Um ponto de vista sobre programação linear

Luiz Fernando Ramos Lemos

Orientador: Lúcio Tunes dos Santos

11 de julho de 2011

Resumo

Este trabalho constitui uma forma alternativa de olhar para o problema de programação linear. É apresentado aqui uma abordagem algébrica distinta da usual utilizada no método Simplex. Trataremos do problema linear com vetor de custo não-negativo, o mesmo será abordado na forma canônica de minimização. São utilizados para tal, artifícios básicos da teoria de cones. Ao final é apresentado uma implementação e resultados de testes do NETLIB.

1 Proposição

Considere o seguinte problema linear de minimização

$$\begin{array}{rrcl} \min & c^T x & & \\ \text{s.a} & Ax & \geq & b \\ & x & \geq & 0, \end{array} \tag{1}$$

onde $A \in \mathbb{R}^{m\times n}$, $x, c \in \mathbb{R}^n$ e $b \in \mathbb{R}^m$. É evidente que este problema teria $x^* = 0$ como solução se ocorresse $b \leq 0$, e crescimento em qualquer direção positiva, isto é, $c \geq 0$.

Tomando isto como motivação, construiremos um método a partir do qual, através de sucessivas translações e mudanças de base, se obtém um problema equivalente que satisfaz as condições citadas. Ao contrário do método Simplex, não faremos hipóteses sobre o vetor b, no entanto, restringiremos a abordagem apenas para o caso em que $c \geq 0$.

2 Cones

Cones são estruturas bem definidas e muito utilizadas no campo da álgebra e otimização. A teoria aqui apresentada é fundamental para entendimento do algoritmo apresentado.

Definição 2.1. *Um* cone *é qualquer subconjunto de um espaço vetorial que seja fechado para a multiplicação por escalares positivos.*

2.1 Cones e restrições lineares

Cones podem ser definidos por restrições lineares, e, de fato, isto consta frequentemente na literatura de programação linear. Esta estrutura é chamada *cone convexo*. Analisaremos apenas o caso em que a matriz de coeficientes das restrições é invertível, e restrições de desigualdade.

Para os resultados a seguir tome que, dado um sistema $Bx \geq b$, subentende-se, B uma matriz quadrada de ordem n e invertível, e b um vetor dimensão n.

Teorema 2.1. *O conjunto $\mathcal{C} = \{x \in \mathbb{R}^m | Bx \geq b\}$ é um cone se, e somente se, b é nulo.*

Prova. $\Rightarrow$ Dado d tal que $Bd \geq 0$ então para $t \geq 0$ segue-se que $B[td] = t[Bd] \geq 0$. Logo $\mathcal{C}$ é um cone para $b = 0$.
$\Leftarrow$ Para $b \neq 0$, existe então d tal que $Bd = b$. E ainda, para todo t segue que $B[td] = t[Bd] = tb$. No entanto é sempre possível $t \geq 0$ tal que a relação $tb \geq b$ não seja satisfeita. ■

Definição 2.2. *Chamaremos um cone convexo de* cone invertível *quando sua matriz de coeficientes é invertível.*

Teorema 2.2. *Seja $\mathcal{C} = \{x \in \mathbb{R}^m | Bx \geq b, b \neq 0\}$. Existe, então, uma translação $x = x' + \overline{x}$ tal que o conjunto $\mathcal{C}' = \{x \in \mathbb{R}^m | x = x' + \overline{x}, Ax \geq b\}$ é um cone.*

Prova. Tome $\overline{x} = B^{-1}b$, logo $Bx = Bx' + B\overline{x} \geq b$. Logo, temos que $Bx \geq 0$. Logo, para esta translação $\mathcal{C}' = \{x' \in \mathbb{R}^m | Bx' \geq 0\}$, que é um cone. ■

Definição 2.3. *Como é natural de se pensar, chamaremos de* cone transladado *um conjunto que satisfaça as hipóteses do teorema acima.*

Definição 2.4. *Dado um cone linear invertível, transladado ou não, chamaremos* ponta do cone *o ponto $x = B^{-1}b$.*

Uma forma mais simples de analisar o comportamento de um cone linear invertível sujeito a restrições lineares é colocado na forma $x' \geq 0$. Para tal, sendo B a matriz de coeficientes deste cone, basta tomarmos a mudança de base $x = B^{-1}x'$.

Definição 2.5. *Chamaremos de* cone identidade *um cone na forma $x \geq 0$.*

2.2 Cones de crescimento

Nesta seção, definiremos a estrutura *cone de crescimento*, também chamado de *cone de direções de subida*, o qual formará a célula básica de funcionamento do algoritmo proposto.

Definição 2.6. *Dizemos o cone $\mathcal{C}$, transladado ou não, é* cone de crescimento *em relação a função objetiva $c^T x$, se para todo $d \in \mathcal{C}$, $c^T d \geq 0$.*

Teorema 2.3. *Um cone identidade é de crescimento em relação a função $c^T x$ se, e somente se, $c \geq 0$.*

Prova. $\Rightarrow$ Em um cone identidade, temos que sempre $d \geq 0$. Logo, se $c \geq 0$, $c^T d \geq 0$ para todo d pertencente ao cone.
$\Leftarrow$ Suponha que exista uma componente i de c tal que $c_i < 0$, tomando $d_i = \delta_{ij}$, temos que $c^T d < 0$, o que contraria a definição de cone de crescimento. ■

Teorema 2.4. *Um cone convexo $\mathcal{C} = \{x \in \mathbb{R}^m | Bx \geq b\}$, transladado ou não, é um cone de crescimento em relação a função objetiva $c^T x$ se, e somente se, $c^T B^{-1} \geq 0$.*

Prova. O resultado é evidente, basta tomar a mudança de base $x = B^{-1}x'$ e relacionar com o teorema anterior. ■

3 Vendo o problema através de cones

Trataremos a partir daqui sobre o problema linear de minimização na forma canônica, problema (12), com $c \geq 0$.

Com intuito de facilitar a compreensão deste, faremos uma breve interpretação dele através de cones de crescimento.

O problema é definido por sua função objetiva, não-negatividade, e restrições lineares. Em termos de cones, poderíamos pensar no par, função objetiva e não-negatividade como sendo um cone de crescimento identidade. Se o conjunto de restrições não inviabiliza a ponta do cone de não-negatividade, este ponto, $x^* = 0$, é ótimo. Isto nos sugere buscar um cone de crescimento com ponta factível definido por um subconjunto de restrições do problema.

Os procedimentos adiante visam obter tal cone por meio de transformações lineares do problema, tentando recair em um problema equivalente de solução trivial.

4 Factibilidade e Otimalidade

Considerando um problema linear, duas questões são levantadas, factibilidade e otimalidade. Veremos para certos casos como retirar estas informações da representação do problema.

4.1 Factibilidade

A factibilidade é uma questão difícil de se avaliar prontamente. Porém, em certas circunstâncias, a verificação é imediata. Analisaremos a seguir os casos imediatos.

Teorema 4.1. *Dado problema (12), assumindo* $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix}$, *se existir uma restrição* i *tal que* $b_i > 0$ *e* $a_i \leq 0$ *então o problema é infactível.*

Prova. A verificação deste teorema é imediata. Basta notar que $a_i^T x \leq 0 < b_i$ para todo $x \geq 0$. ∎

Definição 4.1. *Chamaremos esta ocorrência de infactibilidade explícita.*

Teorema 4.2. *O problema abordado é factível se* $b \leq 0$.

Prova. Para o ponto $x^* = 0$ teremos $Ax^* = 0 \geq b$. Além disso, o ponto também satisfaz $x \geq 0$, logo pelo menos este ponto é factível. ∎

4.2 Otimalidade

Dado um ponto, ele é ótimo se é factível e se todo ponto factível tiver valor objetivo maior ou igual ao seu. Trataremos apenas o caso em que o ponto ótimo é evidente.

Teorema 4.3. *Um problema linear de minimização na forma padrão, com o vetor de custos positivo, tem solução ótima* $x^* = 0$ *se este ponto for factível.*

Prova. Todo ponto factível do problema pertence ao cone de crescimento identidade definido pelo vetor de custo e a não-negatividade. Logo, se a origem é factível, todo ponto factível necessariamente tem valor objetivo maior, sendo então, $x^* = 0$ ótimo. ∎

5 Reescrevendo um problema linear através de cones

Um problema linear pode ser reescrito em outro equivalente, a fim de obtermos um que contenha claramente a informação sobre o ponto ótimo. Analisaremos, primeiramente, formas de reescrever um problema linear por meio de cones sem o intuito de buscar uma solução.

5.1 Translações

Definição 5.1. *Definimos por transladar um problema linear para um ponto $\overline{x}$ como fazer a mudança de variável $x = x' + \overline{x}$.*

Exemplificaremos, agora, este mecanismo para o problema (12). Faremos sua translação para o ponto $\overline{x}$, tomando $x = x' + \overline{x}$, temos

$$\begin{array}{rrcrcl} \min & c^T x & = & c^T x' + c^T \overline{x} & & \\ \text{s.a} & Ax & = & Ax' + A\overline{x} & \geq & b \\ & x & = & x' + \overline{x} & \geq & 0, \end{array} \tag{2}$$

ou seja,

$$\begin{array}{rrcl} \min & c^T x' + c^T \overline{x} & & \\ \text{s.a} & Ax' & \geq & b - A\overline{x} \\ & x' & \geq & -\overline{x}. \end{array} \tag{3}$$

Vemos que este problema não está na forma canônica de minimização. Mostraremos, mais tarde, como fazer operações que resultam em mesmo formato. No entanto, é evidente a relação de equivalência entre o problema inicial e o transladado, bastando tomar $x = x' + \overline{x}$ para relacionar seus valores.

Podemos simplificar a representação do problema transladado ainda mais. Temos que $c^T \overline{x}$ é constante e, portanto, não influi processo de minimização, podendo, então, ser eliminado. Assim, tomando $b' = b - A\overline{x}$,

$$\begin{array}{rrcl} \min & c^T x' & & \\ \text{s.a} & Ax' & \geq & b' \\ & x' & \geq & -\overline{x}. \end{array} \tag{4}$$

5.2 Mudança de base

A mudança de base é uma outra etapa a ser utilizada no algoritmo, com objetivo de chegarmos a um problema de mesmo formato do original. A mudança de base utilizada consiste, sendo x o vetor de variáveis, em reescreveremos o problema na forma $x = B^{-1}x'$, onde B representa os coeficientes das restrições escolhidas.

Particionaremos o conjunto de restrições escolhendo uma matriz invertível B, reordenando as linhas sem perda de generalidade obtemos

$$\begin{array}{rcl} Ax & \geq & b \\ x & \geq & 0 \end{array} \quad \Longleftrightarrow \quad \begin{array}{rcl} Nx & \geq & b_N \\ Bx & \geq & b_B, \end{array} \tag{5}$$

Aplicando a mudança de base $x = B^{-1}x'$, temos, por fim,

$$\begin{array}{rrcl} \min & c^T B^{-1}x' & & \\ \text{s.a} & NB^{-1}x' & \geq & b_N \\ & x' & \geq & b_B \end{array} \tag{6}$$

A solução do problema na base inicial e do problema transformado seguem a relação $x = B^{-1}x'$ para todo ponto. Ainda, a informação de mudanças de base sucessivas se acumulam no locus da não-negatividade inicial, de forma que sempre é possível voltar a base original.

5.3 Reescrevendo um problema linear mantendo a foma canônica de minimização

Agora, uniremos a translação e mudança de base, de forma a reescrever o problema em um equivalente de mesmo formato. Baseado na escolha de um conjunto de restrições, faremos uma translação e mudança de base que torna o conjunto de restrições selecionado uma não-negatividade.

Dado o problema (12), escolheremos um subconjunto das restrições tal que sua matriz de coeficientes B seja invertível, podendo incluir restrições de não-negatividade. Denotaremos as restrições não escolhidas por $Nx \geq b_N$ e o conjunto escolhido por $Bx \geq b_B$. Notamos que o conjunto de restrições escolhido define um cone transladado com ponta $\overline{x} = B^{-1}b_B$. Para facilitar a análise, exibiremos o problema representado de duas formas, diferentes apenas pela ordem das linhas.

$$\begin{array}{rrcl}
\min & c^T x & & \\
\text{s.a} & Ax & \geq & b \\
& x & \geq & 0
\end{array}
\iff
\begin{array}{rrcl}
\min & c^T x & & \\
\text{s.a} & Nx & \geq & b_N \\
& Bx & \geq & b_B.
\end{array} \tag{7}$$

Aplicaremos a translação, $x = x' + \overline{x}$, e teremos por resultado

$$\begin{array}{rrcc}
\min & c^T x' & + & c^T\overline{x} \\
\text{s.a} & Ax' & \geq & b - A\overline{x} \\
& x' & \geq & -\overline{x}
\end{array}
\iff
\begin{array}{rrcr}
\min & c^T x' & + & c^T\overline{x} \\
\text{s.a} & Nx' & \geq & b_N - N\overline{x} \\
& Bx' & \geq & 0.
\end{array} \tag{8}$$

Agora, realizando a mudança de base $x' = B^{-1}x''$, temos, por fim

$$\begin{array}{rrcc}
\min & c^T B^{-1}x'' & + & c^T\overline{x} \\
\text{s.a} & AB^{-1}x'' & \geq & b - A\overline{x} \\
& B^{-1}x'' & \geq & -\overline{x}
\end{array}
\iff
\begin{array}{rrcr}
\min & c^T B^{-1}x'' & + & c^T\overline{x} \\
\text{s.a} & NB^{-1}x'' & \geq & b_N - N\overline{x} \\
& x'' & \geq & 0
\end{array} \tag{9}$$

Veja que podemos ter a informação da base escolhida observando as restrições de não-negatividade iniciais. A equivalência entre valores dos problemas segue relação biunívoca $x = B^{-1}x'' + \overline{x}$. E ainda, este problema consta em seu formato original, canônico de minimização, a menos da constante $c^T\overline{x}$, que pode ser eliminada. Tomando $b' = b - A\overline{x}$ e $c'^T = c^T B^{-1}$, podemos reescrever o nosso resultado final da seguinte forma

$$\begin{array}{rrcl}
\min & c'^T x'' & & \\
\text{s.a} & AB^{-1}x'' & \geq & b' \\
& B^{-1}x'' & \geq & -\overline{x}.
\end{array} \tag{10}$$

Notamos que ao fazermos transformações sucessivas a mudança de base se acumulam no locus da não-negatividade inicial assim como as translações, estas, com sinal trocado mas com valores na base inicial.

5.4 Selecionando cones de crescimento transladados

Mostraremos aqui, especificamente, como selecionar cones de crescimento que tenham ponta não-negativa. Mais especificamente, abordaremos exclusivamente cones formados pelas restrições de não-negatividade, exceto uma, a qual será substituída por uma restrição qualquer do problema. Representaremos o cone transladado escolhido por $Bx \geq b_B$.

Trataremos do problema (12), com $c > 0$, denotando A por $\begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix}$.

Teorema 5.1. *Se existe $b_i > 0$, onde a_i possui pelo menos uma componente a_{ij} positiva, então, existe um cone de crescimento transladado formado por $a_i^T x \geq b_i$ e $n - 1$ restrições de não-negatividade.*

Prova. A ponta de um cone linear ocorre quando obtemos a saturação de todas suas restrições, para todo par b_i e a_{ij} que satisfaça a hipótese, o ponto $\overline{x} = (b_i/a_{ij})\delta_{kj}$ satura a restrição i e $n - 1$ não-negatividades. Portanto satura n restrições linearmente independentes, configurando a ponta de um cone transladado invertível. Basta, agora, demonstrar que uma destas escolhas define a ponta de um cone de crescimento.

Tendo escolhido uma restrição i onde $b_i > 0$, queremos que ocorra $c^T B^{-1} \geq 0$. Assim,

$$c^T B^{-1} = \left[(c_1 - a_{i1}\frac{c_j}{a_{ij}}) \ldots \frac{c_j}{a_{ij}} \ldots (c_n - a_{in}\frac{c_j}{a_{ij}}) \right] \geq 0. \tag{11}$$

Logo, queremos que $c_k - a_{ik}c_j/a_{ij} \geq 0$ para todo k, assim $c_k/a_{ik} \geq c_j/a_{ij}$. Escolhendo j tal que c_j/a_{ij} seja mínimo temos a ponta de um cone de crescimento. ∎

6 Método de cones de crescimento

Com embasamento na teoria proposta, construiremos um método para resolver o problema citado inicialmente.

6.1 Inicialização do conjunto de controle

- Iniciamos $k = 0$.
- Dado um problema inicial (12), faremos $A^k = A$, $b^k = b$, $c^k = c$
- Solução parcial $x^k = 0$;
- Valor objetivo $z^k = 0$
- Dado o problema (12) definiremos a relação de restrições não-básicas $\mathcal{N}_i^k = i$ sendo $i = \{1, ..., n\}$ assim como a relação de restrições básicas $\mathcal{B}_j^k = j$ onde $j = \{n+1, ..., n+m\}$ referentes a não-negatividade.

6.2 Verificação de otimalidade

- Defina o conjunto $\mathcal{R}^k = \{i | b_i^k > 0\}$. Se $\mathcal{R}^k = \emptyset$, estamos em um ponto factível, como c é mantido positivo este ponto. Se $\mathcal{R}^k \neq \emptyset$ selecione um índice de restrição $i \in \mathcal{R}^k$.

6.3 Verificação de factibilidade

- Dado a restrição $A_{i\bullet}^k x \geq b_i^k$, obtenha o conjunto $\mathcal{V}^k = \{i | A_{i\bullet}^k > 0\}$. Se $\mathcal{V}^k = \emptyset$, então o problema é infactível, pois temos uma restrição que possui intercessão nula com a não-negatividade do problema da iteração atual. Se $\mathcal{V}^k \neq \emptyset$ selecione um índice de variável $j = \operatorname{argmin}_l \{c_i / A_{il} | l \in V^k\}$.

6.4 Translação

- Defina o ponto de translado: $p = b_i^k / A_{ij}$.
- Aplique a translação: $b^{k+1} = b^k - A_{\bullet j}^k p$.
- Calcule o valor atual da função objetiva: $z^{k+1} = z^k + c_j p$.

6.5 Mudança de base

- A restrição $\mathcal{N}_i^k$ é escolhida para entrar na base e a restrição $\mathcal{B}_j^k$ é escolhida para sair da base. Assim $\mathcal{N}_i^{k+1} = \mathcal{B}_j^k$ e $\mathcal{B}_j^{k+1} = \mathcal{N}_i^k$.
- Mudaremos o problema para a base B selecionada porém não é nescessário aplicar a inversão da base.
- Construa um A^{k+1} por meio de um pivoteamento por colunas em com pivô A_{ij}^k aplicado a A^k, o que equivale a fazermos $A^{k+1} = A^k B^-1$.
- Insira a troca de restrições não-básicas a matriz A^{k+1} adicionando o vetor de pivoteamento a linha A_i^{k+1}, o que equivale a $A_i^{k+1} = I_j B^-1$.
- Aplicando o mesmo pivoteamento ao vetor de custo temos c^k obtemos o equivalente a $c^{k+1T} = c^{kT} B^-1$.

6.6 Fechando iteração

- $k = k + 1$
- Faça $x^k = 0$ e então para todo i tal que $\mathcal{N}_i^k > n$ tomaremos $j = \mathcal{N}_i^k - n$ e $x_j^k = -b(j)$.
- Retorne ao passo (6.2).

7 Implementação

Foi feita uma implementação para validação deste método e comparação de resultados com o algoritmo Simplex da função linprog, encontrada no kit de otimização do MATLAB®. Algumas adaptações foram nescesárias pelas dificuldades de formato de problema e abordagem de erros numéricos.

7.1 Problemas NETLIB

Foram utilizados os problemas do banco de dados NETLIB. Foram selecionados os problemas que continham $c \geq 0$, menos de 3mil variáveis. Ainda foram eliminados problemas com váriaveis irrestritas. Assim o problema abordado foi:

$$
\begin{array}{rrcl}
\min & c^T x & & \\
\text{s.a} & Ax & = & b \\
& x & \geq & l \\
& x & \leq & u,
\end{array}
\tag{12}
$$

Onde l e u são respectivamente os limites inferiores e superiores.

A adaptação realizada foi a simples translação do problema para os limites inferiores e considerando os limites superiores como restrições de desigualdade ordinárias

7.2 Critério de seleção

Como critério de seleção foi utilizado o maior b_i.

7.3 Tratamento para igualdades

Como adaptação para o sistema de igualdades e desigualdades devemos buscar uma restrição candidata do grupo de igualdades e uma do grupo de desigualdades. E entre elas selecionamos a de pivô numéricamente mais seguro. Ainda, ao adicionar uma restrição de igualdade teremos que esta não deve sair da base.

8 Resultados e análise

8.1 Resultados

Seguem-se as tabelas referentes aos resultados obtidos:

	Restrições	Variáveis	Não-nulos	Valor ótimo			
BEACONFD	173	295	3408	3.3592485807E+04			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.104432	19	3.3592485807E+04	3.065303E-11	0	0	Ponto ótimo
Cones:	0.094804	160	3.3592485807E+04	2.078463E-08	0	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
BRANDY	220	303	2202	1.5185098965E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.409493	91	1.5185098965E+03	1.729254E-12	2.623549E-15	0	Ponto ótimo
Cones:	0.542132	498	1.5185098965E+03	2.853088E-09	7.504999E-14	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
FFFFF800	524	1028	6401	5.5567961165E+05			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	1.136331	193	5.5567956482E+05	2.624829E-10	1.852103E-12	0	Ponto ótimo
Cones:	44.736399	2192	**1.3963537125E+21**	**1.396354E+21**	**3.888481E+08**	0	Infactível
Variáveis Discordantes:	602						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
FIT1P	627	1677	9868	9.1463780924E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	5.264884	770	9.1463780924E+03	7.285442E-12	6.889692E-15	0	Ponto ótimo
Cones:	8.451517	1773	9.1463780924E+03	**4.626685E+02**	0	**2.732158E+01**	Ponto ótimo
Variáveis Discordantes:	14						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
GFRD-PNC	616	1160	2445	6.9022359995E+06			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.429162	189	**6.7107556445E+06**	3.580110E-11	**8.965779E+03**	0	Ponto ótimo
Cones:	0.639854	599	6.9022359995E+06	6.654381E-11	0	1.214566E-14	Ponto ótimo
Variáveis Discordantes:	406						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
QAP8	912	1632	7296	2.0350000000E+02			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	23.537730	4799	**2.0349950551E+02**	2.370621E-14	**9.623578E-06**	0	Ponto ótimo
Cones:	1109.848238	14896	2.0350000000E+02	4.034417E-13	1.828427E-12	0	Ponto ótimo
Variáveis Discordantes:	1024						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCORPION	388	466	1534	1.8781248227E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.244984	36	**-8.7585232287E-16**	3.868084E-16	**1.799404E+00**	0	Ponto ótimo
Cones:	0.243805	398	1.8781248227E+03	9.424136E-15	2.508273E-15	0	Ponto ótimo
Variáveis Discordantes:	318						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCSD1	77	760	2388	8.6666666743E+00			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.233915	123	8.6666666743E+00	1.995456E-16	3.293378E-16	0	Ponto ótimo
Cones:	0.319492	142	8.6666666743E+00	2.756507E-15	1.235821E-15	0	Ponto ótimo
Variáveis Discordantes:	13						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCSD6	147	1350	4316	5.0500000078E+01			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.463957	281	5.0500000075E+01	3.543189E-15	1.597118E-09	0	Ponto ótimo
Cones:	4.169918	510	**5.0172779200E+01**	**3.801733E+00**	1.573277E-08	0	Ponto ótimo
Variáveis Discordantes:	118						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCSD8	397	2750	8584	9.0499999993E+02			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	1.382688	654	9.0499999943E+02	4.231474E-14	7.739547E-08	0	Ponto ótimo
Cones:	115.470086	2094	**9.2379333759E+02**	**5.785855E+01**	6.110114E-09	0	Ponto ótimo
Variáveis Discordantes:	406						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCTAP1	300	660	1872	1.4122500000E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.683924	128	1.4122500000E+03	1.315802E-13	3.607078E-14	0	Ponto ótimo
Cones:	1.819357	572	1.4122500000E+03	1.552282E-12	1.263054E-13	0	Ponto ótimo
Variáveis Discordantes:	135						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SCTAP2	1090	2500	7334	1.7248071429E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	8.213234	645	1.7248071429E+03	3.380430E-13	1.348107E-13	0	Ponto ótimo
Cones:	38.657507	1183	1.7248071429E+03	3.240304E-12	1.953253E-13	0	Ponto ótimo
Variáveis Discordantes:	185						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SEBA	515	1036	4360	1.5711600000E+04			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.520823	52	1.5711600000E+04	4.257869E-12	1.421302E-14	0	Ponto ótimo
Cones:	0.294077	442	1.5711600000E+04	0	0	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SHELL	536	1777	3558	1.2088253460E+09			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.588331	156	1.2088253460E+09	0	0	0	Ponto ótimo
Cones:	1.315598	689	1.2088253460E+09	0	0	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SHIP04L	402	2166	6380	1.7933245380E+06			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.568334	118	1.7933245380E+06	1.692583E-13	1.888293E-14	0	Ponto ótimo
Cones:	0.680427	611	1.7933245380E+06	1.698001E-13	4.797153E-14	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SHIP04S	402	1506	4400	1.7987147004E+06			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.265894	78	1.7987147004E+06	6.653675E-14	1.074514E-14	0	Ponto ótimo
Cones:	0.433597	540	1.7987147004E+06	1.400597E-13	8.989251E-15	0	Ponto ótimo
Variáveis Discordantes:	4						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SHIP08S	778	2467	7194	1.9200982105E+06			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.421359	175	1.9200982105E+06	9.873499E-14	0	0	Ponto ótimo
Cones:	0.980414	767	1.9200982105E+06	1.613268E-13	4.440892E-16	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SHIP12S	1151	2869	8284	1.4892361344E+06			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.601087	144	1.4892361344E+06	7.925980E-14	1.194780E-15	0	Ponto ótimo
Cones:	1.658431	1233	1.4892361344E+06	7.459005E-13	6.754688E-15	0	Ponto ótimo
Variáveis Discordantes:	0						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
SIERRA	1227	2735	8001	1.5394362184E+07			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.541301	162	1.5369911640E+07	4.473403E-12	5.628285E+03	0	Ponto ótimo
Cones:	2.434703	1299	1.5394362184E+07	1.531540E-11	7.246143E-14	0	Ponto ótimo
Variáveis Discordantes:	340						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
STANDATA	359	1274	3230	1.2576995000E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.174664	29	1.7151000000E+02	3.110631E-13	2.509980E+01	0	Ponto ótimo
Cones:	0.361498	169	1.2576995000E+03	1.679834E+03	9.493378E-15	1.414214E+00	Ponto ótimo
Variáveis Discordantes:	102						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
STANDMPS	467	1274	3878	1.4060175000E+03			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.526679	226	1.3027030000E+03	1.226556E-12	6.553671E+00	0	Ponto ótimo
Cones:	1.356505	333	1.4060175000E+03	1.679834E+03	9.254116E-15	1.414214E+00	Ponto ótimo
Variáveis Discordantes:	190						

	Restrições	Variáveis	Não-nulos	Valor ótimo			
WOOD1P	244	2595	70216	1.4429024116E+00			
	Tempo	**Iterações**	**Valor ótimo**	**Resíduo Igualdade**	**Resíduo L.I.**	**Resíduo L.S.**	**Situação retornada**
MATLAB (Simplex):	0.747138	284	1.4429024116E+00	1.569809E-13	1.337092E-14	0	Ponto ótimo
Cones:	9.846218	328	1.4429024116E+00	1.012259E-11	7.063821E-15	0	Ponto ótimo
Variáveis Discordantes:	0						

8.2 Análise

O algoritmo equiparou-se ao Simplex em acertos, o algoritmo teve desempenho numérico levemente inferior e Maior número de iterações.

9 Comentários finais

O ponto de vista apresentado se mostrou válido e os resultados satisfatórios sobre o ponto de vista teórico. O algoritmo implementado esta longe de ser usável na prática no entanto já demonstra que a abordargem é funcional.

O demasiado número de iterações do algoritmo em comparação ao declarado pelo linprog provavelmente se dá em parte pela ausência de tratamento especial para variáveis canalizadas e o fato da não declaração sobre o número de iterações de fase 1 por parte do linprog.

É nescessário realizar um estudo mais detalhado do problema genérico de programação linear e contribuições possíveis que esta abordagem pode trazer. Tendo em vista que até agora o estudo aprofundado só foi realizado para o para o problema inicial proposto.

Referências

[1] M.S. Bazaraa, J.J. Jarvis & H.D. Sherali, *Linear Programming and Network Flows*, John Wiley, 2004.

[2] H.H. Domingues, C.A. Callioli & R.C.F. Costa, *Álgebra Linear e Aplicações*, Atual, 1990.

[3] Lucent Bell Laboratories, *NETLIB: Electronic Mail Distribution of Linear Programming Test Problems*.