UM ALGORITMO $(1/2)$-APROXIMATIVO PARA O PROBLEMA DO MÁXIMO SUBGRAFO ACÍCLICO SOB RESTRIÇÕES DISJUNTIVAS NEGATIVAS

Sílvia Maria Santana Mapa
Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG - CEP 31270-901
silvia.mapa@ifmg.edu.br

Sebastián Urrutia
Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG - CEP 31270-901
surrutia@dcc.ufmg.br

RESUMO

Recentemente, versões com restrições disjuntivas de problemas clássicos em teoria dos grafos tem sido estudados, tais como caminhos mais curtos, árvores geradoras mínimas e emparelhamentos máximos. Em restrições disjuntivas negativas, um certo par de arcos não podem pertencer simultaneamente a uma solução viável. O problema do Máximo Subgrafo Acíclico, em sua forma básica, é bem conhecido na literatura por pertencer à classe NP-Difícil, para o qual há um algoritmo $(1/2)$-aproximativo de aplicação direta. Neste artigo, é proposto um algoritmo aproximativo para o problema do Máximo Subgrafo Acíclico sob Restrições Disjuntivas Negativas, considerando um caso especial no qual as restrições disjuntivas seguem um padrão específico. Para este caso especial, o algoritmo desenvolvido mantém o fator de aproximação do algoritmo original.

PALAVRAS CHAVE. Máximo Subgrafo Acíclico, Restrições de Conflito, Algoritmo Aproximativo, Teoria e Algoritmos em Grafos.

ABSTRACT

Disjunctively constrained versions of classic problems in graph theory such as shortest paths, minimum spanning trees and maximum matchings were recently studied. Negative disjunctive constraints state that a certain pair of edges cannot be contained simultaneously in a feasible solution. The Maximum Acyclic Subgraph in its basic form is well-known in the literature as an NP-Hard problem which has a straightforward $(1/2)$-approximation algorithm. In this paper we propose an approximation algorithm for the Maximum Acyclic Subgraph problem under Negative Disjunctive Constraints, considering a special case in which the disjunctive constraints follow a specific pattern. For this special case, our algorithm maintains the approximation ratio of the original algorithm.

KEYWORDS. Maximum Acyclic Subgraph, Conflict Constraints, Approximation Algorithm, Theory and Algorithms in Graphs.

1. Introdução

Um grafo dirigido pode conter ciclos, os quais podem ser indesejáveis ou não permitidos em algumas aplicações práticas. Portanto, nessas aplicações surge a necessidade de se retirar uma quantidade mínima de arcos do grafo original para se obter um subgrafo acíclico máximo. O problema de se encontrar o Máximo Subgrafo Acíclico (MSA) de um dado grafo dirigido $G = (V, A)$ consiste em determinar um subconjunto máximo $A' \subseteq A$ no qual o subgrafo $G' = (V, A')$ seja acíclico. Este problema é conhecido por ser NP-Difícil (Karp, 1975). Um algoritmo aproximativo direto para este problema produz soluções com pelo menos metade do número de arcos da solução ótima (Jünger, 1985).

É comum a aplicação do MSA em projetos em que existem restrições de preferências entre tarefas. Problemas deste tipo podem ser formulados como MAS com a inclusão de restrições disjuntivas entre pares de arestas. Supondo que os vértices do grafo representam as tarefas e os arcos preferências de precedência, o MAS irá atribuir uma ordenação parcial às tarefas de forma a cumprir com o maior número possível de preferências. As restrições disjuntivas constituem uma forma de forçar certas preferências, consideradas importantes no projeto.

Versões com restrições disjuntivas de problemas clássicos em teoria dos grafos tem sido recentemente estudados, tais como problemas de caminhos mais curtos, árvores geradoras mínimas e emparelhamentos máximos (Darmann et al., 2011). Neste artigo, os autores propõem restrições disjuntivas negativas e positivas. Restrições disjuntivas negativas são aquelas nas quais um certo par de arcos não podem compor simultaneamente uma solução viável. Em restrições disjuntivas positivas, para cada par de arcos, pelo menos um deles deve compor a solução. Os três problemas, caminhos mais curtos, árvores geradoras mínimas e emparelhamentos máximos, são resolvidos em tempo polinomial em suas formas básicas, e se tornam NP-Difíceis quando submetidos às restrições disjuntivas de conflito.

No presente artigo é abordado o problema do Máximo Subgrafo Acíclico sob Restrições Disjuntivas Negativas (MSARDN). Mais especificamente, é proposta uma simples modificação do algoritmo aproximativo existente na literatura para o MSA que produz soluções para o MSARDN mantendo o fator de aproximação $(1/2)$, sempre que as restrições disjuntivas negativas seguem um padrão específico.

O restante deste artigo está organizado da seguinte forma: na próxima seção é feita uma revisão da literatura relevante; na seção 3 o problema MSARDN é formalmente introduzido; na seção 4 é feita uma revisão do algoritmo aproximativo para o MSA existente na literatura; na seção 5 é proposto um novo algoritmo aproximativo para o MSARDN e o seu fator de aproximação é provado; conclui-se o artigo na seção 6.

2. Trabalhos Relacionados

O problema do Máximo Subgrafo Acíclico é também conhecido na literatura como *Linear Ordering Problem* (LOP) ou *Minimum Feedback Arc Set Problem* (MFASP). Em MFASP, o objetivo é encontrar um conjunto mínimo de arcos que devem ser removidos do grafo de forma a torná-lo acíclico. No LOP, dado um grafo direcionado com pesos, deseja-se encontrar um subgrafo de máximo peso que não contenha ciclos. A solução para este problema gera uma ordenação linear dos vértices no qual todos os arcos do subgrafo obtido são arcos diretos. Arcos diretos são aqueles que, dada uma ordenação topológica do grafo em que seus vértices estão ordenados ao longo de uma linha horizontal, seguem da esquerda para a direita. A complexidade do LOP pode ser facilmente provada como NP-difícil transformando-o no problema equivalente MFASP, o qual é amplamente conhecido como um problema NP-Difícil (Garey and Johnson, 1979). Segundo Chaovalitwongse et al. (2011), o LOP possui aplicações em diferentes campos, tais como: computação, economia, arqueologia, ciências sociais, programação, biologia e esportes.

Existem pelo menos três abordagens para tratar problemas da classe NP-Difícil. Primeiramente, se as instâncias são pequenas, algoritmos exponenciais podem ser satisfatórios. Em uma segunda abordagem, casos especiais do problema podem ser isolados e resolvidos em tempo polinomial. Outra abordagem é encontrar soluções próximas ao ótimo em tempo polinomial por meio de técnicas heurísticas ou algoritmos aproximativos. Estes últimos garantem que a solução encontrada está a um certo fator distante da solução ótima.

Abordagens exatas para o LOP constam na literatura usando técnicas tais como *Cutting Plane Algorithms* (Reinelt, 1985) e *Interior Point Methods* (Mitchell, 1997). Métodos heurísticos baseados em GRASP (*Greedy Randomized Adaptive Search Procedure*) (Chaovalitwongse et al., 2011), *Tabu Search* (Laguna et al., 1999), *Variable Neighborhood Search* (Garcia et al., 2006), *Genetic Search* (Schiavinotto and Stützle, 2004) e *Simulated Annealing* (Martí et al., 2012) são capazes de encontrar soluções de boa qualidade para o LOP.

Em Laguna et al. (1999) os autores desenvolveram um algoritmo baseado em *Tabu Search* analisando algumas técnicas de intensificação e diversificação e, por meio de extensivos experimentos computacionais, obtiveram resultados para o LOP melhores que os das heurísticas já reportadas na literatura até aquele momento. Em Chaovalitwongse et al. (2011) os autores propuseram um algoritmo eficiente baseado em GRASP integrado com a técnica *Path Relinking* e um novo procedimento de busca local, sendo capaz de encontrar soluções ótimas para a maior parte das instâncias testadas, obtendo um *gap* máximo de 0.05%. Martí et al. (2012) fizeram um estudo comparativo de 24 técnicas heurísticas e metaheurísticas aplicadas ao LOP objetivando encontrar soluções próximas à otimalidade. Segundo estes autores, os algoritmos mais eficientes para abordar o LOP foram, nesta sequência: *Memetic Algorithm*, *Iterated Local Search* e *Tabu Search*.

Além do uso de técnicas heurísticas e algoritmos exatos, o LOP também é tratado por algoritmos aproximativos. Em Newman (2000) o autor mostra que o LOP não pode ser aproximado por um fator constante melhor que $(1/2)$ usando-se a técnica de relaxações poliedrais e mostra também que obter um algoritmo $(65/66+\epsilon)$-aproximativo, com $\epsilon > 0$, para o LOP é NP-Difícil. Este autor desenvolveu um algoritmo $(8/9)$-aproximativo para o problema do máximo subgrafo acíclico de grau máximo igual a três.

3. O problema do Máximo Subgrafo Acíclico sob Restrições Disjuntivas Negativas

É conveniente representar as restrições disjuntivas negativas em termos de um grafo de conflitos, não direcionado, cujos vértices correspondem aos arcos do grafo original e cujas arestas representam as restrições de conflito. Seja $G = (V, A)$ um grafo direcionado, $\bar{G} = (A, E)$ um grafo de conflitos e $m = |A|$. O problema do Máximo Subgrafo Acíclico sob Restrições Disjuntivas Negativas consiste em determinar um subconjunto máximo

$A' \subseteq A$ no qual o subgrafo $G' = (V, A')$ seja acíclico e A' seja um conjunto independente em $\bar{G}$. Como MSARDN tem o problema MSA como um caso especial (fazendo $E = \emptyset$), MSARDN também é NP-Difícil.

Em Darmann et al. (2011) os autores analisaram os problemas de caminho mais curto, árvore geradora mínima e emparelhamento máximo sob restrições disjuntivas negativas considerando dois tipos de grafos de conflito $\bar{G}$, então denominados: 2-ladder e 3-ladder. No primeiro tipo, todos os componentes conexos de $\bar{G}$ são caminhos de tamanho igual a um. No segundo tipo, todos os componentes conexos de $\bar{G}$ são caminhos de tamanho dois. Neste artigo, o algoritmo aplicado ao problema do MSARDN será resolvido em tempo polinomial para o conjunto de classes de grafos de conflitos nos quais o problema de conjunto independente máximo é resolvido em tempo polinomial. Este conjunto inclui grafos claw-free (Sbihi, 1980) e perfeitos (Grötschel et al., 1988).

O problema do MSARDN pode ser formulado por meio de programação linear inteira. Seja $G = (V, A)$ um grafo dirigido como descrito anteriormente, com o conjunto de vértices V, o conjunto de arcos $A = \{(i, j) : i, j \in V \ \wedge i \neq j\}$ e uma lista de conflitos entre pares de arcos $L = \{\{(i, j), (k, l)\} : (i, j), (k, l) \in A \wedge (i, j) \neq (k, l)\}$, no qual a cada par $\{(i, j), (k, l)\} \in L$ está associado uma aresta $e \in E$ do grafo de conflitos $\bar{G} = (A, E)$. O objetivo é encontrar o conjunto de máxima cardinalidade $A' \subseteq A$ tal que $G' = (V, A')$ é acíclico e satisfaça a lista de restrições de conflitos dada em L.

Para a formulação, se o grafo G não é completo, será então introduzido um subconjunto C de arcos que completa o grafo original a fim de se obter arcos em ambas as direções entre todos pares de vértices, compondo o conjunto total de arcos $A_c = (A \cup C)$ e o grafo completo $G_c = (V, A_c)$. Considerando o grafo G_c, define-se x_{ij} como a variável de decisão, como segue.

$$x_{ij} = \begin{cases} 1, & \text{se (i,j)} \in A', \\ 0, & \text{caso contrário.} \end{cases} \tag{1}$$

O problema do MSARDN pode ser formulado como:

$$max \sum_{(i,j) \in A} x_{ij} \tag{2}$$

sujeito a:

$$x_{ij} + x_{ji} = 1 \qquad \forall i, j \in V, i < j \tag{3}$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \qquad \forall i, j, k \in V, i \neq j, i \neq k, j \neq k \tag{4}$$

$$x_{ij} + x_{kl} \leq 1 \qquad \forall \{(i, j), (k, l)\} \in L \tag{5}$$

$$x_{ij} \in \{0, 1\} \tag{6}$$

A equação 2 representa a função objetivo do problema, cujo propósito é maximizar a cardinalidade do conjunto de arcos selecionados do grafo original, sujeito às restrições presentes em 3, 4, 5 e 6. As restrições 3, 4 e 5 definem o politopo do MSARDN. A restrição 3 previne a ocorrência de ciclos de tamanho igual a dois e pode ser provado que a restrição dada em 4 é suficiente para prevenir quaisquer outros ciclos (Reinelt, 1985). A restrição em 5 impede a ocorrência de conflitos entre pares de arcos no subgrafo obtido. A restrição dada em 6 garante a integralidade das variáveis do problema.

4. Algoritmo Aproximativo para o problema do Máximo Subgrafo Acíclico

Considerando o MSA, um algoritmo direto produz soluções com no mínimo $m/2$ arcos. Uma vez que a cardinalidade de A é um limite superior ao valor da solução ótima do problema, o algoritmo é $(1/2)$-aproximativo. Segue a descrição do algoritmo.

O algoritmo inicialmente ordena os vértices do grafo arbitrariamente, associando um índice distinto π_i a todo vértice $i \in V$. Então o algoritmo separa os arcos do conjunto A em dois subconjuntos: A_f, que armazena todos os arcos $a = (i, j) \in A$ tais que $\pi_i < \pi_j$ e A_b, o qual armazena todos os arcos $a = (i, j) \in A$ tais que $\pi_i > \pi_j$. Ambos subgrafos $G_f = (V, A_f)$ e $G_b = (V, A_b)$ são acíclicos, pois um ciclo $v_1, v_2, \ldots, v_1$ irá implicar na presença de um arco (i, j) com $\pi_i < \pi_j$ e de um arco (k, l) com $\pi_k > \pi_l$. Um dos subconjuntos, A_f ou A_b, deverá ter no mínimo $m/2$ arcos, provendo uma solução para o MSA com no mínimo metade do número de arcos da solução ótima. O algoritmo então seleciona o subconjunto (A_f ou A_b) de cardinalidade máxima.

5. Algoritmo aproximativo para o problema do Máximo Subgrafo Acíclico sob Restrições Disjuntivas Negativas

A aplicação direta do algoritmo apresentado anteriormente para o MAS ao MSARDN não garante uma solução com no mínimo metade do número de arcos de uma solução ótima. A presença das restrições disjuntivas negativas podem tornar o subconjunto A_f ou A_b inviável, mesmo ambos sendo acíclicos. Sendo assim, alguns dos arcos do subconjunto escolhido deverão ser retirados da solução a fim de torná-la viável, consequentemente podendo reduzir o número de arcos a menos da metade do número de arcos da solução ótima. Como exemplo, veja a Figura 1.

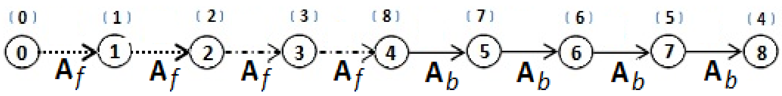

Figura 1. Grafo exemplo

Na Figura 1, os índices π_i arbitrariamente associados aos vértices do grafo estão representados sobre os mesmos. Para este exemplo, a lista de conflitos é dada por $L = \{\{(0, 1), (1, 2)\}; \{(2, 3), (3, 4)\}\}$. Seja $|A_f| = m_f$ e $|A_b| = m_b$. Analisando os índices π_i, para este exemplo temos $m_f = m_b = 4$.

Conforme o algoritmo apresentado anteriormente, este pode inicialmente escolher o conjunto A_f para compor a solução final e então, a fim de tornar o subconjunto escolhido viável respeitando-se as restrições disjuntivas negativas, retirar um dos arcos de cada par de conflitos presente em L tais que ambos os arcos do par sejam classificados como A_f.

Tome c_f (resp. c_b) como o número de conflitos em que ambos os arcos pertencem a A_f (resp. A_b). Para o caso, tem-se $c_f = 2$ e $c_b = 0$. O número de arcos retornado como solução pela aplicação do algoritmo anterior, tendo este algoritmo pré-selecionado o conjunto A_f, e após retirar os arcos que tornam a solução inviável, será $m_f - c_f = 4 - 2 = 2$ arcos. A solução ótima, facilmente identificada neste grafo, tem valor $m_{opt} = 6$ arcos. Sendo assim, o valor da solução retornada pela aplicação direta do algoritmo para este exemplo é $(1/3)$ da solução ótima.

No novo algoritmo proposto neste artigo, dado o grafo $G = (V, A)$, os vértices em V são ordenados arbitrariamente e os arcos em A são classificados nos subconjuntos A_f e A_b, como no algoritmo apresentado na seção anterior. A única diferença entre os dois algoritmos está na seleção do subconjunto considerado para encontrar a solução.

Considere $\bar{G}(A_f)$ o subgrafo de $\bar{G}$ induzido pelo conjunto de vértices A_f e $\bar{G}(A_b)$ o subgrafo de $\bar{G}$ induzido pelo conjunto de vértices A_b. Tome α_f como a cardinalidade do máximo conjunto independente em $\bar{G}(A_f)$, e α_b como a cardinalidade do máximo conjunto independente em $\bar{G}(A_b)$.

Observe que no máximo α_f (resp. α_b) vértices de $\bar{G}(A_f)$ (resp. $\bar{G}(A_b)$) podem estar simultaneamente em uma solução viável, considerando as restrições disjuntivas negativas. Portanto, o algoritmo seleciona A_f se $\max(\alpha_f, \alpha_b) = \alpha_f$ ou A_b caso contrário. A solução do algoritmo é então obtida computando-se o máximo subconjunto independente sobre o grafo induzido pelo subconjunto selecionado no grafo $\bar{G}$. Máximos subconjuntos independentes podem ser computados em tempo polinomial para várias classes de grafos de conflitos $\bar{G}$, incluindo grafos claw-free (Sbihi, 1980) e perfeitos (Grötschel et al., 1988).

5.1. Fator de Aproximação

Considerando a descrição do algoritmo previamente apresentado, introduz-se o seguinte teorema: dados os grafos $G = (V, A)$ e $\bar{G} = (A, E)$, o procedimento é um algoritmo $(1/2)$-aproximativo para o MSARDN. Segue na sequência a prova do teorema.

Seja I o máximo conjunto independente em $\bar{G}$ e $\alpha_{\bar{G}}$ a sua cardinalidade. Seja $I_f = I \cap A_f$ (resp. $I_b = I \cap A_b$). Considere m_{opt} o valor da função objetivo de uma solução ótima para o MSARDN e observe que $m_{opt} \leq \alpha_{\bar{G}}$. Na sequência, mostra-se que o número de arcos na solução produzido pelo algoritmo apresentado anteriormente é no mínimo $\alpha_{\bar{G}}/2$.

Note que $\alpha_f \geq |I_f|$, já que I_f é conjunto independente de $\bar{G}(A_f)$, e $\alpha_b \geq |I_b|$. Então $\alpha_f + \alpha_b \geq |I_f| + |I_b| = \alpha_{\bar{G}}$. Isto implica que $\alpha_f \geq \alpha_{\bar{G}}/2$ ou $\alpha_b \geq \alpha_{\bar{G}}/2$. O número de arcos na solução produzido pelo algoritmo apresentado é equivalente a $\max(\alpha_f, \alpha_b)$, e portanto é maior que ou igual a $\alpha_{\bar{G}}/2$. Consequentemente, o procedimento proposto neste artigo é um algoritmo $(1/2)$-aproximativo para o MSARDN.

6. Conclusão

Foi introduzido neste artigo um algoritmo aproximativo para o MSARDN. O algoritmo matém o fator $(1/2)$-aproximativo de um algoritmo clássico para tratar o MSA, e poderá ser resolvido em tempo polinomial sempre que o problema do máximo conjunto independente seja polinomial no grafo de conflitos.

Pesquisas futuras incluem o estudo do MSARDN sob outros tipos de grafos de conflitos e do MAS sob restrições disjuntivas positivas, em que, para um dado par de arcos, qualquer solução ótima deve conter no mínimo um arco de cada par. Devido a estas restrições, não é garantido a existência de uma solução viável para este problema, também classificado como NP-Difícil por ter o MSA como um caso especial, o que justifica sua abordagem por meio de técnicas heurísticas. Propõe-se o desenvolvimento de uma heurística cujo objetivo preliminar seja obter uma solução inicial viável, caso esta exista, e então proceder à maximização do conjunto de arcos $A' \subseteq A$ do grafo $G = (V, A)$ tal que o subgrafo $G' = (V, A')$ seja acíclico e obedeça às restrições disjuntivas positivas.

Agradecimentos

Os autores agradecem ao CNPq e a CAPES pelo auxílio financeiro a esta pesquisa.

Referências

Chaovalitwongse, W., Oliveira, C., Chiarini, B., Pardalos, P., and Resende, M. (2011). Revised grasp with path-relinking for the linear ordering problem. In *J. Comb. Optim.*, volume 22, pages 572–593.

Darmann, A., Pferschy, U., Schauer, J., and Woeginger, G. (2011). Paths, trees and matchings under disjunctive constraints. In *Discrete Applied Mathematics*, volume 159, pages 1726–1735.

Garcia, C., Pérez-Brito, D., Campos, V., and Martí, R. (2006). Variable neighborhood search for the linear ordering problem. In *Comput. Oper. Res.*, volume 33, pages 3549––3565.

Garey, M. and Johnson, D. (1979). *Computers and intractability: a quide to the theory of NP-completeness*. Freeman, New York.

Grötschel, M., Lovász, L., and Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag.

Jünger, M. (1985). *Polyhedral combinatorics and the acyclic subdigraph problem*, volume 7. Heldermann, Berlin.

Karp, R. (1975). Reducibility among combinatorial problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer. Computations*. Plenum Press, New York.

Laguna, M., Martí, R., and Campos, V. (1999). Intensification and diversification with elite tabu search solutions for the linear ordering problem. In *Comput. Oper. Res.*, volume 26, pages 1217–1230.

Martí, R., Reinelt, G., and Duarte, A. (2012). A benchmark library and a comparison of heuristic methods for the linear ordering problem. In *Comput. Optim. Appl.*, volume 51, pages 1297–1317.

Mitchell, J. (1997). Computational experience with an interior point cutting plane algorithm. In Rensellaer Polytechnic Institute, Troy, N. U., editor, *Tech. Rep.*, pages 12180–3590. Mathematical Sciences.

Newman, A. (2000). Approximating the maximum acyclic subgraph. In of Technology, M. I., editor, *Master Thesis*.

Reinelt, G. (1985). *The linear ordering problem: algorithms and applications*, volume 8. Heldermann, Berlin.

Sbihi, N. (1980). Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. In *Discrete Mathematics*, volume 29, pages 53–76.

Schiavinotto, T. and Stützle, T. (2004). The linear ordering problem: instances, search space analysis and algorithms. In *J. Math. Model. Algorithms*, volume 3, pages 367––402.