

# A EFICIÊNCIA DO SIMPLEX PARA REDES EM UM PROBLEMA DO CAMINHO MAIS CURTO

**Carlos Eduardo Varejão Marinho**

Universidade Estadual do Norte Fluminense Darcy Ribeiro  
Av. Alberto Lamego, 2000 - Campos dos Goytacazes - RJ  
CEP 28013-6000

[druamarinho@bol.com.br](mailto:druamarinho@bol.com.br)

**Antônio José dos Santos Neto**

Universidade Estadual do Norte Fluminense Darcy Ribeiro  
Av. Alberto Lamego, 2000 - Campos dos Goytacazes - RJ  
CEP 28013-6000

[aneto@uenf.br](mailto:aneto@uenf.br)

## Resumo

Este trabalho apresenta um algoritmo simplex para rede que encontra uma árvore de caminhos mais curtos, de um nó para todos os nós em uma rede direcionada, de  $n$  nós e  $m$  arcos ou um *ciclo negativo*. Esse algoritmo, no pior caso, é tão rápido quanto qualquer algoritmo polinomial que resolve este problema.

**Palavras-chave:** Algoritmo simplex para redes, Complexidade, Árvore de busca.

## Abstract

This work presents a simplex algorithm to find a shortest path of a node to any another node or a *negative cycle* in a directed network with  $n$  nodes and  $m$  arcs. This algorithm, in the worse case, is so fast how much any polynomial algorithm that solve this problem.

**Word-key:** Simplex algorithm network, Complexity, Search tree

## 1. INTRODUÇÃO

O problema de caminho mais curto (PCMC), apresentado neste 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. Este problema apresentado por Bellman, 1958; Ford, 1956 e Moore 1957, constituiu-se em uma generalização do método de rotulação permanente proposto, anteriormente por Dijkstra (1957), por trabalhar com distâncias negativas. Encontrando uma árvore de caminhos mais curtos, caso o problema tenha uma solução ótima; em caso contrário, termina a busca quando encontra um ciclo negativo.

Este método que ficou conhecido como *método de rotulação temporária*, apresenta uma complexidade de ordem  $O(nm)$ , e, foi considerado o método mais eficiente para resolver o problema em questão.

Goldfarb (1999) apresenta uma implementação para resolver o PCMC de um nó para todos os outros nós, que explora uma estrutura de dados para estabelecer a ordem de revisão dos nós da árvore de busca, para escolha do nó candidato a entrar na árvore e,

conseqüentemente, o arco para sair dessa árvore. Estudando-se o PCMC como uma formulação do problema de fluxos em redes, utilizando-se uma implementação do método simplex para redes denominada de *simplex acelerado* (*fast simplex*). No simplex acelerado o modelo de fluxos usado para formular o PCMC, consegue identificar nas variáveis duais os valores das distâncias entre o nó raiz a 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, entre elas, as Redes Pert/CPM, localização de pontos de conexão na rede de transporte, exemplos síntese de atividades interligadas, além, de outras aplicações na área social, industrial, etc.

## 2- DESCRIÇÃO SIMPLEX ACELERADO

Considerando 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 do 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:

$$\begin{aligned} & \text{Minimizar } \sum_{(i, j) \in A} c_{ij} x_{ij}. \\ & \text{s.a} \\ & \sum_{j(i, j) \in A} x_{ij} - \sum_{j(j, i) \in A} x_{ji} = b(i), \text{ onde } \begin{cases} b(s) = -(n-1), i=s, \text{nó fornecedor} \\ b(i) = 1, \text{ com } i \neq s, \text{nó atrator} \end{cases} \end{aligned}$$

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

O algoritmo simplex acelerado é uma implementação do algoritmo simplex, para o PCMC clássico, preparado por Goldfarb et al. (1990<sup>a</sup>) e que apresenta uma complexidade fortemente polinomial, no pior caso, da ordem  $O(nm)$ .

A seguir é apresentado o procedimento do 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 de busca  $T^A$  e escolher o arco a ser retirado. Atualizando a de árvore busca e testando as condições de *otimalidade* da nova árvore.

**Primeira etapa (ordenação).** Como qualquer algoritmo, o simplex acelerado 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  $B(w_n), B(w_{n-1}), \dots, B(w_1)$ , onde  $w_1, w_2, \dots, w_n$  são os nós cabeça numa pré-ordenação na árvore de busca  $T^A$  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 ao se percorrer 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,  $\pi^A(h)$  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 os multiplicadores do simplex, associados à árvore de busca  $T$ ; nomeando o nó 1 como a raiz (o nó raiz é caracterizado como um nó que apenas oferta fluxo no problema PCMC) 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 de busca  $T$ , no algoritmo, e os pseudocustos reduzidos  $c_{vw}^{\pi^{\wedge}} \leq 0$  para todos os arcos  $(v, w) \in T$ . *O vetor de multiplicadores do simplex  $\pi$  é um conjunto particular de pré-multiplicadores  $\pi$  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  $T^{\wedge}$ .

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 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 pivoteamento 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  $(nm)$ ”*

### 3 – O SIMPLEX ACELERADO

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

**Início**

escolha uma árvore inicial  $T^\wedge$ ;

optimal  $\leftarrow$  false;

$k \leftarrow 0$ ;

**enquanto** optimal=false **fazer**

**início**

$T^\wedge \leftarrow T$ ;

Construa a pilha  $S$  de acordo com a ordem de visita aos nós de  $T^\wedge$ ;

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

calcule os multiplicadores do simplex  $p(u)=p^\wedge(u)$  correspondentes a  $T=T^\wedge$  para todo  $u \in N$ ;

$k \leftarrow k + 1$

pivots  $\leftarrow 0$

optimal  $\leftarrow$  true;

**enquanto**  $S \neq \emptyset$  **fazer**

**início**

$h \leftarrow$  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  $\leftarrow$  pivot + 1;

**se**  $h=s$  ou pivots  $\geq n - k$  **então** PARAR (ciclo negativo)

**senão**

$T \leftarrow T + \{g, h\} - \{\text{pred}(h), h\}$

$\pi^\wedge(h) \leftarrow \pi^\wedge(h) + c_{gh}^{\pi^\wedge}$

optimal  $\leftarrow$  false;

**fim;**

**fim;**

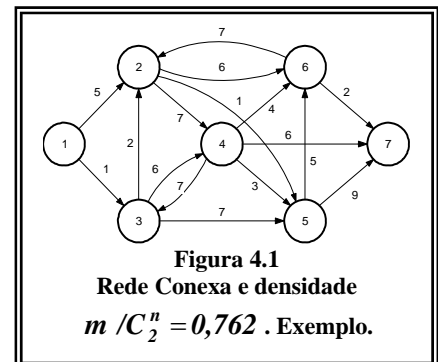
**fim;**

**Fim.**

### 4 – EXEMPLO DA EXECUÇÃO

A rede utilizada é mostrada na figura

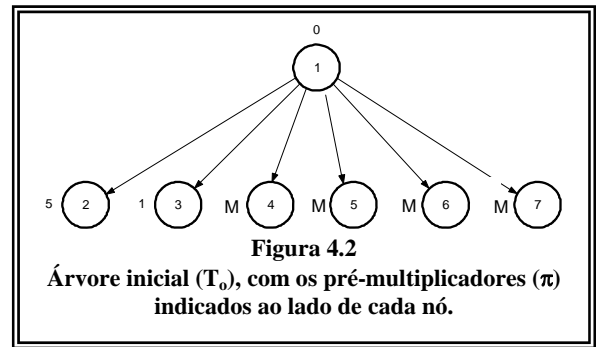
4.1 (Exemplo 2-figura 6.16 da seção 6.4 pp [231] do livro *Operations Research*. Taha, 1997).



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 de busca  $T_0(1, v)$ , mostrada na figura 4.2. 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 iterativo 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,  $M = 2 \sum_{(v,w) \in A} c_{vw}$ .

Para se construir a pilha S (ordem de visita aos nós de G), no passo  $k=0$ , percorre-se a árvore  $T^0$  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  $(v, h) \in T$ , 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  $c_p^\pi$  dos arcos do conjunto  $B(7)$ .

$$c_p^\pi(4,7) = c(4,7) + \pi(4) - \pi(7) = 6 + M - M > 0$$

$$c_p^\pi(5,7) = c(5,7) + \pi(5) - \pi(7) = 9 + M - M > 0$$

$$c_p^\pi(6,7) = c(6,7) + \pi(6) - \pi(7) = 2 + M - M > 0$$

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  $c_p^\pi$  dos arcos do conjunto tem-se:

$$c_p^\pi(4,6) = c(4,6) + \pi(4) - \pi(6) = 4 + M - M > 0$$

$$c_p^\pi(2,6) = c(2,6) + \pi(2) - \pi(6) = 6 + 5 - M < 0$$

$$c_p^\pi(5,6) = c(5,6) + \pi(5) - \pi(6) = 5 + M - M > 0$$

Neste caso o arco  $(2,6)$ , por contrariar as condições de otimalidade, é candidato a entrar na árvore. O arco  $(\text{pred}(h), h) = (1, 6)$  é o arco a ser retirado. O valor do pré-multiplicador é  $\pi^{\wedge}(6) = \pi^{\wedge}(6) + c_p^\pi(2,6) = M + 6 + 5 - M = 11$ .

Analisando o conjunto  $B(5)$  e calculando os  $c_p^\pi$  dos arcos deste conjunto tem-se:

$$c_p^\pi(2,5) = c(2,5) + \pi(2) - \pi(5) = 1 + 5 - M \ll 0 \text{ (} \ll \text{ o mais negativo)}$$

$$c_p^\pi(3,5) = c(3,5) + \pi(3) - \pi(5) = 7 + 1 - M < 0$$

$$c_p^\pi(4,5) = c(4,5) + \pi(4) - \pi(5) = 3 + M - M > 0$$

Há dois arcos candidatos para serem escolhidos por contrariar as condições de otimalidade. Neste caso é escolhido a arco  $(2,5)$ , por apresentar o custo reduzido mais

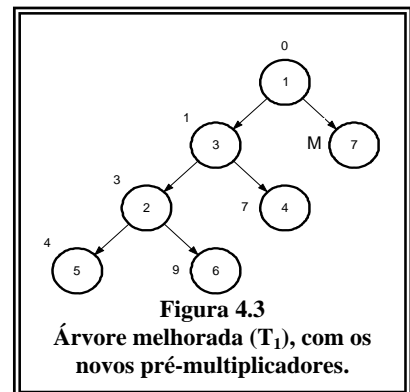
negativo. O arco que sai é (1,5). O valor do pré-multiplicador  $\pi^{\wedge}(5) = \pi^{\wedge}(5) + c_p^{\pi^{\wedge}}(2,5) = M + 1 + 5 - M = 6$ .

Analisando o conjunto B(4) e, calculando-se os  $c_p^{\pi^{\wedge}}$  dos arcos do conjunto, se tem:

$$c_p^{\pi^{\wedge}}(2,4) = c(2,4) + \pi(2) - \pi(4) = 7 + 5 - M < 0$$

$$c_p^{\pi^{\wedge}}(3,4) = c(3,4) + \pi(3) - \pi(4) = 6 + 1 - M \ll 0$$

O arco escolhido para ser adicionado à árvore é (3,4). O arco a ser retirado é (1,4). O valor do pré-multiplicador  $\pi^{\wedge}(4) = \pi^{\wedge}(4) + c_p^{\pi^{\wedge}}(3,4) = M + 6 + 1 - M = 7$



Analisando o conjunto B(3) e calculando  $c_p^{\pi^{\wedge}}$  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 ( $c_p^{\pi^{\wedge}}(3,2) = c(3,2) + \pi(3) - \pi(2) = 2 + 1 - 5 < 0$ ), 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  $\pi^{\wedge}(2) = \pi^{\wedge}(2) + c_p^{\pi^{\wedge}}(3,2) = 5 + 2 + 1 - 5 = 3$ .

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 4.3, constrói-se a seguinte pilha S = {1,3,2,6,5,4,7}. Os novos conjuntos B(w<sub>i</sub>) 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)}.

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

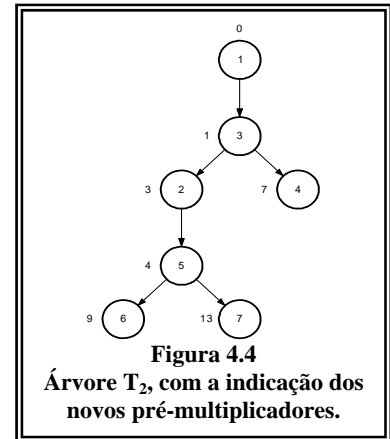
- i) No conjunto B(7) é escolhido o arco (5, 7) para entrar na árvore T<sub>1</sub>, e sai o arco (1, 7). O valor do pré-multiplicador  $\pi^{\wedge}(7) = \pi^{\wedge}(7) + c_p^{\pi^{\wedge}}(5,7) = 13$ . Os arcos dos conjuntos B(4), B(5), B(2) e B(3) atendem as condições de otimalidade.
- ii) Do conjunto B(6) apenas o arco (5,6) contraria a sua condição de otimalidade, portanto, esse arco é adicionado em T<sub>1</sub> e o arco (2,6) é retirado. O pré-multiplicador  $\pi^{\wedge}(6) = 9$

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

A nova árvore  $T_2$  é mostrada na figura 4.4, com a indicação dos novos multiplicadores.

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

- $B(4) = \{(2, 4)\};$
- $B(7) = \{(4, 7), (6, 7)\};$
- $B(6) = \{(2, 6), (4, 6)\};$
- $B(5) = \{(3, 5), (4, 5)\};$
- $B(2) = \{(1, 2), (6, 2)\}$
- $B(3) = \{(4, 3)\};$
- $B(1) = \{(0)\}.$



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  $\pi^{\wedge}(7) = 11$ . Todos os arcos dos conjuntos  $B(5)$ ,  $B(2)$  e  $B(3)$  atendem as condições de otimalidade, permanecendo na árvore os arcos atuais.

A nova árvore  $T_3$  é mostrada na figura 4.5.

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  $v$ .

A tabela abaixo apresenta uma análise da eficiência do método, em cada passo  $k$ , a partir do número de pivoteamentos permitidos, no caso do problema ter uma solução ótima.

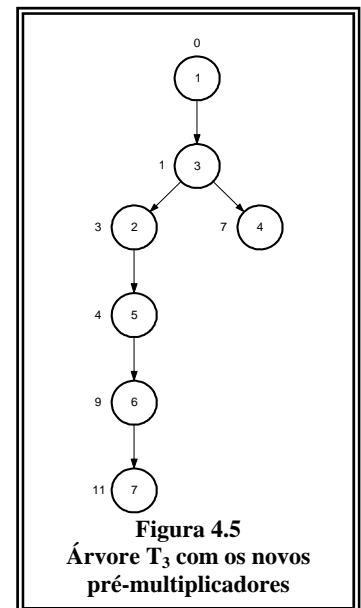
K	Árvore	Nós							Pivoteamento	
		1	2	3	4	5	6	7	NMP	NPR
0	$T_0$	0	1	1	1	1	1	1	6	-
1	$T_1$	0	3	1	3	2	2	1	5	4
2	$T_2$	0	3	1	3	2	5	5	4	2
3	$T_3$	0	3	1	3	2	5	6	3	1
4	$T^*$	0	3	1	3	2	5	6	2	0

NMP – número máximo de pivoteamentos permitidos em um passo  $k$ . NPR – número de pivoteamentos efetivamente realizados.

Observa-se que o número máximo de pivoteamentos 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 este exemplo.

## 5- CONCLUSÃO

Durante a pesquisa foi observado que 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. Foi assim que observamos que quando são utilizadas estruturas de dados mais elaboradas, melhora-se também 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  $\pi^{\leftarrow}(h) \leftarrow \pi^{\leftarrow}(h) + c_{gh} \pi^{\leftarrow}(h)$ ; *segundo*, pela garantia de que o método termina após realizar  $(n-1)(n-2)/2$  pivoteamentos, independente de encontrar ou não um ciclo negativo.

A simplicidade do pivoteamento vista pelo primal do problema, isto é, o problema analisado pelo fluxo no arco, decorre de cada nó cabeça demandar apenas 1 unidade de fluxo, isto é, qualquer arco escolhido para entrar na árvore conduz apenas uma unidade de fluxo, idêntica a quantidade do fluxo conduzido pelo arco escolhido para deixar a árvore. Esse fluxo é o mínimo entre todos os fluxos do *ciclo de pivoteamento*, em cada iteração.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ 1] AHUJA, R. K; MAGNANTI, T. L; ORLIN, J. B. (1993) - Network Flows, Theory, Algorithms and Applications. Prentice-Hall, Inc.
- [ 2] BELLMAN R. (1958) On a Route Problem. Quarterly Applied Mathematics 16 p 87-90.
- [ 3] CUNNINGHAM, W. (1976.)- A Network Simplex Method. Math. Programming 11 p105-116.
- [ 4] DIJKSTRA, E W. (1959) A Note on Two Problems in Connexion with Graphs. Numerische Mathematik Vol. 1. p 269-271.



- [ 5] FORD, L R. (1956) Network Flow Theory. The Rand corporation Report P-923, Santa Monica, CA.
- [ 6] GOLDFARB, D. AND JIN, Z. (1999) - An  $O(nm)$  Time Network Simplex Algorithm For the Shortest Path Problem. Operations Research, Vol.47, N<sup>o</sup> 3. p 445-448.
- [ 7] GOLDFARB, D; HAO, J. AND KAI, S.-R. (1990a) - Efficient Shortest Path Simplex Algorithms. Operations Research, vol. 38 No 4, p 624-628.
- [ 8] GOLDFARB, D; JIANXIU H; KAI S.-R. (1990b) - Anti-Stalling Pivot Rules For The Network Simplex Algorithm. Networks. John Wiley and Sons, Inc. Vol.20 p 79-91.
- [ 9] LAWLER, E L. (1956) - Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York.
- [10] MARINHO, C.E.V (2001) - 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.
- [11] MOORE, Z F. (1957) - The Shortest Path Through a Maze. Proc. Internat. Sympos. On Theory of Switching, Part II, p. 285-292.
- [12] ORLIN B. J. (1997) - 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.
- [13] TAHA, H. A. (1997) - Operations Research: An Introduction. Prentice Hall Inc, 6<sup>th</sup> Edition.