiago.gouveia@ifpb.edu.br.

Área de conhecimento: 1.03.03.04-9 Sistemas de Informação.

Palavras-chave: grafos com arestas rotuladas; fluxo máximo; cortes coloridos; algoritmo genético; GRASP.

1 Introdução

Em um grafo com arestas rotuladas (GAR), cada aresta está associada a um rótulo, ou cor. Esses rótulos representam características qualitativas que não podem ser expressas por pesos, custos ou capacidades. Muitos problemas definidos sobre GARs são discutidos na literatura, como o problema da árvore geradora de rotulação mínima, que consiste em encontrar uma árvore geradora em um dado GAR usando um número mínimo de rótulos distintos (SILVA et al., 2019).

Este trabalho aborda o problema do fluxo com rotulação mínima (PFRM), definido sobre um grafo orientado, ou dígrafo, com arcos rotulados e capacitados (DARC). O problema consiste em encontrar um fluxo em um dado DARC usando um número mínimo de rótulos e de modo que o valor do fluxo seja maior ou igual a um valor dado. Um estudo aprofundado sobre a teoria e aplicações de fluxos pode ser consultado em Ahuja et al. (2013).

O PFRM pode ser aplicado em áreas como redes de computadores, telecomunicações, transporte multimodal, entre outras. Um exemplo no projeto de redes consiste em encontrar a infraestrutura mínima para atender à demanda esperada de uma rede. Os rótulos podem representar diferentes provedores de serviço para uma rede de computadores, tecnologias e protocolos para uma rede IoT; modos de transporte para uma rede de transporte multimodal, etc. Nesses casos, minimizar o número de rótulos geralmente reduz o custo total. Isso é ilustrado na Figura 1, onde uma demanda de 5 unidades de fluxo entre s e t é satisfeita usando apenas dois rótulos.

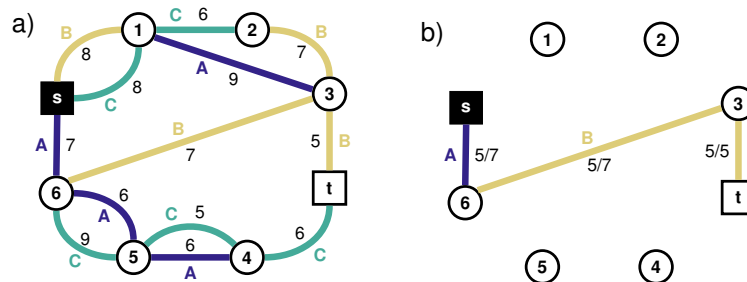


Figura 1: Aplicação do PFRM no projeto de rede com demanda 5 (b) a partir do DARC das possibilidades (a).

Este trabalho propõe métodos exatos e heurísticos para o PFRM, incluindo uma formulação baseada em fluxo com algoritmos de *branch and cut* e métodos baseados em algoritmo genético e na meta-heurística GRASP e um método de manutenção dinâmica de fluxo. Também são apresentados os resultados dos experimentos computacionais, comparando o desempenho e qualidade da solução com os métodos encontrados na literatura.

2 Materiais e Métodos

Seja um DARC $D = (V, L, A)$, onde V é o conjunto de vértices, L de rótulos e A de arcos, e cada arco $a \in A$ possui um rótulo $\ell_a \in L$ e uma capacidade $c_a \geq 0$ associados. Sejam, ainda, um valor desejado de fluxo $0 \leq f_g \leq f_{\max}$ e vértices de origem $s \in V$ e destino $t \in V$. O PFRM busca obter um subconjunto mínimo de rótulos $L' \subseteq L$ tal que, no subgrafo gerador induzido por L' , ou $D[L']$, exista um fluxo f com valor $|f| \geq f_g$. No caso particular onde $f_g = f_{\max}$, trata-se de uma instância do problema do fluxo máximo com rotulação mínima (PFMRM), abordado por Granata et al. (2013).

Como formulação matemática para o PFRM, foi proposto o programa linear (1) a (3). A função objetivo (1) busca minimizar o número de rótulos utilizados na solução, já considerando que o conjunto L_N de rótulos necessários estarão presentes. Os rótulos necessários são aqueles para os quais não existe fluxo com o valor dado, se eles forem removidos do grafo. As variáveis binárias z_ℓ indicam se cada rótulo ℓ estará presente ou não na solução, com domínio definido por (2).

$$|L_N| + \min_{\ell \in L \setminus L_N} z_\ell \quad (1)$$

sujeito a

$$z_\ell \in \{0, 1\}, \quad \forall \ell \in L \quad (2)$$

$$\sum_{\ell \in L^+(S) \setminus L_N} z_\ell \min \left(\sum_{\substack{a \in \delta^+(S) \\ \ell = \ell_a}} c_a, f_g - \sum_{\substack{a \in \delta^+(S) \\ \ell_a \in L_N}} c_a \right) \geq f_g - \sum_{\substack{a \in \delta^+(S) \\ \ell_a \in L_N}} c_a, \quad \forall S \subset V, s \in S, t \notin S \quad (3)$$

A restrição (3) é uma versão melhorada da inequação $\sum z_\ell \sum c_a \geq f_g$, que resulta do teorema do fluxo máximo e corte mínimo. Uma prova desse teorema pode ser consultada em Bondy e Murty (1976). Esses somatórios consideram apenas os arcos (u, v) , com $u \in S$ e $v \notin S$, representados pelo conjunto $\delta^+(S)$. O conjunto dos rótulos dos arcos em $\delta^+(S)$ é $L^+(S)$. Essa inequação, então, estabelece o valor do fluxo como um limite inferior para a capacidade do corte, limitada aos rótulos da solução. Considerando todos os cortes do grafo, tem-se um fluxo válido.

Como há uma quantidade exponencial de restrições, uma para cada corte do grafo, são propostos dois algoritmos de *branch and cut*. O primeiro algoritmo parte de um modelo contendo apenas restrições básicas e calcula o fluxo no subgrafo correspondente à solução do modelo. Se o valor do fluxo for inferior ao desejado, identifica-se um corte violado e adiciona-se a restrição correspondente ao modelo, repetindo o processo até obter o valor desejado de fluxo. O segundo algoritmo é semelhante, mas em duas etapas. Inicialmente, as restrições de integralidade são relaxadas para permitir que o resolvidor trabalhe apenas na raiz da árvore de decisão. Em seguida, as restrições de integralidade são reintroduzidas, de modo que o modelo já inicia com os cortes previamente identificados, melhorando a estabilidade e a convergência.

Como métodos heurísticos, foram propostos métodos baseados nas meta-heurísticas GRASP (*greedy randomized adaptive search procedure*) e algoritmo genético. O método baseado no GRASP reativo é composto por duas fases principais: construção gulosa aleatorizada e busca local. Na fase de construção, a solução inicia com os rótulos necessários e é expandida com rótulos escolhidos de forma aleatória entre os mais promissores, de acordo com sua contribuição para aumentar o fluxo. O nível de aleatoriedade é controlado por um parâmetro $\alpha \in (0, 1)$, ajustado dinamicamente ao longo das iterações, com base no desempenho das soluções geradas por cada valor possível do parâmetro. Após a construção, aplica-se uma busca local que tenta remover rótulos supérfluos da solução, mantendo o valor de fluxo acima do mínimo exigido. O processo é repetido até atingir o tempo limite, e ao final, retorna-se a melhor solução encontrada.

O método baseado no algoritmo genético (AG) incorpora diversas estratégias para intensificar e diversificar a busca por soluções de qualidade. A população inicial é gerada utilizando uma construção gulosa aleatorizada baseada no GRASP, garantindo soluções iniciais mais promissoras. Também usa-se uma estratégia elitista, organizando a população em três classes: melhores soluções, intermediárias e aleatórias, e a recombinação ocorre entre indivíduos da classe de melhores soluções e demais indivíduos. A recombinação é feita por meio de *path-relinking*, construindo um caminho entre as soluções para obtenção dos descendentes. As soluções geradas passam por uma etapa de busca local e, a cada geração, as melhores soluções são preservadas nas primeiras classes e novas soluções são geradas para a terceira classe. O processo é repetido até o tempo limite e a melhor solução obtida é retornada.

Com o objetivo de melhorar a eficiência computacional das heurísticas propostas, também foi proposto um método de manutenção dinâmica de fluxo. Em vez de recalcular o fluxo máximo do zero a cada modificação no conjunto de rótulos, esse método atualiza incrementalmente o fluxo existente, reutilizando informações da solução anterior. Quando um rótulo é removido, o fluxo é ajustado apenas nos arcos afetados por aquela remoção e, da mesma forma, ao adicionar um rótulo, a estrutura do fluxo é adaptada para considerar os novos arcos disponíveis. Essa abordagem reduz significativamente o tempo de processamento, especialmente durante as etapas de construção e busca local, onde é necessário verificar repetidamente se o fluxo mínimo é atendido após a adição ou remoção de rótulos. Mais detalhes sobre os métodos propostos podem ser consultados em <<https://gitlab.com/pfrm-simpif/algoritmos>>.

3 Resultados e Discussão

Em relação aos métodos exatos, os experimentos computacionais compararam os dois algoritmos de *branch and cut* propostos para a formulação baseada em cortes coloridos (FCC1 e FCC2) com a formulação baseada em fluxo (FF) de Granata et al. (2013). Observou-se que os métodos FCC1 e FCC2 superaram o desempenho da FF na maioria dos casos: FCC1 superou FF em 99% das instâncias, enquanto FCC2 superou FF em 97% das instâncias. Convém destacar que em 4 das 720 instâncias testadas, o algoritmo FCC1 não convergiu dentro do limite de memória, devido a um número alto de nós e cortes necessários. Em todos eles, o algoritmo obteve uma solução com exatamente um rótulo a mais do que o ótimo. Esse problema é resolvido pelo FCC2, que convergiu em todas as instâncias. O desempenho do FCC1 superou o do FCC2 em 82% dos testes. No entanto, pode-se afirmar que o FCC2 foi mais estável em termos de convergência.

Em relação aos métodos heurísticos, foi utilizado um conjunto de 1800 instâncias para comparar o GRASP e o AG com o SVNS proposto por Granata et al. (2013). Nas instâncias menores, os três algoritmos obtiveram as mesmas soluções, com o AG e o GRASP obtendo a solução em menos tempo. Nas instâncias intermediárias, o AG alcança uma solução 0,2% melhor que o SVNS, enquanto o GRASP obtém uma solução 0,1% melhor que o SVNS, em média. Em termos de desempenho, tanto o AG quanto o GRASP são mais rápidos que o SVNS, melhorando o tempo em 58,2% e 52,7%, respectivamente, em média. Nas maiores instâncias, o AG obteve o melhor resultado em todas as instâncias, sendo 8,3% melhor que o SVNS, em média. Nesses casos, o GRASP foi 8,2% melhor que o SVNS, em média. Em relação ao tempo, o AG obteve um tempo 80,1% melhor que o SVNS, e o GRASP obteve 79,2%, em média.

Quanto ao método de manutenção dinâmica de fluxo (EKD), foi utilizada uma base de 1000 instâncias para compará-lo com o algoritmo de Edmonds e Karp (1972) (EK). Em relação ao cálculo inicial e adição de rótulos, o desempenho de EK e EKD foi bem semelhante, dado que os algoritmos tem a mesma estrutura básica. Em relação à remoção de rótulos, o EKD obteve um desempenho 48% a 89% melhor que o EK. Observou-se que o valor dado de fluxo afeta inversamente o desempenho do algoritmo. Nas instâncias com fluxo dado maior, há mais arcos com fluxo não nulo e, conseqüentemente, mais arcos a serem removidos e mais caminhos por onde a remoção pode ser realizada. Finalmente, também observou-se que o desempenho do EKD na remoção de rótulos é melhor em grafos mais densos e multigrafos. As tabelas completas podem ser consultadas em <<https://gitlab.com/pfrm-simpif/resultados>>.

4 Considerações Finais

O PFRM é uma generalização do PFMRM, ambos NP-difíceis e com grande potencial de aplicação. Como contribuições desta pesquisa tem-se os métodos exatos e heurísticos propostos para o PFRM, além do método de manutenção dinâmica de fluxo e os experimentos computacionais realizados.

Em relação aos métodos exatos, foi proposta uma formulação baseada em cortes coloridos e dois algoritmos de *branch and cut* para lidar com o número exponencial de restrições. O primeiro teve melhor desempenho, mas falhou em alguns casos. O segundo foi mais estável, mas menos eficiente. Ambos superaram a formulação baseada em fluxo da literatura. Futuramente, sugere-se melhorar as inequações e explorar algoritmos aproximativos.

Quanto aos métodos heurísticos, foram propostos métodos baseados em GRASP reativo e algoritmo genético com construção gulosa aleatorizada, elitização e recombinação por caminhos, ambos superando o SVNS da literatura em termos de desempenho e qualidade da solução. O método de manutenção dinâmica de fluxo também mostrou-se eficiente ao remover rótulos, em comparação com o algoritmo de Edmonds e Karp (1972). Essa vantagem acentua-se em grafos mais densos e multigrafos, que também podem constituir objeto de pesquisa em trabalhos futuros.

Referências

- AHUJA, R. A. et al. *Network Flows: Theory, Algorithms and Applications*. New Jersey: Prentice Hall, 2013.
- BONDY, J. A.; MURTY, U. S. R. *Graph Theory with Applications*. London: Macmillan, 1976.
- EDMONDS, J.; KARP, R. M. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, v. 19, n. 2, p. 248–264, 1972.
- GRANATA, D. et al. Maximum flow problems and an np-complete variant on edge-labeled graphs. *Handbook of combinatorial optimization*, Springer New York, New York, NY, p. 1913–1948, 2013.
- SILVA, T. G. et al. A hybrid metaheuristic for the minimum labeling spanning tree problem. *European Journal of Operational Research*, v. 274, n. 1, p. 22–34, 2019. ISSN 0377-2217.