A EFICIÊNCIA POLIONOMIAL DO SIMPLEX PARA REDES

Aplicação em um problema do caminho mais curto

Carlos Eduardo Varejão Marinho
Doutorando em Engenharia de Produção - UENF

Antonio José dos Santos Neto
Mestre em Engenharia de Produção - UENF

RESUMO ***Neste trabalho é apresentado um algoritmo simplex para rede de complexidade*** **O(nm)** ***que encontra uma árvore de caminhos mais curtos, de um nó para todos os outros nós em uma rede direcionada, de*** **n** ***nós e*** **m** ***arcos, ou encontra um ciclo negativo. O tempo de execução desse algoritmo, no pior caso, é tão rápido quanto qualquer algoritmo polinomial que resolva este problema.***

PALAVRAS-CHAVE *Algoritmo simplex para redes, complexidade, árvores de busca.*

INTRODUÇÃO

O *problema de caminho mais curto* (PCMC) que se estuda no trabalho trata de achar o(s) caminho(s) de comprimento(s) mais curto(s) de um nó para todos os outros nós em uma rede direcionada, de *n* nós e *m* arcos, e que foi apresentado por Bellman, (1958); Ford, (1956) e Moore (1957), como uma generalização do *método de rotulação permanente* proposto, anteriormente, por Dijkstra (1957), por trabalhar com distâncias negativas, que encontrando uma árvore de caminhos mais curtos, caso o problema tenha uma solução ótima; em outro caso, termina a busca quando encontra um ciclo negativo. Esse método ficou conhecido como *método de rotulação temporária* e apresenta uma complexidade de ordem *O(nm),* e, até Goldfarb (1999), considerado o mais eficiente método para resolver o problema em questão.

Neste trabalho estuda-se o PCMC como uma formulação do problema de fluxos em redes, utilizando-se uma implementação do método simplex para redes denominadas de *simplex acelerado (fast simplex)*. No método simplex acelerado o modelo de fluxos usado para formular o PCMC consegue identificar nas variáveis duais os valores das distâncias entre o nó raiz e qualquer outro nó, de qualquer caminho que os ligue. Como o problema trata da busca do caminho mais curto de um nó para todos os outros nós, o método proposto potencializa as distâncias nos arcos, em vez dos fluxos, igualando-se à eficiência do método Bellman-Ford-Moore (Marinho, 2001).

O PCMC continua sendo um assunto bastante estudado na Pesquisa Operacional devido à gama de aplicações, em diversas áreas do conhecimento, como: o estado das relações de liderança num sociedade generalizada, no planejamento de atividades, em arranjo físico e é um problema fundamental na área de transportes, seja na área de planejamento viário do sistema urbano ou do sistema interurbano, como na área de controle de tráfego. Por exemplo:

O binômio "sinal + guarda de trânsito" para o controle de tráfego nas grandes cidades tornou-se um sistema ineficiente e ineficaz. Hoje, nessas cidades, o controle do sistema viário, exige não só o controle de fluxo, mas também o fornecimento de informações ao motorista, principalmente durante os congestionamentos, com indicações de caminhos alternativos, em tempo real, para os diversos destinos de interesse do motorista, com o tempo estimado de cada percurso proposto. Para essa modernização, há a necessidade do desenvolvimento de softwares eficientes e eficazes capazes de manipular dezenas ou centenas de variáveis, e fornecer informações precisas para auxiliar o motorista na tomada de decisão. (Marinho, 2001).

Syslo (1983) cita que Dijkstra, no ano de 1957, apresentou um método para resolver o problema de transporte, isto é, uma heurística para encontrar a menor distância entre dois pontos específicos numa rede. Dijkstra denominou esses pontos de interesse como: origem *s* e destino *t*. Esse método é bastante usado na área de transportes visto que, em qualquer ponto em que o motorista estiver, dentro da malha viária, esse ponto pode ser considerado como origem; e o objetivo imediato desse motorista é chegar em algum ponto que possa ser considerado como destino. Ele permite a redução da distância entre origem e destino pela seleção do caminho entre as várias alternativas de tempo e distância à disposição do motorista.

Paralelamente a Ddijkstra, outros pesquisadores trataram do mesmo assunto, destacando-se Bellman (1958), Ford (1957) e Moore (1958) que generalizaram o método, inclusive, para trabalhar até com distâncias "negativas". O método de Bellman-Ford-Moore que, neste trabalho, é identificado como método BFM resolve o problema de encontrar o caminho mais curto entre um nó específico, chamado fornecedor, para todos os outros nós da rede, chamados atratores.

O método apresentado a seguir para resolver o PCMC apresenta uma complexidade fortemente polinomial, assemelhando-se aos métodos heurísticos (BFM), para o mesmo problema, conforme observado por Denardo e Fox (1979), Vliet (1978) e Dial (1969).

DESCRIÇÃO SIMPLEX ACELERADO

Considerar um grafo direcionado G = (N, A), com *n* nós e *m* arcos. A cada arco $(v, w) \in A$ é associado um comprimento $c_{v, w}$. O problema de caminho mais curto abordado consiste em achar uma árvore de caminhos orientados entre o nó s $\in$ N e todos os outros nós de G. A formulação matemática do problema como um problema de programação linear é:

$$Minimizar \sum_{(i, j) \in A} c_{ij} x_{ij}.$$

$$s.a$$

$$\sum_{j(i, j) \in A} x_{ij} - \sum_{j(j, i) \in A} x_{ji} = b(i), \text{ onde}$$

$$x_{ij} \geq 0 \text{ para todo } (i, j) \in A, \text{ e tal que } \sum_{i=1}^{n} b(i)=0.$$

O algoritmo simplex acelerado é uma implementação do algoritmo simplex, para o PCMC clássico, preparado por Goldfarb et al. (1990[a]), que apresenta uma complexidade fortemente polinomial, no pior caso, da ordem O *(nm)*.

A seguir é apresentado o procedimento para o simplex acelerado onde, a cada iteração, o algoritmo executa em 3 (três) etapas a seguinte rotina para selecionar um arco que deve ser adicionado na árvore geradora T^ e escolher o arco a ser retirado, atualizando a árvore e testando as condições de otimalidade da nova árvore de busca.

Primeira etapa (ordenação). Como qualquer algoritmo é estruturado como uma seqüência de passos, onde em cada passo, todos os *m* arcos do grafo são analisados com o seguinte critério: arcos com o mesmo nó cabeça são analisados em grupo. A ordem em que esses grupos são analisados é $B(w_n)$, $B(w_{n-1})$,..., $B(w_1)$, onde w_1, w_2,..., w_n são os nós cabeça numa pré-ordenação na árvore geradora T^ para o começo do *passo* inicial. Os grupos, $B(w) = \{(v,w) \in A\}$, são conjuntos de arcos que têm como nó cabeça o nó *w*. Os nós que definem os grupos são ordenados em uma pilha *(S)* montada percorrendo-se a árvore em profundidade ou pré-ordem. Outras ordenações dos nós w_i são possíveis desde que seja mantida a relação i < j se w_i é o pai de w_j.

Segunda etapa: Somente a potência do nó *h*, isto é, π^*(h)* do nó cabeça do arco (g, h) escolhido para entrar na árvore T é atualizado num pivoteamento do simplex. Esse arco é escolhido pela equação de pseudocusto reduzido $c_{vw}\pi^{\wedge} = c_{vw} + \pi^{\wedge}(v) - \pi^{\wedge}(w)$, onde os $\pi^{\wedge}(u)$, $u \in N$ são valores superestimados dos multiplicadores do simplex, associados à árvore geradora T; nomeando o nó 1 como a raiz ou, como o nó fonte. Os valores superestimados dos multiplicadores são conhecidos como *pré-multiplicadores.* (Orlin, 1995). Para um arco *(v, w)* ser candidato a entrar na árvore T a condição $c_{vw}\pi^{\wedge} < 0$ deve ser satisfeita. As potências dos nós, $\pi^{\wedge}(u)$ com $u \in N$, são referenciadas como *pré-multiplicadores,* com respeito à árvore T, enraizada no nó 1 e os pseudocustos reduzidos $c_{vw}\pi^{\wedge} \leq 0$ para todos os arcos *(v, w)* $\in$ *T. O vetor de multiplicadores do simplex* π *é um conjunto particular de pré-multiplicadores* π *que satisfazem as condições* $c_{vw}\pi^{\wedge} = 0$ *para todos os arcos (v, w)*$\in$*T e* $\pi^{\wedge}(s) = 0$.

Terceira etapa: Todos os arcos de um grupo B (w_i) são analisados, e seus pseudocustos reduzidos calculados. Se mais de um arco do conjunto B é elegível para entrar na base, então, aquele arco com pseudocusto reduzido ($c_{vw}\pi$) mais negativo é escolhido para entrar na árvore; e, um pivoteamento simplex é realizado. Quando $w_i = 1$, chegou-se ao fim de uma iteração, isto é, ao fim da pilha S. Caso contrário, é visitado o nó w_{i-1} e revisam-se os arcos do grupo $B(w_{i-1})$, o grupo predecessor do nó w_i em T^.

A regra do pivoteamento do simplex acelerado é uma versão modificada da regra de escolha do arco com a maior violação negativa (*inward most negative pivot)* proposta por Cunningham (1979) e Goldfarb et al. (1990b). Este método inclui duas técnicas para a detecção de um ciclo negativo. A primeira técnica é se a árvore T contém um ciclo de comprimento negativo, que não inclui o nó 1, então T é composta de duas ou mais componentes; e, uma delas, que inclui o nó 1, é uma árvore com menos de n nós. A segunda baseia-se no método BFM: se, durante a k-ésima iteração, (isto é, após uma revisão completa de todos os arcos da rede), as rotulações dos nós, de pelo menos $n - k$ nós diminuem, então, a rede contém um ciclo negativo (Lawler, 1976, teorema 11.2). A aplicabilidade desse critério no algoritmo simplex acelerado é justificada pelo Lema: *Uma vez que T torna-se cíclica, ela permanecerá cíclica.*

Diferente do teste para saber se o nó g é descendente do nó h, isto é, $g \in D_h$, esse critério nem sempre detecta a criação de um ciclo negativo em um pivoteamento. Entretanto, este teste é importante para estabelecer o limite do número de pivoteamentos requeridos pelo algoritmo.

A seguir é enunciada a propriedade de Goldfarb *et al*. (1999) que resume e formaliza a observação acima mencionada:

Teorema:

O algoritmo simplex para redes pode ser implementado para resolver o problema de caminho mais curto realizando no máximo (n-1) (n-2)/2 pivoteamentos com uma complexidade, de pior caso, da ordem (nm).

O SIMPLEX ACELERADO

A seguir se descreve o algoritmo simplex acelerado para redes no seguinte pseudo-código.

Início

escolha uma árvore inicial T^;

optimal ←false;

k←0;

enquanto optimal=false **fazer**
início

T^←T;
Construa a pilha S de acordo com a ordem de visita aos nós de T^;

se |S| < n **então** PARAR (T^=T contem um ciclo negativo);

calcule os multiplicadores do simplex p(u)=p^(u) correspondentes a T=T^ para todo $u \in N$;
k ←k + 1
pivots←0
optimal← true;
enquanto S≠ϕ **fazer**
início

h← nó do topo de S, e remova-o de S;
se $c_{gh}\ \pi^{\wedge} = \min\{\ c_{vh}\pi^{\wedge}/(v, h)\in T\} < 0$ **então**
início

pivot ← pivot + 1;
se h=s ou pivots ≥ n - k **então** PARAR (ciclo negativo)
senão

T ←T + {g, h} - {pred(h), h}
π^(h)?← π^(h) + $c_{gh}\pi^{\wedge}$
optimal←false;
fim;
fim;
fim;
Fim.

EXEMPLO PRÁTICO

A rede utilizada é mostrada na figura 1.

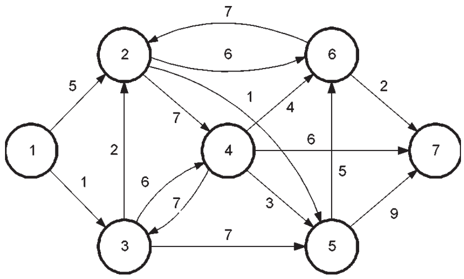

Figura 1. Rede Conexa e densidade $m/C_2^n = 0{,}762$. Exemplo.

Encontrar o caminho mais curto do nó 1, para todos os demais de G(N, A), onde |N|=7 nós e |A| = 16 arcos.

Inicia-se a resolução do problema com a abstração da árvore geradora viável T_o (1, v), mostrada na figura 2. No caso de não haver uma ligação direta entre os nós 1 e v, arcos artificiais com custos penalizados, devem ser introduzidos em G. No processo interativo esses arcos artificiais devem sair da árvore T. No caso da solução final apresentar algum arco artificial, o problema é inviável. Para garantir que esse(s) arco(s) seja(m) retirado(s) da árvore T, durante o procedimento simplex, é usual atribuir a cada arco artificial, um custo cujo valor seja maior que duas vezes a soma de todos os custos do grafo, isto é, $M = 2\sum_{(v,w) \in A} c_{vw}$.

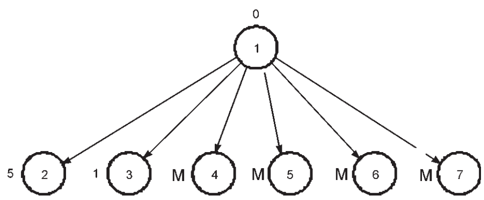

Figura 2. Árvore inicial (T_o), com os pré-multiplicadores (π) indicados ao lado de cada nó.

Para se construir a pilha S (ordem de visita aos nós de G), no passo k=0, percorre-se a árvore T^$_o$ em profundidade na busca dos filhos e, em amplitude, na busca dos irmãos, para montar a pilha S. A pilha encontrada é S= {1,2,3,4,5,6,7}.

Os arcos (v, h)ÏT, que formam os conjuntos B(w_i), com o nó cabeça *h*, são:

B(7) = {(4, 7), (5, 7), (6, 7)};

B(6) = {(2, 6), (4, 6), (5, 6)};

B(5) = {(2, 5), (3, 5), (4, 5)};

B(4) = {(2, 4), (3, 4)};

B(3) = {(4, 3)};

B(2) = {(3, 2), (6, 2)}

B(1) = {(0)}.

Com o nó que está no topo da pilha S inicia-se o método, calculando-se os custos reduzidos c$_{p^\wedge}$p dos arcos do conjunto B(7).

Como os custos reduzidos dos arcos em B(7) são todos positivos, às condições de otimalidade são satisfeitas. Portanto, nenhum dos arcos melhora o caminho do nó 1 ao nó 7.

Analisando o conjunto B(6) e calculando os dos arcos do conjunto tem-se:

Neste caso o arco (2,6), por contrariar as condições de otimalidade, é candidato a entrar na árvore. O arco (pred(h), h) = (1, 6) é o arco a ser retirado. O valor do pré-multiplicador é

Analisando o conjunto B(5) e calculando os dos arcos deste conjunto tem-se:

(o mais negativo)

Há dois arcos candidatos para serem escolhidos por contrariar as condições de otimalidade. Neste caso é escolhido o arco (2,5), por apresentar o custo reduzido mais negativo. O arco que sai é (1,5). O valor do pré-multiplicador

Analisando o conjunto B(4) e, calculando-se os dos arcos do conjunto, tem-se:

O arco escolhido para ser adicionado à árvore é (3,4). O arco a ser retirado é (1,4). O valor do pré-multiplicador

Analisando o conjunto B(3) e calculando do arco (4,3), verifica-se que este arco está na sua condição de otimalidade. Sua inclusão na árvore básica não melhora o caminho do nó 1 até ele, portanto, permanece o arco atual, isto é, o arco (1,3).

Analisando os arcos (3, 2) e (6, 2) do conjunto B(2), verifica-se que o custo reduzido do arco (3, 2) é negativo

, e do arco (6, 2) é positivo. Nesse caso, o arco (3,2) é incorporado à árvore e é retirado o arco (1,6). O valor do pré-multiplicador

.

O conjunto B(1) é sempre um conjunto vazio, por se ter considerado o nó raiz como o nó 1.

A figura 4.3 mostra a nova árvore constituída pelos novos arcos que melhoram as distâncias dos caminhos do nó 1 ao nó v.

Percorrendo-se a árvore da figura 3, constrói-se a seguinte pilha S = {1,3,2,6,5,4,7}. Os novos conjuntos $B(w_i)$ são:

B(7) = {(4,7), (5,7), (6,7)};

B(4) = {(2,4)};

B(5) = {(3,5), (4,5)};

B(6) = {(4,6), (5,6)};

B(2) = {(1,2), (6,2)}

B(3) = {(4,3)};

B(1) = {(0)}.

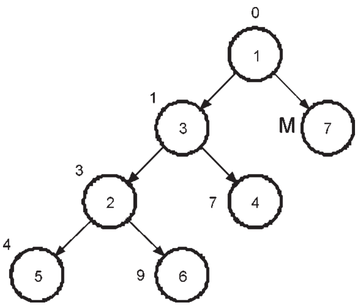

Figura 3. Árvore melhorada (T_1), com os novos pré-multiplicadores.

Aplicando a mesma rotina anterior para escolha do arco que contraria a sua condição de otimalidade, para ser incorporado a árvore, tem-se:

i) No conjunto B(7) é escolhido o arco (5, 7) para entrar na árvore T_1, e sai o arco (1, 7). O valor do pré-multiplicador . Os arcos dos conjuntos B(4), B(5), B(2) e B(3) atendem às condições de otimalidade.

ii) Do conjunto B(6) apenas o arco (5,6) contraria a sua condição de otimalidade, portanto, esse arco é adicionado em T_1 e o arco (2,6) é retirado. O pré-multiplicador

iii) O conjunto B(1)= {(0)}.

A nova árvore T_2 é mostrada na figura 4, com a indicação dos novos multiplicadores.

Da árvore T_2, constrói-se a seguinte pilha S={1,3,2,5,6,7,4}. Os conjuntos dos arcos não básicos são:

B(4) = {(2,4)};

B(7) = {(4,7), (6,7)};

B(6) = {(2,6), (4,6)};

B(5) = {(3,5), (4,5)};

B(2) = {(1,2), (6,2)}

B(3) = {(4,3)};

B(1) = {(0)}.

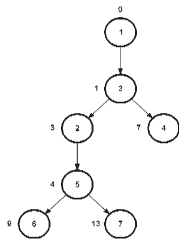

Figura 4. Árvore T_2, com a indicação dos novos pré-multiplicadores.

O arco do conjunto B(4) está na sua condição de otimalidade. Permanece em T_2, o arco atual. O arco (6,7), pertencente ao conjunto B(7), contrária à condição de otimalidade; portanto, o arco (6,7) é adicionado à árvore T_2 sendo retirado o arco (5, 7). O pré-multiplicador . Todos os arcos dos conjuntos B(5), B(2) e B(3) atendem às condições de otimalidade, permanecendo na árvore os arcos atuais.

A nova árvore T_3 é mostrada na figura 5.

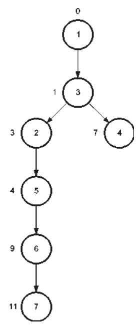

Figura 5 - Árvore T_3 com os novos pré-multiplicadores

Da nova árvore cria-se a pilha S={1,3,2,5,6,7,4} e os conjuntos B(w_i).

B(4) = {(2,4)};

B(7) = {(4,7), (5,7)};

B(6) = {(2,6), (4,6)};

B(5) = {(3,5), (4,5)};

B(2) = {(1,2), (6,2)}

B(3) = {(4,3)};

B(1) = {(0)}.

Calculando-se os custos reduzidos dos arcos pertencentes aos conjuntos B(w_i) acima, verifica-se que todos arcos atendem as condições de otimalidade; portanto, permanecem os arcos atuais e a *árvore T_3 é ótima*.

Os multiplicadores indicados representam as distâncias mínimas dos caminhos do nó 1 para cada nó *v*.

Observa-se, na tabela 1, que o número de pivoteamento permitidos pelo método para encontrar as distâncias mínimas, entre o nó raiz e todos os outros nós da rede, ou um ciclo negativo é 15. Foram realizados ao todo 7 (sete) pivoteamentos para resolver o exemplo.

K	Árvore	Nós							Pivoteameno	
		1	2	3	4	5	6	7	NMP	NPR
0	T_0	0	1	1	1	1	1	1	6	-
1	T_1	0	3	1	3	2	2	1	5	4
2	T_2	0	3	1	3	2	5	5	4	2
3	T_3	0	3	1	3	2	5	6	3	1
4	T^*	0	3	1	3	2	5	6	2	0

Tabela 1. Evolução do pivoteamento

NMP – número máximo de pivoteamentos permitidos no passo k.

NPR – número de pivoteamentos efetivamente realizados

CONCLUSÃO

A performance de um algoritmo que resolve problemas sobre redes, em geral, não depende somente da forma como o algoritmo é desenvolvido e da seqüência de passos a serem executados, mas, também, das estruturas de dados utilizadas para representar essa rede no computador. A utilização de estruturas de dados mais elaboradas melhora o tempo de execução do algoritmo. Portanto, a dualidade – estudo teórico e computacional – é necessária para tornar eficientes algoritmos que resolvem problemas de otimização em redes.

A eficiência polinomial do método simplex acelerado é explicada: primeiro, pela técnica de ordenação dos nós em uma pilha gerada a partir das árvores de busca (T). A visão sobre a análise do conjunto dos arcos não básicos que apontam para o nó topo da pilha foi o passo fundamental para a simplicidade do pivoteamento do simplex, pois, se um arco não básico do conjunto de arcos que apontam para o nó topo é candidato para entrar em T, por contrariar a condição de otimalidade, então, o arco atual que aponta para esse topo, sai da árvore. O valor do pré-multiplicador desse nó topo é calculado por $\pi\hat{}(h)? \leftarrow \pi\hat{}(h) + c_{gh}\pi^{\hat{}}$ e segundo, a garantia de que o método termina após realizar pivoteamentos, independente de encontrar ou não um ciclo negativo.

REFERÊNCIAS BIBLIOGRÁFICAS

AHUJA, R. K; MAGNANTI, T. L; ORLIN, J. B. - **Network Flows, Theory, Algorithms and Applications**. Prentice-Hall, Inc, 1993.

BELLMAN R. On a Route Problem. **Quarterly Applied Mathematics** 16 p 87-90. 1958.

CUNNINGHAM, W. A Network Simplex Method. **Math. Programming** 11. p.105-116. 1976.

DIJKSTRA, E W. A Note on Two Problems in Connexion with Graphs. **Numerishe Matematik,** vol. 1. p 269-271. 1959.

FORD, L R. Network Flow Theory. **The Rand corporation Report** p.923, Santa Monica, CA. 1956.

GOLDFARB, D. AND JIN, Z. An O(nm) Time Network Simplex Algorithm For the Shortest Path Problem. **Operations Research**, vol.47, N0 3. p 445-448. 1999.

GOLDFARB, D; HAO, J. AND KAI, S.-R. Efficient Shortest Path Simplex Algorithms. **Operations Research**, vol. 38 No 4, p 624-628. 1990a.

GOLDFARB, D; JIANXIU H; KAI S. R. Anti-Stalling Pivot Rules For The Network Simplex Algorithm**. Networks. John Wilev and Sons**, Inc. vol.20 p 79-91. 1990b.

LAWLER, E L. **Combinatorial optimization**: networks and matroids. Holt, Rinehart and Winston, New York. 1956.

MARINHO, C.E.V. **Eficiência Polinomial do Método Simplex para Redes**: Análise Sobre um Problema do Caminho mais Curto (PCMC). Tese de Mestrado - UENF. Campos dos Goytacazes, RJ- Brasil. 2001.

MOORE, Z F. The Shortest Path Through a Maze. Proc. Internat. Sympos. **On Theory of Switching**, Part II, p. 285-292. 1957.

ORLIN, B. J. A Polynomial Time Primal Network Simplex Algorithm For Minimum Cost Flows. **The Mathematical Programming Society Inc**, Published by Elsevier Science B.V-p. 109-127. 1997.

TAHA, H. A. **Operations Research**: An Introduction. Prentice Hall Inc, 6th Edition. 1997.