

Uma Prova Conceitual do Algoritmo out-of-Kilter para o Problema de Fluxos em Rede

Maycon da Silveira Paiva
Programa de Pós-Graduação em Matemática Aplicada - PPGMAP
Universidade Federal do Rio Grande do Sul - UFRGS
Rio Grande, Brasil
<https://orcid.org/0000-0003-0773-9433>

Catia Maria dos Santos Machado
Instituto de Matemática, Física e Estatística - IMEF
Universidade Federal do Rio Grande - FURG
Rio Grande, Brasil
<https://orcid.org/0000-0003-4055-6014>

Abstract — Apesar da Teoria de Grafos ser uma importante área da programação matemática com diversas aplicações, o desenvolvimento de aplicativos especializados ainda ocorre lentamente, sem conseguir acompanhar a demanda da comunidade acadêmica pelos mesmos. Visando colaborar com o desenvolvimento de softwares produzidos por cientistas, faz-se necessário realizar um estudo aprofundado dos fundamentos matemáticos que governam os algoritmos desenvolvidos. O presente trabalho tem como objetivo geral apresentar a fundamentação matemática da teoria e da técnica de solução do algoritmo out-of-Kilter, desenvolvido pelo matemático Delbert Fulkerson, para o Problema de Fluxos em Rede. Para tanto, são apresentados conceitos da topologia dos grafos e da Álgebra Linear necessários para o desenvolvimento do algoritmo. Além disso, o presente estudo busca contribuir com a realização de uma prova conceitual do algoritmo na resolução de problemas importantes como os de fluxos em redes.

Keywords — Fluxos em Rede, Out-Of-Kilter, Prova Conceitual, Grafos.

I. INTRODUÇÃO

Segundo [1], a teoria de grafos é uma importante área da programação matemática, com um papel de destaque em áreas de engenharia e na pesquisa operacional, fornecendo ferramentas para tratar problemas de redes. No entanto, de acordo com [2], existem dois modos de produção envolvidos no desenvolvimento de softwares científicos: o desenvolvimento comercial, realizado por empresas de software com programadores profissionais; e o desenvolvimento feito por cientistas, não necessariamente programadores, geralmente realizado de forma colaborativa.

[3] afirma que algumas disciplinas, como a de Teoria de Grafos, necessitam da utilização de ferramentas computacionais no decorrer de seu andamento e que apesar disso, carecem de massa crítica para estimular o desenvolvimento de pacotes comerciais para seus problemas específicos. Logo, faz-se necessário um estudo aprofundado sobre a fundamentação matemática envolvida nos algoritmos de solução de problemas

em rede conhecidos para que os mesmos possam vir a ser implementados.

Um exemplar deste tipo de algoritmo trata-se de um método alternativo conhecido como algoritmo *out-of-Kilter*, desenvolvido por Delbert Fulkerson em 1961. Ao contrário do algoritmo primal simplex em um grafo, o algoritmo *out-of-Kilter* não é a especialização de um método mais geral. O algoritmo foi desenvolvido especificamente para problemas de rede e é único na literatura de programação matemática [4].

Assim, o presente trabalho busca realizar um estudo sobre a fundamentação matemática da teoria e da técnica de solução para problemas de fluxo em rede através do algoritmo *out-of-Kilter*. Para tanto, este estudo busca apresentar as propriedades matemáticas da topologia dos grafos e os conceitos de Álgebra Linear necessários para o entendimento do desenvolvimento do algoritmo, bem como realizar uma introdução aos problemas de fluxos em rede e seus métodos de solução.

Por fim, a contribuição do trabalho é mostrar e incentivar o leitor, a partir do estudo da fundamentação teórica do algoritmo, sobre a importância e a necessidade de criar as próprias implementações computacionais. Pois, devido ao peso computacional das operações matriciais que, geralmente, são abordadas nas técnicas de solução implementadas por pacotes comerciais, resolver um problema de rede, poderá tornar-se inviável.

II. FUNDAMENTAÇÃO MATEMÁTICA

Nesta seção serão apresentados os fundamentos matemáticos que envolvem os problemas de fluxo em rede, tais fundamentos se alicerçam sobre a Teoria de Grafos e a Álgebra Linear. É válido ressaltar que as notações e convenções utilizadas são as mesmas adotadas por [4], mesmo quando são abordadas definições de outros autores.

Assim, considera-se um vetor linha (matriz $1 \times n$) quando pré-multiplica uma matriz e um vetor coluna (matriz $n \times 1$) quando pós-multiplica uma matriz. Considere a seguinte formulação para o problema de rede de custo mínimo:

$$\text{Min } cx \quad (1)$$

$$\text{s.a. } Ax = r \quad (2)$$

$$l \leq x \leq u \quad (3)$$

onde, A é a matriz de incidência nó-arco, x o vetor de fluxos associado aos arcos, r o vetor de ofertas e demandas associados aos nós da rede, c o vetor de custos, l e u representam, respectivamente, as capacidades mínimas e máximas de fluxo nos arcos da rede.

Segundo [5], um grafo dirigido trata-se de uma estrutura constituída por conjuntos de nós e arcos, representados respectivamente por N e R . Assim, pode-se observar na Fig. 1 o grafo $G = (N, R)$ formado pelos conjuntos $N = \{1, 2, 3, 4\}$ e $R = \{e_1, e_2, e_3, e_4, e_5\}$.

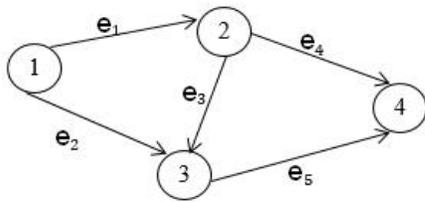


Figura 1 – Representação de uma rede através de um grafo G .

Ainda, o grafo $G = (N, R)$ pode ser representado por uma matriz onde as colunas representam os arcos (e_1, e_2, e_3, e_4, e_5) e as linhas representam os nós (1, 2, 3, 4), a qual chama-se matriz de incidência nó-arco. Os elementos de tal matriz assumem três valores distintos: 1 quando o arco sai do nó i ; -1 quando o arco chega ao nó i ; 0 nos demais casos. Logo, a matriz de incidência nó-arco que representa o grafo da Fig. 1 é dada por:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix} \quad (4)$$

De acordo com [6], um percurso como uma família de ligações sucessivamente adjacentes, ou seja, cada ligação tem uma extremidade adjacente à extremidade da ligação anterior enquanto sua outra extremidade é adjacente à posterior. Ainda de acordo com os autores, um percurso diz-se fechado quando a última ligação da sucessão for adjacente a primeira, além disso, diz simples quando não repetir ligações.

Assim, [6] define um ciclo como sendo um percurso simples e fechado além de definir um caminho como um percurso que segue a mesma orientação a partir do vértice inicial. A Fig. 2 (a) representa um ciclo $C = \{1, e_1, 2, e_4, 4, e_5, 3, e_2, 1\}$ enquanto a Fig. 2 (b) um caminho $P = \{1, e_1, 2, e_3, 3, e_5, 4\}$ para o grafo G apresentado anteriormente.

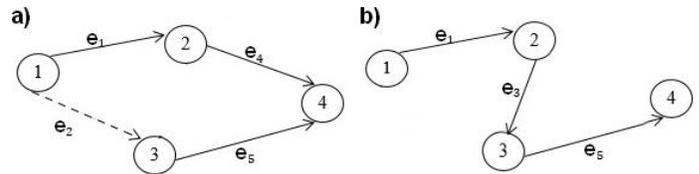


Figura 2 – (a) ilustração do ciclo C ; (b) ilustração do caminho P .

O comprimento de um caminho ou ciclo é o número de arcos no caminho ou ciclo. Para cada caminho ou ciclo, com comprimento n , a orientação da sequência $O(P)$ de n elementos é definida como segue:

$$O(P) = \begin{cases} +1, & \text{se } e_{ji} = (n_i, n_{i+1}) \\ -1, & \text{se } e_{ji} = (n_{i+1}, n_i) \end{cases} \quad (5)$$

Logo, é possível observar que para o caminho $P = \{1, e_1, 2, e_3, 3, e_5, 4\}$ ilustrado na Fig. 2 (b) a orientação da sequência é $\{+1, +1, +1\}$.

Segundo [4] uma árvore se trata de um grafo conexo e acíclico, ou seja, todo par de nós do conjunto N pode ser conectado através de um caminho, além disso, para cada par de nós do conjunto N existe apenas um caminho que os conecta. Ainda segundo os mesmos, uma árvore com raiz refere-se a uma árvore contendo um arco conectado apenas a um nó. Além disso, uma árvore τ é um subgrafo gerador de G , também chamada de árvore geradora para G .

A seguir são enunciadas algumas proposições demonstradas por [4] que servirão como suporte para os resultados encontrados posteriormente.

Proposição 1: se uma sequência finita $P = \{n_1, e_{j_1}, n_2, e_{j_2}, \dots, n_n, e_{j_n}, n_{n+1}\}$ é um caminho ou ciclo em um grafo próprio G com matriz de incidência A , então:

$$\sum_{i=1}^n O_i(P) A(j_i) = e^{n_1} - e^{n_{n+1}} \quad (6)$$

onde e é o vetor canônico com entrada +1 correspondente aos vértices 1 e n_{n+1} .

Proposição 2: se $C = \{n_1, e_{j_1}, n_2, e_{j_2}, \dots, n_n, e_{j_n}, n_{n+1}\}$ é ciclo de um grafo próprio G com matriz de incidência A , então:

$$\sum_{i=1}^n O_i(C) A(j_i) = 0 \quad (7)$$

Como consequência direta da proposição 2, temos que um ciclo $C = \{n_1, e_{j_1}, n_2, e_{j_2}, \dots, n_n, e_{j_n}, n_{n+1}\}$ de um grafo próprio G , com matriz de incidência A , tem seu conjunto de arcos $\{A(j_i): i = 1, \dots, n\}$ linearmente dependente.

Proposição 3: seja A uma matriz de incidência nó-arco para um grafo próprio G . Seja $\tau = [N, R]$ um subgrafo de G , isto é, uma

árvore contendo pelo menos dois nós. Então $\{A(j): e_j \in R\}$ é linearmente independente.

Proposição 4: seja A uma matriz de incidência nó-arco para um grafo próprio G conexo e com n nós. Então o posto da matriz A é $n - 1$.

Proposição 5: seja A uma matriz de incidência nó-arco para um grafo próprio $G = [N, R]$ onde G possui n nós. Seja \widehat{R} um subconjunto de R tal que $\{A(j): e_j \in \widehat{R}\}$ é linearmente independente com \widehat{R} tendo $n - 1$ arcos. Então, $\tau = [N, \widehat{R}]$ é uma árvore.

a. *Caracterização de uma Matriz Básica para o Problema de Fluxos em Redes*

Pela Proposição 4, a matriz de restrição para o problema não possui o posto completo. Então, a fim de completar o posto para a matriz de restrição, o problema (1) - (3) passa ter a seguinte formulação:

$$\text{Min } cx \tag{8}$$

$$\text{s.a. } Ax + ae^l = r \tag{9}$$

$$l \leq x \leq u \tag{10}$$

$$0 \leq a \leq 0 \tag{11}$$

onde, l é um inteiro positivo não maior que o número de nós n com a estritamente igual a zero. A solução ótima para (1) também será solução ótima para (8).

Proposição 6: seja A uma matriz de incidência nó-arco para um grafo próprio conexo $G = (N, R)$ com n nós. Seja $\tau = (N, \widehat{R})$ uma árvore geradora para G . Então, $B = \{A(j): e_j \in \widehat{R}\} \cup \{e^l\}$ gera E^n , i.e, existe um conjunto de colunas n a partir de $[A|e^l]$ que gera E^n .

Proposição 7: seja A uma matriz de incidência nó-arco para um grafo próprio conexo $G = (N, R)$ com nó raiz l . Se B é uma base para $[A|e^l]$, então $e^l \in B$ e $\tau = (N, \widehat{R})$ é uma árvore geradora para G , onde $\widehat{R} = \{e_j: A(j) \in B\}$.

Considere as seguintes árvores representadas na Fig. 3. Apesar dos conjuntos de vetores representados em ambas serem linearmente independente, na Fig. 3 (a) conjunto não gera o espaço vetorial E^n , enquanto na Fig. 3 (b), com a adição do arco raiz à árvore, passam a formar uma base para o espaço E^n .

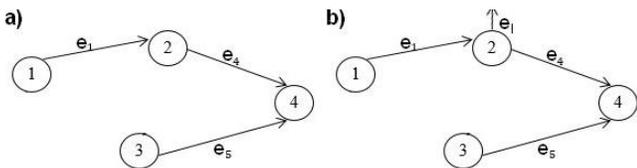


Figura 3 – (a) árvore τ de G ; (b) árvore básica τ_B com arco e nó raiz.

A partir das proposições acima, pode-se caracterizar uma base para a equação (9).

Proposição 8: seja A uma matriz de incidência nó-arco para um grafo próprio, conexo e enraizado com nó raiz l . Uma base para $[A|e^l]$ é um conjunto de arcos correspondentes a árvore geradora τ_B para G .

Proposição 9: seja A uma matriz de incidência nó-arco para um grafo próprio, conexo e enraizado com nó raiz l . Seja B uma base a partir de $[A|e^l]$. Logo, é sempre possível remodelar a matriz B na forma de uma matriz triangular.

O algoritmo 1, descrito a seguir opera diretamente na árvore τ_B associada a matriz B .

Algoritmo 1

P1. Inicialização:

Faça B ser uma base qualquer para o problema de fluxo em redes. Associe a essa base B a uma árvore geradora τ_B com nó raiz l . Seja n o número de nós da árvore τ_B . Faça $i \leftarrow 1$.

P2. Encontre uma folha da árvore que não seja um nó raiz l .

Seja $r \neq l$, uma folha qualquer de τ_B . Faça e_s ser o arco da árvore τ_B incidente em r .

P3. Insira i -ésima linha e coluna.

Faça a i -ésima linha de B corresponder ao nó r e faça a i -ésima coluna de B corresponder ao arco e_s .

P4. Reduza a árvore.

Se $i = n - 1$, vá para o passo 5; senão $\tau_B \leftarrow [N - \{r\}, R - \{e_s\}]$, $i \leftarrow i + 1$, e vá para o passo 2.

P5. Insira o nó e arco raiz.

Faça a n -ésima linha de B corresponder ao nó l , e faça a n -ésima coluna de B ser o arco raiz e^l .

Considere a matriz básica B , não triangular associada a τ_B ilustrada na Fig. 3. Através da aplicação do algoritmo 1 descrito anteriormente, é possível transformar a matriz B na matriz básica triangular B' .

$$B = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & -1 \end{bmatrix} \tag{12}$$

$$B' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} \tag{13}$$

b. Algoritmo Especializado para o Problema de Fluxo em Redes

O intuito de um algoritmo de resolução para o problema de fluxo de custo mínimo é melhorar a cada iteração o valor da função objetivo, minimizando-a ou maximizando-a, enquanto respeita as restrições impostas ao problema. Partindo de uma solução básica viável conhecida, o algoritmo alterna entre soluções básicas buscando encontrar uma solução básica ótima viável [7, 8].

Particionando as matrizes A , c , x , l e u em variáveis básicas e não básicas tem-se: $A = [B | N]$, $c = [c^B | c^N]$, $x = [x^B | x^N]$, $l = [l^B | l^N]$ e $u = [u^B | u^N]$ então:

$$\text{Min } c^B x^B + c^N x^N \quad (14)$$

$$\text{s.a } Bx^B + Nx^N = r \quad (15)$$

$$l^B \leq x^B \leq u^B \quad (16)$$

$$l^N \leq x^N \leq u^N \quad (17)$$

Isolando x^B na equação (15), tem-se:

$$x^B = B^{-1}r - B^{-1}Nx^N \quad (18)$$

Substituindo (18) em (14), (15) e (16) obtém-se:

$$\text{Min } c^B B^{-1}r + (c^N - c^B B^{-1}N)x^N \quad (19)$$

$$\text{s.a } l^B \leq B^{-1}b - B^{-1}Nx^N \leq u^B \quad (20)$$

$$l^N \leq x^N \leq u^N \quad (21)$$

Em problemas de rede de custo mínimo é garantido pelas proposições anteriores que o cálculo matricial

$$(c^N - c^B B^{-1}N)x^N \quad (22)$$

pode ser efetuado diretamente no grafo que representa a rede, eliminando o peso computacional do cálculo da matriz B^{-1} que se faz necessário quando o algoritmo não é especializado. Para determinar (16) é necessário primeiramente obter

$$\pi = c^B B^{-1} \quad (23)$$

o que significa determinar os π'_i relativo aos nós da árvore τ_B .

No, entanto a equação (23) é calculada resolvendo o sistema linear:

$$\pi B = c^B \quad (24)$$

Assim, todos os π'_i são unicamente determinados pela Proposição 9. A matriz básica B pode sempre ser colocada na forma triangular. O sistema linear (24) é resolvido fazendo uso da árvore τ_B , com nó raiz l e resume-se a determinar:

$$\begin{cases} \pi_l = 0 \\ \pi_{F(j)} - \pi_{T(j)} = c_j \quad \forall e_j \in \tau_B. \end{cases} \quad (24-a)$$

Substituindo (23) em (22) tem-se:

$$c^N - \pi N \quad (25)$$

Equivalente a:

$$c_j - \pi_{F(j)} + \pi_{T(j)} \quad (26)$$

O cálculo (26) é então realizado para determinar se a função objetivo pode ser melhorada com a entrada de uma variável não básica x^N . Assim, são definidos dois conjuntos:

$\psi_1 = \{ e_j: x_j = l_j \text{ e } (-\pi_{F(j)} + \pi_{T(j)} + c_j) < 0 \}$, variáveis limitadas inferiormente.

$\psi_2 = \{ e_j: x_j = u_j \text{ e } (-\pi_{F(j)} + \pi_{T(j)} + c_j) > 0 \}$, variáveis limitadas superiormente.

Se existir algum $x^N \in \psi_1 \cup \psi_2$, então a função objetivo pode ser melhorada. Somente quando $\psi_1 \cup \psi_2 = \emptyset$ a solução ótima é encontrada. Considerando $\psi_1 \cup \psi_2 \neq \emptyset$, uma variável não básica $x_k \in \psi_1 \cup \psi_2$ é selecionada a entrar na base.

O i -ésimo elemento do produto matricial $c^B B^{-1}N$ é dado por:

$$c^B B^{-1}N_{(i)} \quad (27)$$

Como a coluna de $N_{(i)} = A_{(k)}$ então

$$c^B B^{-1}A_{(k)} \quad (28)$$

A expressão (28) pode ser efetuada operando diretamente no grafo que representa a rede, eliminando o peso computacional do cálculo de B^{-1} . Para realizar o cálculo de (28) é necessário primeiro determinar:

$$y = B^{-1}A_{(k)} \quad (29)$$

O que significa determinar $O_i(P)$, a orientação dos arcos na árvore τ_B . No entanto, a equação (29) é calculada resolvendo o sistema linear

$$By = A_{(k)} = e^{F(k)} - e^{T(k)} \quad (30)$$

cuja solução pode ser obtida operando diretamente na árvore τ_B . Substituindo (29) em (30) tem-se:

$$c^B y \quad (31)$$

equivalente a:

$$\sum_{i=1}^n c_{j_i} O_i(P) \quad (32)$$

Pela Proposição 1, a equação (4) indica que o teste de razão pode ser obtido operando no grafo. Se os arcos da árvore τ_B são ordenados como $e_{k1}, e_{k2}, \dots, e_{kl}$, correspondendo as colunas \bar{l} de B . Então, as componentes de y podem ser determinadas pela orientação da sequência

$$\begin{cases} O_i(P) & \text{se } e_{kn} = e_{ji} \in P \\ 0, & \text{caso contrário} \end{cases} \quad (33)$$

E o cálculo de Δ_1 e Δ_2 no teste de razão é especializado como

$$\Delta_1 \leftarrow \min_{O_i(P)=\delta} \{x_{ji} - l_{ji}, \infty\} \quad (34)$$

$$\Delta_2 \leftarrow \min_{-O_i(P)=\delta} \{u_{ji} - x_{ji}, \infty\} \quad (35)$$

$$\Delta \leftarrow \min \{\Delta_1, \Delta_2, u_k - l_k\} \quad (36)$$

O teste de razão determina o valor Δ que será adicionado ou subtraído, ao fluxo do arco que entra, que encontra-se em sua capacidade mínima ou máxima. O valor Δ então é repassado para todos os arcos básicos que formam ciclo com o arco que entra. Assim, o algoritmo para o problema de rede de custo mínimo pode ser desenhado:

Algoritmo 2

Conforme mencionado anteriormente, o algoritmo *out-of-Kilter* foi desenvolvido por Delbert Fulkerson, em 1961, especificamente para encontrar soluções ótimas para problemas de fluxo em rede. Mais detalhes sobre condições de otimalidade para problemas de fluxo em rede podem ser encontrados em [4].

Considerando:

$$\Delta_{z(j)} = \pi_{F(j)} - \pi_{T(j)} - c_j \quad (37)$$

Temos que as condições de otimalidade para o problema de fluxos em rede podem ser escritas como:

$$Ax = b \quad (38)$$

$$\forall e_j \in E \begin{cases} \Delta_{z(j)} < 0 & \text{quando } x_j = l \\ \Delta_{z(j)} = 0 & \text{quando } l \leq x_j \leq u_j \\ \Delta_{z(j)} > 0 & \text{quando } x_j = u_j \end{cases} \quad (39)$$

Assim, é possível classificar os arcos de acordo com o quadro abaixo:

	$\Delta_{z(j)} < 0$	$\Delta_{z(j)} = 0$	$\Delta_{z(j)} > 0$
$x_j = u_j$	out-of-Kilter	in-Kilter	in-Kilter
$l < x_j < u_j$	out-of-Kilter	in-Kilter	out-of-Kilter
$x_j = l$	in-Kilter	in-Kilter	out-of-Kilter

Este método parte de uma solução básica viável conhecida, contendo tanto arcos *in-Kilter* quanto arcos *out-of-Kilter*. A cada iteração o algoritmo melhora o resultado da função objetivo até encontrar uma solução básica ótima viável, ou seja, quando todos arcos estiverem *in-Kilter*.

Algoritmo 2.1 – Out-of-Kilter

P0. Seja x uma solução tal que $Ax = b, 0 \leq x \leq u_j$. Seja π qualquer vetor de variáveis duais.

P1. Encontre um arco *out-of-Kilter*. Seja e_s um arco *out-of-Kilter*. Se todos arcos estiverem *in-Kilter* pare, a solução é ótima.

P2. Fase primal.

Execute o Algoritmo 2.2 com o arco e_s . Se o Algoritmo 2.2 termina com a conclusão de que não existe ciclo, então vá para P3; em caso contrário volte ao P1.

P3. Fase dual.

Execute o Algoritmo 2.3 com a árvore obtida no P2. Se o arco e_s está *out-of-Kilter*, então volte ao P2; em caso contrário volte ao P1.

Algoritmo 2.2 – Fase Primal

P0. Inicialização.

Se $\Delta_{z(s)} < 0$ então $\tilde{N} \leftarrow \{T(s)\}$ e $\Delta_{T(s)} \leftarrow x_s$; caso contrário $\tilde{N} \leftarrow \{F(s)\}$ e $\Delta_{F(s)} \leftarrow u_s - x_s$.

Faça $\hat{\mathcal{R}} \leftarrow \emptyset$.

P1. Determinação dos candidatos para a árvore τ .

Sejam:

$$\psi_1 = \{e_j : e_j \neq e_s, \Delta_{z(j)} \geq 0, x_j < u_j, F(j) \notin \tilde{N} \text{ e } T(j) \in \tilde{N}\}$$

$$\psi_2 = \{e_j : e_j \neq e_s, \Delta_{z(j)} \leq 0, x_j > l, F(j) \in \tilde{N} \text{ e } T(j) \notin \tilde{N}\}$$

Se $\psi_1 \cup \psi_2 = \emptyset$, então termine a fase primal com a conclusão de que não existe ciclo.

P2. Adição de novo arco na árvore τ .

Selecione o arco $e_k \in \psi_1 \cup \psi_2$.

Se $e_k \in \psi_1$, então $\Delta_{F(k)} \leftarrow \min \{\Delta_{T(k)}, u_k - x_k\}$.

Se $e_k \in \psi_2$, então $\Delta_{T(k)} \leftarrow \min \{\Delta_{F(k)}, x_k\}$.

Faça $\tilde{N} \leftarrow \tilde{N} \cup \{F(k), T(k)\}$ e $\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} \cup \{e_k\}$.

Se $\{F(s), T(s)\} \subset \tilde{N}$, então vá para P3;

Se $\{F(s), T(s)\} \notin \tilde{N}$, então volte ao P1.

P3. Atualização do fluxo no ciclo.

Se $\Delta_{z(s)} < 0$, então induzir o fluxo no ciclo por $\Delta_{F(s)}$, caso contrário induzir por $\Delta_{T(s)}$.

Algoritmo 2.3 – Fase Dual

P1. Determinar os arcos incidentes na árvore τ .

Sejam:

$$\psi_3 = \{e_j: T(j) \in \tilde{N}, F(j) \notin \tilde{N} \text{ e } \Delta_{z(j)} < 0\}$$

$$\psi_4 = \{e_j: T(j) \notin \tilde{N}, F(j) \in \tilde{N} \text{ e } \Delta_{z(j)} > 0\}$$

P2. Determinar a maior mudança possível.

$$\text{Faça } \theta \leftarrow \min e_j \in \psi_3 \cup \psi_4 [|\Delta_{z(j)}|].$$

P3. Faça $\pi_i \leftarrow \pi_i - \theta, \forall i \in \tilde{N}$.

III. UMA PROVA CONCEITURAL DA TEORIA MATEMÁTICA DO ALGORITMO ESPECIALIZADO PARA REDES

Conforme mencionado anteriormente, o problema de fluxos em rede apresenta diversas aplicações em diferentes áreas do conhecimento. Um exemplo prático consiste em minimizar o custo total no transporte de produtos de um ponto (nó) a outro através de estradas (arcos). Neste caso, pode-se considerar c_j como o custo por unidade de produto transportado por cada arco j e π_i o preço de transportar o produto pelo arco j até o nó i .

A Fig. 4 representa a rede onde 9 unidades de um produto devem ser deslocadas do nó de origem 2 até os nós de destino 1 e 3, onde serão alocadas respectivamente 1 e 8 unidades deste produto. A Tab. 1 informa os custos por unidade de produto associados a cada arco presente na rede.

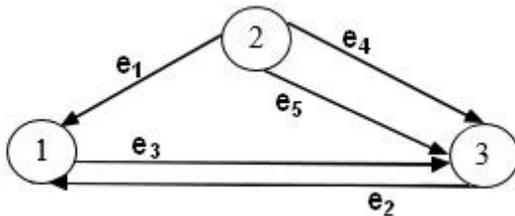


Figura 4 – Rede de transporte de um produto.

Tabela 1 – Custos de Transporte nos arcos.

Arco	Custo (c_j)
e_1	0
e_2	100
e_3	1
e_4	2
e_5	100

O problema enunciado pode ser formulado matematicamente da seguinte maneira:

$$\text{Min } Z = 0x_1 + 100x_2 + x_3 + 2x_4 + 100x_5 + 0x_6$$

$$\text{s.a } \begin{cases} -x_1 - x_2 + x_3 = -1 \\ x_1 + x_4 + x_5 = 9 \\ x_2 - x_3 - x_4 - x_5 = -8 \end{cases}$$

$$0 \leq x_1 \leq 8; 0 \leq x_2 \leq 5; 0 \leq x_3 \leq 9; 0 \leq x_4 \leq 10; 0 \leq x_5 \leq 10.$$

Uma solução básica inicial para o problema é $x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 9, x_5 = 0$ cujo valor Z inicial é 118 u.m. (unidade monetária). Além disso, nos nós tem-se $\pi = (0, 0, 0)$. Aplicando o Algoritmo 2:

1ª iteração:

No primeiro passo trata-se de verificar quais arcos estão *out-of-Kilter*. Na Tab. 2 é possível observar quais arcos satisfazem tal condição.

Tabela 2 – Análise da condição dos arcos na 1ª iteração.

$\Delta_{z(j)}$	x (solução atual)	Condição
$\Delta_{z(1)} = 0$	$x_1 = 0$	<i>in-Kilter</i>
$\Delta_{z(2)} = -100$	$x_2 = 1$	<i>out-of-Kilter</i>
$\Delta_{z(3)} = -1$	$x_3 = 0$	<i>in-Kilter</i>
$\Delta_{z(4)} = -2$	$x_4 = 9$	<i>out-of-Kilter</i>
$\Delta_{z(5)} = -100$	$x_5 = 0$	<i>in-Kilter</i>

Seja $e_s = e_2$. Assim:

Fase Primal:

P0. Inicialização:

Como $\Delta_{z(2)} < 0$, então $\tilde{N} \leftarrow \{1\}$ e $\Delta_1 \leftarrow 1$.

$\hat{\mathcal{R}} \leftarrow \emptyset$.

P1. Determinação dos candidatos para a árvore τ :

Nesta etapa apenas e_1, e_3 são candidatos pois e_4, e_5 não se relacionam com o nó 1. Assim:

$e_1: \Delta_{z(1)} = 0; x_1 = 0; F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \in \tilde{N}$. Logo $e_1 \in \psi_1$.

$e_3: \Delta_{z(3)} = -1; x_3 = 0$. Logo $e_3 \notin \psi_1$ e $e_3 \notin \psi_2$.

Em futuras etapas quando um arco $e_k \notin \psi_1$ e $e_k \notin \psi_2$, será dito apenas que o arco e_k falha como candidato.

P2. Adição de novo arco na árvore τ .

Como $e_1 \in \psi_1$, então $\Delta_2 \leftarrow \min \{\Delta_1, u_1 - x_1\}$;

$\Delta_2 \leftarrow \min \{1, 8\}$, $\Delta_2 \leftarrow 1$.

$\tilde{N} \leftarrow \tilde{N} \cup \{F_{(k)}, T_{(k)}\}$;

$\tilde{N} \leftarrow \{1\} \cup \{1, 2\}$, $\tilde{N} = \{1, 2\}$.

$\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} \cup \{e_k\}$;

$\hat{\mathcal{R}} \leftarrow \emptyset \cup \{e_1\}$, $\hat{\mathcal{R}} = \{e_1\}$.

$\{F_{(s)}, T_{(s)}\} = \{3, 1\} \not\subseteq \tilde{N}$, volte ao P1.

P1. Determinação dos candidatos para a árvore τ :

$e_1: \Delta_{z(1)} = 0; x_1 = 0; F_{(1)} = 2 \in \tilde{N}$. Falha;

$e_3: \Delta_{z(3)} = -1; x_3 = 0$. Falha;

$e_4: \Delta_{z(4)} = -2; x_4 = 9; F_{(4)} = 2 \in \tilde{N}$ e $T_{(4)} = 3 \notin \tilde{N}$. Logo $e_4 \in \psi_2$;

$e_5: \Delta_{z(5)} = -100; x_5 = 0$. Falha.

P2. Adição de novo arco na árvore τ .

Como $e_4 \in \psi_2$ então $\Delta_3 \leftarrow \min \{1, 9\}$, $\Delta_3 = 1$.

$\tilde{N} \leftarrow \{1, 2\} \cup \{2, 3\}$, $\tilde{N} = \{1, 2, 3\}$.

$\hat{\mathcal{R}} \leftarrow \{e_1\} \cup \{e_4\}$, $\hat{\mathcal{R}} = \{e_1, e_4\}$.

$\{F_{(s)}, T_{(s)}\} = \{3, 1\} \subset \tilde{N}$, então vá para P3.

P3. Atualização do fluxo no ciclo.

Como $\Delta_{z(2)} < 0$, induzir fluxo por $\Delta_3 = 1$.

Esta etapa consiste em atualizar o fluxo no ciclo $C_l = \{3, e_4, 2, e_1, 1, e_2, 3\}$. Destaca-se que o ciclo tem origem no nó 3, assim orientação da sequência é $\{-1, +1, -1\}$, conforme ilustra a Fig. 5.

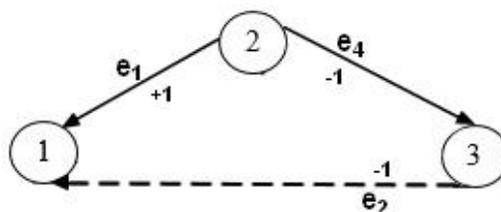


Figura 5 – Ilustração do ciclo C_1 .

Para a atualização do fluxo (solução) é preciso determinar os componentes de y conforme a Equação (33). Assim, têm-se $y = [1 -1 0 -1 0]$. Portanto:

$$x = x + \Delta_3 * y;$$

$$x = [0 1 0 9 0] + (1) * [1 -1 0 -1 0];$$

$$x = [1 0 0 8 0].$$

2ª iteração:

Tabela 3 – Análise da condição dos arcos na 2ª iteração.

$\Delta_{z(j)}$	x (solução atual)	Condição
$\Delta_{z(1)} = 0$	$x_1 = 1$	<i>in-Kilter</i>
$\Delta_{z(2)} = -100$	$x_2 = 0$	<i>in-Kilter</i>
$\Delta_{z(3)} = -1$	$x_3 = 0$	<i>in-Kilter</i>
$\Delta_{z(4)} = -2$	$x_4 = 8$	<i>out-of-Kilter</i>
$\Delta_{z(5)} = -100$	$x_5 = 0$	<i>in-Kilter</i>

Seja $e_s = e_4$. Assim:

Fase Primal:

P0. Inicialização:

Como $\Delta_{z(4)} < 0$, então $\tilde{N} \leftarrow \{3\}$ e $\Delta_3 \leftarrow 8$.

$\hat{\mathcal{R}} \leftarrow \emptyset$.

P1. Determinação dos candidatos para a árvore τ :

$e_1: \Delta_{z(1)} = 0; x_1 = 1; F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \notin \tilde{N}$. Falha;

$e_2: \Delta_{z(2)} = -100; x_2 = 0$. Falha;

$e_3: \Delta_{z(3)} = -1; x_3 = 0$. Falha;

$e_5: \Delta_{z(5)} = -100; x_5 = 0$. Falha.

Não há ciclos.

Fase Dual:

P1. Determinar os arcos incidentes na árvore τ .

$e_1: \Delta_{z(1)} = 0$. Falha;

$e_2: \Delta_{z(2)} = -100; T_{(2)} = 1 \notin \tilde{N}$. Falha;

$e_3: \Delta_{z(3)} = -1; T_{(3)} = 3 \in \tilde{N}$ e $F_{(3)} = 1 \notin \tilde{N}$. Logo $e_3 \in \psi_3$;

$e_4: \Delta_{z(4)} = -2; T_{(4)} = 3 \in \tilde{N}$ e $F_{(4)} = 2 \notin \tilde{N}$. Logo $e_4 \in \psi_3$;

$e_5: \Delta_{z(5)} = -100; T_{(5)} = 3 \in \tilde{N}$ e $F_{(5)} = 2 \notin \tilde{N}$. Logo $e_5 \in \psi_3$.

$\psi_3 = \{e_3, e_4, e_5\}$.

P2. Determinar a maior mudança possível.

$\theta \leftarrow \min \{|\Delta_{z(3)}|, |\Delta_{z(4)}|, |\Delta_{z(5)}|\}$

$\theta \leftarrow \min \{1, 2, 100\}, \theta = 1$.

P3. Atualização dos Preços em \tilde{N} .

$\tilde{N} = \{3\}$,

$\pi_i \leftarrow \pi_i - \theta, \forall i \in \tilde{N}$.

$\pi_3 = 0 - 1, \pi_3 = -1$.

$\pi = (0, 0, -1)$.

Finalizada a fase dual é possível refazer a fase primal visando obter novos resultados a respeito da condição dos arcos, pois devido aos novos valores de π , os custos $\Delta_{z(j)}$ serão alterados.

3ª iteração:

Tabela 4 – Análise da condição dos arcos na 3ª iteração.

$\Delta_{z(j)}$	x (solução atual)	Condição
$\Delta_{z(1)} = 0$	$x_1 = 1$	<i>in-Kilter</i>
$\Delta_{z(2)} = -101$	$x_2 = 0$	<i>in-Kilter</i>
$\Delta_{z(3)} = 0$	$x_3 = 0$	<i>in-Kilter</i>
$\Delta_{z(4)} = -1$	$x_4 = 8$	<i>out-of-Kilter</i>
$\Delta_{z(5)} = -99$	$x_5 = 0$	<i>in-Kilter</i>

Seja $e_5 = e_4$. Assim:

Fase Primal:

P0. Inicialização:

Como $\Delta_{z(4)} < 0$, então $\tilde{N} \leftarrow \{3\}$ e $\Delta_3 \leftarrow 8$.

$\hat{\mathcal{R}} \leftarrow \emptyset$.

P1. Determinação dos candidatos para a árvore τ :

$e_1: \Delta_{z(1)} = 0; x_1 = 1; F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \notin \tilde{N}$. Falha;

$e_2: \Delta_{z(2)} = -101; x_2 = 0$. Falha;

$e_3: \Delta_{z(3)} = 0; x_3 = 0; F_{(3)} = 1 \notin \tilde{N}$ e $T_{(3)} = 3 \in \tilde{N}$. Logo $e_3 \in \psi_1$;

$e_5: \Delta_{z(5)} = -99; x_5 = 0$. Falha.

P2. Adição de novo arco na árvore τ .

Como $e_3 \in \psi_1$ então $\Delta_1 \leftarrow \min \{8, 9\}, \Delta_1 = 8$.

$\tilde{N} \leftarrow \{3\} \cup \{1, 3\}, \tilde{N} = \{1, 3\}$.

$\hat{\mathcal{R}} \leftarrow \emptyset \cup \{e_3\}, \hat{\mathcal{R}} = \{e_3\}$.

$\{F_{(4)}, T_{(4)}\} = \{2, 3\} \not\subseteq \tilde{N}$ volte ao P1.

P1. Determinação dos candidatos para a árvore τ :

$e_1: \Delta_{z(1)} = 0; x_1 = 1; F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \in \tilde{N}$. Logo $e_1 \in \psi_1$;

$e_2: \Delta_{z(2)} = -101; x_2 = 0$. Falha;

$e_3: \Delta_{z(3)} = 0; x_3 = 0; F_{(3)} = 1 \in \tilde{N}$ e $T_{(3)} = 3 \in \tilde{N}$. Falha;

$e_5: \Delta_{z(5)} = -99; x_5 = 0$. Falha.

P2. Adição de novo arco na árvore τ .

Como $e_1 \in \psi_1$ então $\Delta_2 \leftarrow \min \{8, 7\}, \Delta_2 = 7$.

$\tilde{N} \leftarrow \{1, 3\} \cup \{2, 1\}, \tilde{N} = \{1, 2, 3\}$.

$\hat{\mathcal{R}} \leftarrow \{e_3\} \cup \{e_1\}, \hat{\mathcal{R}} = \{e_3, e_1\}$.

$\{F_{(4)}, T_{(4)}\} = \{2, 3\} \subset \tilde{N}$, então vá para P3.

P3. Atualização do fluxo no ciclo.

Como $\Delta_{z(4)} < 0$, induzir fluxo por $\Delta_2 = 7$.

Esta etapa consiste em atualizar o fluxo no ciclo $C_2 = \{2, e_1, 1, e_3, 3, e_4, 2\}$, cuja orientação é $\{+1, +1, -1\}$, conforme ilustra a Fig. 6. Assim, temos que $y = [1 \ 0 \ 1 \ -1 \ 0]$.

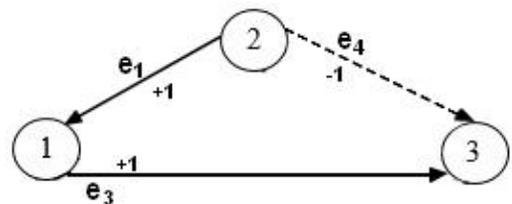


Figura 6 – Ilustração do ciclo C_2 .

Logo, a nova solução é:

$$x = [1 \ 0 \ 0 \ 8 \ 0] + (7) * [1 \ 0 \ 1 \ -1 \ 0],$$

$$x = [8 \ 0 \ 7 \ 1 \ 0].$$

4ª iteração:

Tabela 5 – Análise da condição dos arcos na 4ª iteração.

$\Delta_{z(j)}$	x (solução atual)	Condição
$\Delta_{z(1)} = 0$	$x_1 = 8$	<i>in-Kilter</i>
$\Delta_{z(2)} = -101$	$x_2 = 0$	<i>in-Kilter</i>
$\Delta_{z(3)} = 0$	$x_3 = 7$	<i>in-Kilter</i>
$\Delta_{z(4)} = -1$	$x_4 = 1$	<i>out-of-Kilter</i>
$\Delta_{z(5)} = -99$	$x_5 = 0$	<i>in-Kilter</i>

Seja $e_s = e_4$. Assim:

Fase Primal:

P0. Inicialização:

Como $\Delta_{z(4)} < 0$, então $\tilde{N} \leftarrow \{3\}$ e $\Delta_3 \leftarrow 1$.

$$\hat{\mathcal{R}} \leftarrow \emptyset.$$

P1. Determinação dos candidatos para a árvore τ :

e_1 : $\Delta_{z(1)} = 0$; $x_1 = 8$; $F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \notin \tilde{N}$. Falha;

e_2 : $\Delta_{z(2)} = -101$; $x_2 = 0$. Falha;

e_3 : $\Delta_{z(3)} = 0$; $x_3 = 7$; $F_{(3)} = 1 \notin \tilde{N}$ e $T_{(3)} = 3 \in \tilde{N}$. Logo $e_3 \in \psi_1$;

e_5 : $\Delta_{z(5)} = -99$; $x_5 = 0$. Falha.

P2. Adição de novo arco na árvore τ .

Como $e_3 \in \psi_1$ então $\Delta_1 \leftarrow \min \{1, 2\}$, $\Delta_1 = 1$.

$$\tilde{N} \leftarrow \{3\} \cup \{1, 3\}, \tilde{N} = \{1, 3\}.$$

$$\hat{\mathcal{R}} \leftarrow \emptyset \cup \{e_3\}, \hat{\mathcal{R}} = \{e_3\}.$$

$\{F_{(4)}, T_{(4)}\} = \{2, 3\} \notin \tilde{N}$ volte ao P1.

P1. Determinação dos candidatos para a árvore τ :

e_1 : $\Delta_{z(1)} = 0$; $x_1 = 8$; $F_{(1)} = 2 \notin \tilde{N}$ e $T_{(1)} = 1 \notin \tilde{N}$. Falha;

e_2 : $\Delta_{z(2)} = -101$; $x_2 = 0$. Falha;

e_3 : $\Delta_{z(3)} = 0$; $x_3 = 7$; $F_{(3)} = 1 \in \tilde{N}$ e $T_{(3)} = 3 \in \tilde{N}$. Falha;

e_5 : $\Delta_{z(5)} = -99$; $x_5 = 0$. Falha.

Não há ciclos.

Fase Dual:

P1. Determinar os arcos incidentes na árvore τ .

e_1 : $\Delta_{z(1)} = 0$. Falha;

e_2 : $\Delta_{z(2)} = -101$; $T_{(2)} = 1 \in \tilde{N}$ e $F_{(2)} = 3 \in \tilde{N}$. Falha;

e_3 : $\Delta_{z(3)} = 0$. Falha;

e_4 : $\Delta_{z(4)} = -1$; $T_{(4)} = 3 \in \tilde{N}$ e $F_{(4)} = 2 \notin \tilde{N}$. Logo $e_4 \in \psi_3$;

e_5 : $\Delta_{z(5)} = -99$; $T_{(5)} = 3 \in \tilde{N}$ e $F_{(5)} = 2 \notin \tilde{N}$. Logo $e_5 \in \psi_3$.

$$\psi_3 = \{e_4, e_5\}.$$

P2. Determinar a maior mudança possível.

$$\theta \leftarrow \min \{1, 99\}; \theta = 1.$$

P3. Atualização dos Preços em \tilde{N} .

$$\tilde{N} = \{1, 3\}.$$

$$\pi_1 = 0 - 1 = -1.$$

$$\pi_3 = -1 - 1 = -2.$$

$$\pi = (-1, 0, -2).$$

5ª iteração:

Tabela 6 – Análise da condição dos arcos na 5ª iteração.

$\Delta_{z(j)}$	x (solução atual)	Condição
$\Delta_{z(1)} = 1$	$x_1 = 8$	<i>in-Kilter</i>
$\Delta_{z(2)} = -101$	$x_2 = 0$	<i>in-Kilter</i>
$\Delta_{z(3)} = 0$	$x_3 = 7$	<i>in-Kilter</i>
$\Delta_{z(4)} = 0$	$x_4 = 1$	<i>in-Kilter</i>
$\Delta_{z(5)} = -98$	$x_5 = 0$	<i>in-Kilter</i>

Todos arcos encontram-se *in-kilter*, portanto o problema está em sua solução ótima $x = [8 \ 0 \ 7 \ 1 \ 0]$, com $Z = 9$ u.m..

A Fig. 7 ilustra a solução.

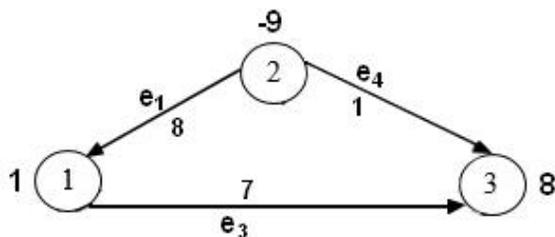


Figura 7 – Ilustração da solução ótima viável básica encontrada.

IV. CONSIDERAÇÕES FINAIS

O presente trabalho realizou uma prova conceitual, do algoritmo out-of-Kilter, desenvolvida especificamente para problemas de fluxos em rede. Tal prova deu-se através da utilização de resultados e propriedades matemáticas da Teoria de Grafos, Álgebra Linear, Topologia e Combinatória.

Visando exibir a estreita relação com os conteúdos matemáticos supracitados, bem como fazer a ponte entre a teoria e aplicação, estudou-se os conjuntos convexos, revisitou-se conceitos de ponto extremo, de fronteira e de interior. Possibilitando, assim, a compreensão do porquê existe uma solução exata para o problema de fluxo em redes. É importante também salientar que a fundamentação teórica do algoritmo para grafos está alicerçada sobre Espaços Vetoriais, trazendo importantes conceitos como base, dimensão, conjunto gerador, vetores linearmente independentes e dependentes.

Além disso, a relevância do trabalho desenvolvido consiste no estudo da fundamentação matemática dos algoritmos de fluxos em redes auxiliar pesquisadores no desenvolvimento de softwares específicos para grafos. Algoritmos desenvolvidos para problemas de fluxo em redes podem operar diretamente no grafo, sem a necessidade de operações matriciais. No caso geral, aplicativos são suportados por algoritmos desenvolvidos utilizando tais operações, onde a cada iteração faz-se necessário o cálculo da matriz inversa B^{-1} , seja para determinar as variáveis duais (valores de π) ou para realizar o teste da razão (orientação dos arcos na árvore τ_B).

Finalmente, pode-se dizer que o trabalho desenvolvido resultou em avanços sobre o entendimento da fundamentação matemática de um algoritmo de fluxo em redes. Espera-se que esse trabalho também seja um ponto de partida para aqueles que desejam implementar algoritmos especializados para problemas de rede.

Em trabalhos futuros, pretende-se desenvolver uma ferramenta de livre acesso a fim de incentivar aqueles que queiram estudar a Teoria dos Grafos aplicada a realidade empresarial e industrial, logística e transporte, investigação operativa, desenho de redes entre outros.

AGRADECIMENTOS

PAIVA, M. S. agradece à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro concedido através do Programa de Demanda Social.

REFERÊNCIAS

- [1] F. HERNANDES, “Algoritmos para Problemas de Grafos com Incertezas”. Tese de Doutorado – UNICAMP. Campinas. 2007.
- [2] F. C. COELHO, “Computação Científica com Python: Uma introdução à programação para cientistas”. Petrópolis: Edição do Autor, 2007.
- [3] C. A. VAZ, “Os Fundamentos Matemáticos do Algoritmo Especializado para Redes de Custo Mínimo Implementado na Ferramenta Computacional Grafos”. Trabalho de Conclusão de Curso – FURG. Rio Grande. 2017.
- [4] J. L. KENNINGTON, R. V. HELGASON, “Algorithm for Network Programming”. New York: John Willey & Sons, 1980.
- [5] C. M. dos S. MACHADO, D. F. ADAMATTI, L. R. MEINERZ, G. S. P. CARVALHO, “Um olhar sobre os Fundamentos Matemáticos da Geometria e Álgebra Linear na Solução de um Modelo de Programação Linear”. *XLII COBENGE*, Juiz de Fora, 2014.
- [6] P. O. BOAVENTURA NETTO, S. JURKIEWICZ, “*Grafos: introdução e prática*”. São Paulo: Blucher. 2017.
- [7] C. LOESCH, N. HEIN, “*Pesquisa Operacional*”. Blumenau: Editora da Furb, 1999.
- [8] A. A. LONGARAY, “*Introdução à Pesquisa Operacional*”. Rio Grande: Saraiva, 2013.