

DOIS PROBLEMAS DE LAYOUT EM GRAFOS CATERPILLAR

Carolina S. de Carvalho

PPGI - Universidade Federal do Rio de Janeiro
CP 68530 - Ilha do Fundão - Rio de Janeiro - RJ - CEP 21941-590
carolinaszkruc@gmail.com

Lilian Markenzon

NCE - Universidade Federal do Rio de Janeiro
Av. Athos da Silveira, 274, CCMN, Cidade Universitaria, RJ, 21941-611
markenzon@nce.ufrj.br

Luziane F. de Mendonça

DCC - Universidade Federal do Rio de Janeiro
CP 68530 - Ilha do Fundão - Rio de Janeiro - RJ - CEP 21941-590
luziane@dcc.ufrj.br

RESUMO

Problemas de layout de grafos são problemas de otimização combinatória cujo objetivo é encontrar um layout linear de um grafo de tal forma que um certo custo seja otimizado. São conhecidos na literatura resultados eficientes para esses problemas em algumas classes de grafos. Neste trabalho consideramos uma importante subclasse de árvores, os grafos caterpillar, e apresentamos soluções ótimas de complexidade linear para dois problemas de layout de grafos: separação de vértices e perfil.

PALAVRAS CHAVE. Grafos caterpillar, Separação de vértices, Perfil.

Área Principal: Teoria e Algoritmos em Grafos

ABSTRACT

Graph layout problems are combinatorial optimization problems whose goal is to find a linear layout for a graph in such way that the cost is optimized. For these problems, efficient solutions for some classes of graphs have been known in the literature. In this work we consider an important subclass of trees, the caterpillars, and we present optimal solutions with linear time complexity for two problems: the vertex separation problem and the profile problem.

KEYWORDS. Caterpillar, Vertex separation, Profile.

Main Area: Theory and Algorithms in Graphs

1. Introdução

Problemas de layout de grafos são problemas de otimização combinatória cujo objetivo é encontrar um layout linear de um grafo de tal forma que um certo custo seja otimizado. Um *layout linear* é uma rotulação dos vértices de um grafo com inteiros sequenciais e distintos e foi também chamado na literatura de *ordem linear* (Adolphson e Hu (1973)) ou uma *numeração* (Chinn *et al.* (1982)). Dois trabalhos com levantamentos cuidadosos dos resultados existentes na área são os *surveys* de Díaz *et al.* (2002) e Petit (2011).

Grafos caterpillar constituem uma importante subclasse de árvores e, devido a suas propriedades estruturais são muito estudados no desenvolvimento de algoritmos eficientes. Neste trabalho apresentamos a resolução de dois problemas de layout para essa classe: o problema da separação de vértices e o problema da determinação do perfil do grafo. O primeiro possui aplicação em circuitos VLSI e o segundo é utilizado para melhorar o desempenho de operações em sistemas de equações não lineares. Existem soluções polinomiais para a resolução dos dois problemas em árvores; entretanto, a solução para caterpillar se apresenta mais eficiente e bastante interessante. Para o problema da separação de vértices utilizamos um caso particular dos resultados obtidos por Ellis *et al.* (1994). Para o problema do perfil apresentamos uma solução original, obtida a partir da observação de uma sequência de vértices previamente gerada e que é provada ser a sequência ótima.

2. Conceitos Básicos

Assume-se a familiaridade com os conceitos básicos de grafos que podem ser encontrados em Diestel (2000). Nesta seção alguns conceitos são revistos.

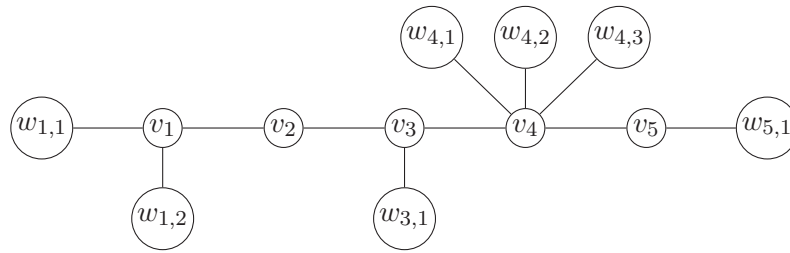
Seja $G = (V, E)$ um grafo com *ordem* $|V| = n > 0$ e *tamanho* $|E| = m$. Neste trabalho, todos os grafos considerados são conexos. A *vizinhança de um vértice* $v \in V$ é o conjunto $N(v) = \{w \in V \mid vw \in E\}$ e sua *vizinhança fechada* é $N[v] = N(v) \cup \{v\}$. O *subgrafo induzido* $G[S]$ é um subgrafo de G que tem S como conjunto de vértices e possui todas as arestas de E com extremidades em S e $S \subseteq V$.

Um *layout linear* de um grafo $G = (V, E)$, com $|V| = n$ vértices, é uma função bijetiva $L : V \rightarrow [n] = \{1, 2, 3, \dots, n\}$.

Dado um grafo $G = (V, E)$, $\mathcal{L} = \{L_1, L_2, L_3, \dots, L_l\}$ denota o conjunto de todos os layouts de G . Seja L um layout de um grafo G e i um inteiro fixo em $[n]$; definimos os conjuntos $\mathcal{E}(i, L, G) = \{u \in V : L(u) \leq i\}$ e $\mathcal{D}(i, L, G) = \{u \in V : L(u) > i\}$ como, respectivamente, os vértices à esquerda e à direita do vértice w mapeado em i (ou seja, $L(w) = i$). Dado um grafo G , a *separação de vértices* na posição i é definida por $\delta(i, L, G) = |\{u \in \mathcal{E}(i, L, G) : \exists v \in \mathcal{D}(i, L, G) \text{ com } uv \in E\}|$. Um *layout* L é representado por uma sequência de pares constituídos por vértices do grafo e suas posições em L :

$$L = \{(v_1, L(v_1)), (v_2, L(v_2)), \dots, (v_n, L(v_n))\}.$$

Um *caminho simples* é uma sequência $\langle v_0, v_1, \dots, v_p \rangle$, $p \geq 0$, de vértices distintos tal que $v_i v_{i+1} \in E$ para $0 \leq i \leq p$. O *comprimento* do caminho é p . Um *grafo caterpillar* (ou simplesmente caterpillar) é uma árvore T , com pelo menos 3 vértices, que verifica a seguinte condição: quando eliminamos todas as folhas de T obtemos um caminho de comprimento não negativo. O caminho assim obtido no caterpillar é chamado de *espinha dorsal* do caterpillar. Vamos denotar por v_i , $i = 1, \dots, n_1$, os vértices que fazem parte do caminho e por $w_{i,j}$ os vértices folhas, onde i indica o vértice v_i da espinha dorsal ao qual a folha é adjacente, j variando de 1 até o número de folhas adjacentes a v_i . O número de vértices do caterpillar é então $n = n_1 + n_2$, onde n_2 é a sua quantidade total de folhas. A Figura 1 ilustra a notação utilizada.


 Figura 1: Exemplo de um caterpillar G

3. Problema I: Separação de Vértices

O problema da separação de vértices foi introduzido por Lengauer (1981), que definiu o que chamou um *jogo de separação de vértices*. Consideramos aqui o mesmo conceito, em termos de layouts lineares. A versão de decisão do problema de separação de vértices para grafos em geral foi provada no mesmo trabalho ser um problema NP-completo.

Seja $vs(L, G) = \max_{i \in [n]} \delta(i, L, G)$, onde $[n] = \{1, 2, \dots, n\}$.

Dado um grafo $G = (V, E)$ o problema da *determinação da separação de vértices* consiste em encontrar um layout L^* tal que

$$vs(L^*, G) = \min_{L \in \mathcal{L}} vs(L, G)$$

sendo \mathcal{L} o conjunto de todos os layouts de G . O valor $vs(L^*, G)$ é denotado $\text{MINVS}(G)$.

Seja G um grafo (Figura 2) e sejam L_1 e L_2 dois de seus layouts (Figura 3). Um dos layouts ótimos do grafo G é o L_1 .

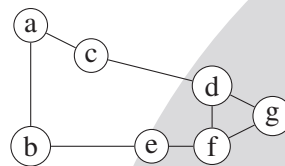
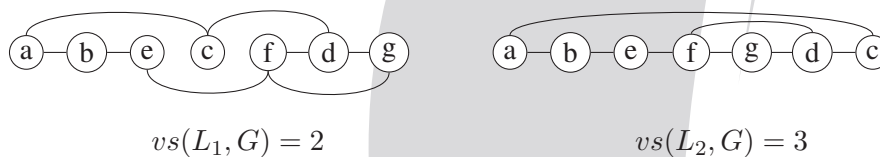

 Figura 2: Grafo G


Figura 3: Layouts

Para algumas classes de grafos existem soluções em tempo polinomial. Díaz *et al.* (2002) e Petit (2011) listam alguns desses resultados. Bodlaender e Mohring (1993) mostraram um algoritmo com complexidade de tempo de $O(n)$ que determina a largura de caminho para cografos. Para separação de vértices dos grafos grade com n dimensões existem resultados com complexidade de tempo de $O(n^2)$ (Bollobás e Leader (1991)).

Ellis *et al.* (1994) apresentaram um algoritmo que computa a separação de vértices e o layout ótimo para árvores com complexidade de tempo $O(n \log n)$. Ainda para árvores, Skodinis (2000) estabelece um algoritmo com tempo de complexidade de $O(n)$. Ellis e Markov (2004) apresentaram um algoritmo, com complexidade de tempo de $O(n \log n)$, que determina a separação

de vértices e o layout ótimo para grafos unicyclics (grafo composto por uma árvore acrescida de uma aresta). São usados no algoritmo os métodos descritos em Ellis *et al.* (1994) e Skodinis (2000) para árvores. Algum tempo depois, Chou *et al.* (2006) modificaram o algoritmo dado por Ellis e Markov (2004) chegando assim em uma complexidade de tempo de $O(n)$. Kawasaki e Justel (2005) mostraram que a separação de vértices de uma árvore binária cheia T_h de altura h (árvore binária na qual o número de vértices é $2^{h+1} - 1$) está relacionado diretamente com sua altura h .

Concentramos nosso estudo em árvores. Vamos denotar $T[x]$ uma árvore enraizada pelo vértice x e $vs(T[x])$ a separação de vértices dessa árvore. Chamamos de *subárvores induzidas* por um vértice x as subárvores de $T[x]$ que são formadas ao se retirar o vértice x da árvore.

O algoritmo de Ellis *et al.* (1994) considera recursivamente as subárvores induzidas de nível mais alto (as folhas) até chegar na árvore T de nível mais baixo (nível 0) onde se encontra a raiz, combinando os resultados. Nesse algoritmo são utilizados o conceito de vértice crítico, o Teorema 1 e o Corolário 2, vistos a seguir.

Um vértice x é k -crítico em uma árvore enraizada T se e somente se $vs(T[x]) = k$ e existem, apenas, dois filhos y e z de x tal que $vs(T[y]) = vs(T[z]) = k$.

Teorema 1. *Seja T uma árvore e seja $k \geq 1$. Então $vs(T) \leq k$ se e somente se para todo vértice x em T no máximo duas subárvores induzidas por x possuem separação de vértices k e todas as outras subárvores possuem separação de vértices menor ou igual a $k - 1$.*

Corolário 2. *Seja $T[u]$ uma árvore com raiz u dentro de uma árvore enraizada T contendo filhos v_1, \dots, v_d e $k = \max_i \{vs(T[v_i])\}$. Podemos afirmar que:*

1. *Se mais de duas das árvores $T[v_i]$ possuem separação de vértices k , então $vs(T[u]) = k + 1$.*
2. *Se exatamente duas das árvores $T[v_i]$ possuem separação de vértices k e pelo menos uma contém um vértice k -crítico, então $vs(T[u]) = k + 1$.*
3. *Se exatamente duas das árvores $T[v_i]$ possuem separação de vértices k e nenhuma delas contém um vértice k -crítico, então $vs(T[u]) = k$.*
4. *Se exatamente uma das árvores $T[v_i]$ possui separação de vértices k e ela contém um vértice, x , k -crítico, e $vs(T[u, x]) = k$, então $vs(T[u]) = k + 1$.*
5. *Se exatamente uma das árvores $T[v_i]$ possui separação de vértices k e ela contém um vértice, x , k -crítico, e $vs(T[u, x]) < k$, então $vs(T[u]) = k$.*
6. *Se exatamente uma das árvores $T[v_i]$ possui separação de vértices k e ela não contém um vértice k -crítico, então $vs(T[u]) = k$.*

A inicialização do algoritmo considera que uma árvore com um único vértice tem separação de vértices zero e uma árvore possui separação de vértices igual a 1 se e somente se ela contém no mínimo uma aresta e não contém subárvores com separação de vértices maior que 1.

Como um caterpillar é uma árvore, para determinar sua separação de vértices foram utilizados os seis casos descritos no Corolário 2. Foi constatado que independentemente do caterpillar considerado sua separação de vértices é sempre constante, como provado no teorema a seguir.

Teorema 3. *Se G é um caterpillar então $\text{MINVS}(G) = 1$.*

Demonstração. Seja G um caterpillar. Com o intuito de utilizar o Corolário 2, é necessário enraizar o caterpillar G por um de seus vértices. Podemos enraizar G pelos vértices da sua espinha dorsal (vértices das extremidades - caso 1 - ou intermediários - caso 2), ou por uma das folhas (caso 3).

Caso 1: Enraizando em v_1 ou em v_{n_1}

Suponha o caterpillar G enraizado por v_{n_1} . Para o vértice v_1 temos $vs(T[v_1]) = 1$, pois a separação de vértices das folhas é 0 e uma árvore onde todas suas subárvores induzidas pela raiz são folhas, possui separação de vértices igual a 1.

Considere o vértice v_2 e suas subárvores, $T[v_1], T[w_{2,1}], \dots, T[w_{2,j}]$. Todas, exceto $T[v_1]$, são folhas. Logo $vs(T[w_{2,1}]) = \dots = vs(T[w_{2,j}]) = 0$. Como $T[v_1] = 1$, então $vs(T[v_2]) = 1$ pelo caso 6 do Corolário 2, pois ela possui apenas uma subárvore induzida com separação de vértices igual a k .

Por raciocínio análogo, podemos concluir que $T[v_3] = T[v_4] = \dots = T[v_{n_1-1}] = 1$. Por fim, a subárvore $T[v_{n_1-1}]$ é induzida pelo vértice v_{n_1} , onde todas as outras subárvores induzidas por esse vértice são folhas. Logo, $vs(T[v_{n_1}]) = 1$.

Resultado análogo pode ser obtido se o caterpillar G foi enraizado em v_1 .

Caso 2: Enraizando em $v_i, 1 < i < n_1$.

Ao enraizar a árvore por um vértice $v_i, 1 < i < n_1$, esse vértice induz duas subárvores que não são folhas, $T[v_{i-1}]$ e $T[v_{i+1}]$. Essas subárvores são idênticas àquela estudada no caso 1. Logo, $vs(T[v_{i-1}]) = vs(T[v_{i+1}]) = 1$. Com isso $T[v_i]$ se enquadra no caso 3 do Corolário 2, e portanto, $vs(T[v_1]) = 1$.

Caso 3: Enraizando por uma folha $w_{i,j}$

Considere a árvore enraizada por uma folha $w_{i,j}$. Como cada $w_{i,j}$ esta ligado a apenas um vértice v_i , existe somente uma subárvore induzida pela raiz.

Quando enraizamos por uma folha que esta ligada a um dos vértices das extremidades da espinha dorsal ($i = 1$ ou $i = n_1$) temos um caminho, como no primeiro caso provado.

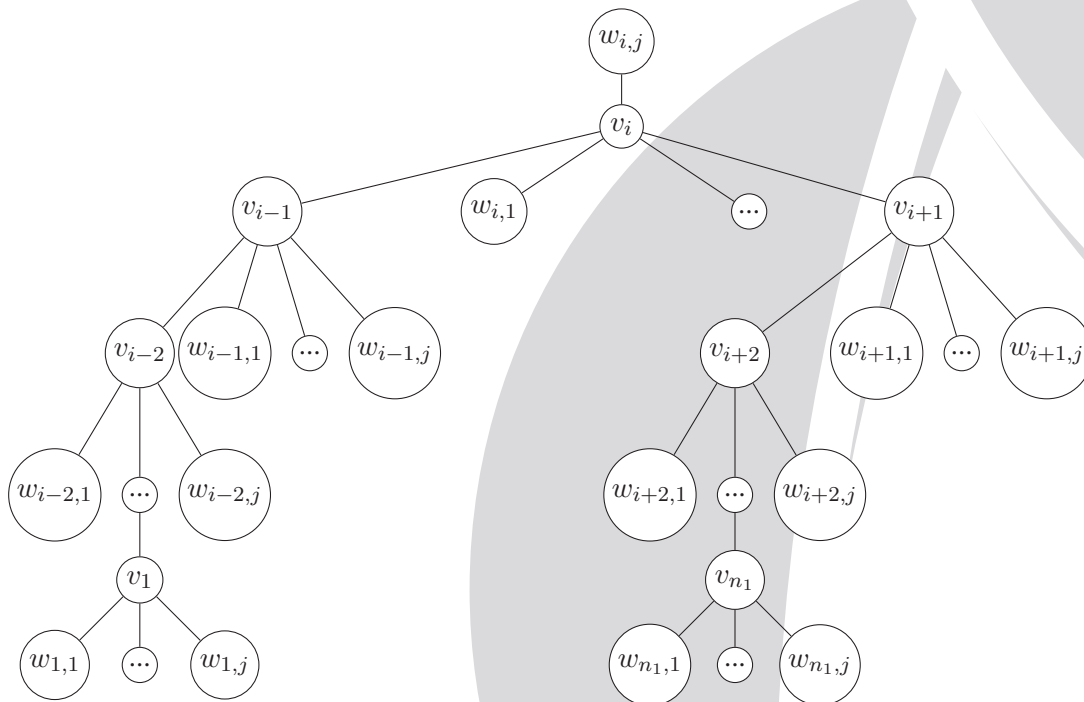


Figura 4: Árvore enraizada pelo vértice: folha $w_{i,j}$.

Já ao enraizar por qualquer outra folha ($i = 2, \dots, n_1 - 1$) a raiz induz uma subárvore, que por sua vez, induz duas subárvores com separação de vértices igual a 1, como a do segundo caso provado. Portanto o vértice ligado à raiz será crítico. Como a raiz possui uma subárvore com separação de vértices igual a 1 e a árvore formada ao se retirar a subárvore enraizada pelo vértice crítico tem separação de vértices menor que 1, então temos o caso 5 do Corolário 2. Logo $vs(T[w_{i,j}]) = 1$. □

A Figura 5 mostra um dos layouts ótimos do caterpillar da Figura 1. Observe que o layout tem na primeira posição o primeiro vértice da espinha dorsal, v_1 . Nas posições seguintes estão todas as folhas de v_1 . Em seguida vem v_2, v_3 e todas as folhas de v_3 . Essa ordenação de vértice da espinha dorsal seguido de suas folhas segue até o fim do layout. Temos então uma regra para determinar um dos layouts ótimos de um caterpillar: um vértice da espinha dorsal seguido de suas folhas adjacentes, começando o layout por um dos vértices das extremidades e respeitando a ordem de adjacência da espinha dorsal. É imediato concluir que esta construção pode ser efetuada com complexidade de tempo de $O(n)$.

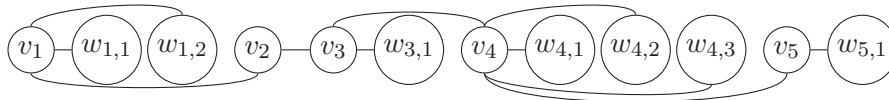


Figura 5: Layout ótimo do caterpillar

4. Problema II: Perfil

A determinação do perfil de um grafo foi originalmente proposta como uma abordagem para reduzir os requisitos de espaço para armazenamento de matrizes esparsas (Tewarson (1973)), sendo também usado para melhorar o desempenho das operações em sistemas de equações não lineares. O problema é equivalente ao problema *SumCut* (Agrawal *et al.* (1991)), outro problema de layout em grafos.

$$\text{Seja } PR(L, G) = \sum_{u \in V} \left(L(u) - \min_{v \in N[u]} L(v) \right).$$

Dado um grafo $G = (V, E)$, o problema de *determinação do perfil* de G consiste em encontrar um layout L^* tal que

$$PR(L^*, G) = \min_{L \in \mathcal{L}} PR(L, G),$$

sendo \mathcal{L} o conjunto de todos os layouts de G . O valor $PR(L^*, G)$ é denotado $\text{MINPR}(G)$,

Para os dois layouts L_1 e L_2 apresentados na Figura 3, tem-se $PR(L_1, G) = 11$ e $PR(L_2, G) = 12$. Somente o layout L_1 é ótimo.

O problema da determinação do perfil de um grafo é NP-completo (Garey e Johnson (1979)). No entanto, em algumas classes o problema pode ser resolvido em tempo polinomial. Kuo e Chang (1994) apresentaram um algoritmo com complexidade de tempo polinomial para encontrar o perfil ótimo de árvores, e Lin e Yuan (1994) propuseram vários algoritmos polinomiais para encontrar a solução ótima do perfil para caminhos, rodas, grafos bipartidos completos e árvores com diâmetro 4.

O primeiro resultado aqui apresentado diz respeito ao perfil de grafos quaisquer. Este resultado será utilizado na determinação ótima do perfil para caterpillars.

Teorema 4. *Seja G um grafo conexo e L um layout de G . Então*

$$PR(L, G) \geq m.$$

Demonstração. Seja $L = \{(v_1, 1), (v_2, 2), \dots, (v_n, n)\}$ um layout de G . Sejam: $N^-(v_i) = \{w \in N[v_i] | L(w) < i\}$ e $N^+(v_i) = \{w \in N[v_i] | L(w) > i\}$. Considerando o perfil L , $PR(L, G)$ resulta do somatório de n parcelas p_1, p_2, \dots, p_n tais que:

$$p_i = L(v_i) - \min_{w \in N^-[v_i]} L(w).$$

Como $v_i \in N[v_i]$, podemos considerar inicialmente $p_i \geq 0$. Ao determinamos a parcela p_i , só serão então considerados os vértices $w \in N^-(v_i)$, isto é, somente os w com posições inferiores a v_i em L . E mais, quanto maior a cardinalidade de $N^-(v_i)$, menor é a posição em que se encontra w com $L(w)$ mínimo. Então, $p_i \geq |N^-(v_i)|$.

Logo,

$$PR(L, G) \geq 0 + p_2 + p_3 + \dots + p_n$$

$$PR(L, G) \geq 0 + |N^-(v_2)| + |N^-(v_3)| + \dots + |N^-(v_n)|.$$

Sabemos que $\sum_{v_i \in V} N(v_i) = 2m$. Como cada aresta vw aparece em $N(v)$ e $N(w)$, e no cálculo do perfil só aparece uma vez (em $N^-(v)$) tem-se que $PR(L, G) \geq m$. \square

Tratando especificamente de árvores, Kuo e Chang (1994) apresentaram um algoritmo com complexidade de tempo $O(n^{1.722})$ para resolver o problema. Esse artigo trata a existência do que é chamado um *caminho básico* na árvore, que possui os centróides da árvore. O algoritmo é executado recursivamente para as subárvores da árvore.

Os caterpillars serão aqui tratados de forma diferente. Será construído um layout particular para os mesmos, que denotaremos L' , da seguinte forma: primeiramente posicionamos os vértices da espinha dorsal, v_i , na ordem em que eles aparecem no caterpillar; em seguida posicionamos as folhas adjacentes a v_i à esquerda de v_i . O layout L' será chamado *layout pré-folhas*. A Figura 6 mostra o layout L' para o grafo da Figura 1.

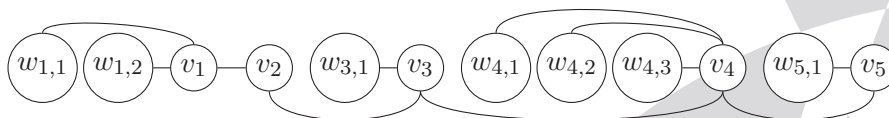


Figura 6: Layout L' do grafo G

O valor de $\text{MINPR}(G)$ um caterpillar é dado no teorema a seguir.

Teorema 5. *Seja G um caterpillar. Então $\text{MINPR}(G) = n - 1$.*

Demonstração. Seja G um caterpillar e L' seu layout pré-folhas.

Analisando as folhas:

Por construção, as folhas adjacentes a v_i estão à sua esquerda em L' . Então $N[w_{i,j}] = \{w_{i,j}\}$. Logo,

$$L(w_{ij}) - L(w_{ij}) = 0$$

Analisando os vértices da espinha dorsal:

Para os vértices da espinha dorsal temos:

$$L(u) - \min_{v \in N[u]} L(v) = \begin{cases} L(v_1) - 1, & \text{para } i = 1, \\ L(v_i) - L(v_{i-1}), & \text{para } i = 2, 3, \dots, n_1. \end{cases}$$

Logo,

$$\sum_{u \in V} \left(L(u) - \min_{v \in N[u]} L(v) \right) = L(v_1) - 1 + \sum_{i=2}^{i=n_1} L(v_i) - L(v_{i-1}).$$

Observe que $L(v_1) - 1$ é igual ao número de folhas adjacentes a v_1 . E $L(v_i) - L(v_{i-1})$ é o número de folhas adjacentes a v_i mais 1. Logo teremos o somatório do número de folhas adjacentes

a cada um dos vértices da espinha dorsal, que é igual a n_2 . A partir de v_2 adicionamos 1 ao número de folhas adjacentes, com $n_1 - 1$ parcelas (índice de 2 até n_1). Temos então:

$$\sum_{u \in V} \left(L(u) - \min_{v \in N[u]} L(v) \right) = n_2 + n_1 - 1 = n - 1.$$

Logo, $PR(L', G) = n - 1$ e pelo Teorema 4 este valor é mínimo; o layout L' é um layout de custo ótimo. \square

O Teorema 5 mostra que o layout pré-folhas é ótimo. É imediato concluir que a construção deste layout é simples e tem complexidade de tempo linear.

5. Conclusão

Neste trabalho foram apresentadas soluções particulares para a classe dos grafos caterpillar de dois problemas de layout em grafos, separação de vértices e perfil. Levando-se em conta as boas soluções encontradas pretendemos continuar a pesquisa abordando outros problemas de layout, como por exemplo o arranjo linear mínimo.

Agradecimentos: Os autores agradecem ao CNPq (Processos 473798/2012-3 e 304676/2013-6) pelo financiamento parcial desta pesquisa.

Referências

- Adolphson, D., Hu, T.C.** (1973), Optimal linear ordering, *SIAM J. Appl. Math.*, 25, 403-423.
- Agrawal, A., Klein, P., Ravi, R.** (1991), Ordering problems approximated: single-processor scheduling and interval graph completion, *Automata, Languages and Programming*, 510, 751-762.
- Bodlaender, H.L., Mohring, R.H.** (1993), The pathwidth and treewidth of cografos, *SIAM J. Discrete Math.*, 6, 181-188.
- Bollobás, B., Leader, I.** (1991), Edge- isoperimetric inequalities in the grid, *Combinatorica*, 4, 299-314.
- Chinn, P., Chvátalová, J., Dewdney, A., Gibbs, N.** (1982). The bandwidth problem for graphs and matrices, *A survey. Journal of Graph Theory*, 6, 223-254.
- Chou, H., Ko, M., Ho, C., Chen, G.** (2006), On the vertex separation of unicyclic graphs, *Ann. of the 23rd Workshop on Combinatorial Mathematics and Computation Theory*.
- Díaz, J., Petit, J., Serna, M.** (2002), A survey of graph layout problems, *ACM Computing Surveys*, 34, 314-356.
- Diestel, R.** (2000), *Graph theory*, 2nd edition, Springer Verlag, New York.
- Ellis, J. A., Sudborough, I. H., Turne, J. S.** (1994), The vertex separation and search number of a graph, *Inform. and Comput.*, 113, 59-79.
- Ellis, J., Markov, M.** (2004), Computing the vertex separation of unicyclic graphs, *Inform. and Comput.*, 192, 123-161.
- Garey, M.R., Johnson, D.S.** (1979), *Computers and intractability: a guide to the theory of NP-completeness*, 1nd edition, W. H. Freeman and Co., New York.
- Kawasaki, S.K., Justel, C.M.** (2005), Determinação de vertex separation para árvores binárias cheias, *Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional*, 2005. v. 1. p. 2343-2352.
- Kuo, D., Chang, G. J.** (1994), The profile minimization problem in trees, *SIAM J. Comput.*, 23, 71-81.
- Lengauer, T.** (1981), Black-white pebbles and graph separation, *Acta Inform.*, 16, 465-475.
- Lin, Y., Yuan, J.** (1994), Profile minimization problem for matrices and graphs, *Acta Math. Appl. Sinica, English-Series, Yingyong Shuxue-Xuebas* 10, 107-112.
- Petit, J.** (2011), Addenda to the survey of layout problems, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 105, 177-201.

Skodinis, K. (2000), Computing optimal linear layouts of trees in linear time, *Lecture Notes in Comput. Sci.*, 1879, 403-414.

Tewarson, J., R. (1973), *Sparse matrices*, Academic Press, New York.

