ESTRATÉGIAS PARA OTIMIZAÇÃO LINEAR SEMI-INFINITA, COMBINADAS COM MÉTODO DE PONTOS INTERIORES

Matheus da Silva Menezes
Universidade Federal do Semi-Árido (UFERSA)
matheus@ufersa.edu.br

Roberto Quirino do Nascimento
Departamento de Estatística (DE)
Universidade Federal da Paraíba (UFPB)
quirino@de.ufpb.br

Marco César Goldbarg
Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
gold@dimap.ufrn.br

Elizabeth Ferreira Goldbarg
Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
beth@dimap.ufrn.br

Resumo: Este trabalho tem por objetivo apresentar técnicas de discretização tradicional e discretização com troca, aplicadas na resolução de problemas de otimização linear semi-inifinita. A estratégia geral adotada é a de utilizar tais técnicas para transformar o problema semi-infinito em uma sequencia de problemas de otimização linear, onde podemos aplicar um método de pontos interiores primal-dual para solucionar o mesmo. Testes numéricos de problemas já consolidados na literatura são efetuados utilizando estes métodos, de forma a comparar com as soluções numéricas encontradas por outros algoritmos, mostrando a eficiência dos algoritmos aqui propostos.

Palavras Chave: Otimização Linear Semi-Infinita, Programação Linear, Métodos de Discretização e Troca.

Abstract: This paper aims at presenting traditional techniques of discretization and exchange applied to solving linear semi-infinite optimization problems. The general strategy adopted is to use such techniques to transform the linear semi-infinite problem into a sequence of linear optimization problems, where we can apply primal-dual interior point method to solve it. Numerical testing problems already established in the literature are made using these methods, in order to compare with the numerical solutions found by other algorithms, showing the efficiency of the algorithms proposed here

Palavras Chave: Linear Semi-Inifinite Optimization, Linear Programming, Discretization Methods and Exchange Methods.

1 Introdução

Problemas de Otimização podem ser vistos, de forma geral, como problemas que visam maximizar ou minimizar uma função numérica, denominada de função objetivo, que possui um determinado número de variáveis, as quais podem ou não estar sujeitas a certas limitações ou restrições.

Embora as técnicas de otimização já sejam tradicionais em muitas áreas há décadas, nas últimas três surgiram vários novos e importantes problemas de otimização, provenientes dos mais variados campos, tais como: física, economia, computação e administração dentre outros, os quais possuem

uma grande aplicação prática em operações governamentais, comerciais, industriais e militares, por exemplo.

Com o surgimento dessas demandas, foram desenvolvidos novos métodos e técnicas para resolver os mesmos. Contudo, certos tipos de problema possuem algumas peculiaridades, e tais técnicas devem ser ajustadas e/ou complementadas para atender aos seus objetivos.

Dentro do contexto citado, o presente artigo aborda técnicas de solução para a Otimização Linear Semi-Infinita. Este tipo de problema caracteriza-se por apresentar, em sua composição, um número infinito de restrições possíveis. Como consequência, é usual que tais problemas sejam submetidos a um processo de adequação para que possam, então, ser resolvidos com o uso de técnicas tradicionais de resolução de problemas de otimização.

Considerando esses aspectos gerais, o presente trabalho é desenvolvido da seguinte forma: a seção 2 apresenta uma breve introdução sobre o problema de otimização semi-infinito, apresentando sua formulaçao padrão. A seção 3 apresenta um breve resumo sobre o estado da arte da solução. Na seção 4 são apresentados os algorimos propostos, bem como a sua fundamentação. Finalmente, na seção 5 é relatada uma implementação numérica e computacional dos métodos propostos em que alguns problemas já consolidados da literatura são solucionados. Os resultados do experimento são então analisados em confrontação com os vários métodos descritos inicialmente. Ao final são apresentadas as conclusões alcançadas no experimento.

2 Definição do Problema

Um problema de programação linear (PPL) é um problema de otimização em que a função objetivo e as restrições são expressas por meio de funções lineares. Embora não exista uma forma padrão para sua representação e sim pares de problemas relacionados pela teoria da dualidade, podemos considerar, sem perda de generalidade, que o modelo primal é um problema de minimização, expresso da seguinte forma:

$$\begin{aligned} \text{Minimize } c^t x, \\ \text{Sujeito a: } Ax = b, x \geq 0 \\ x \in \mathbb{R}^n \end{aligned} \tag{1}$$

onde c é um vetor pertencente ao $\mathbb{R}^n$, A é uma matriz $m \times n$, b e um vetor pertencente ao $\mathbb{R}^m$ e x pertencente ao $\mathbb{R}^n$ é uma solução para o problema. Caso o vetor x satisfaça as condições impostas pelas restrições, tal solução é dita viável.

Um problema de otimização *semi-infinito*, segundo *Hettich & Kortanek (1993)*, pode ser representado de uma maneira geral da seguinte forma:

$$\begin{aligned} v(P) = \max\{F(z) : z \in Z^P\} \text{ com} \\ Z^P = \{z : g(z,t) \leq 0, t \in B\} \subset \mathbb{R}^n \end{aligned} \tag{2}$$

onde B e um conjunto compacto e as funções F, g são continuamente diferenciáveis com respeito a z, em todo o $\mathbb{R}^n$ e $\mathbb{R}^n \times \mathbb{R}^m$, respectivamente.

Um problema de otimização linear semi-infinito, é definido por *Still & Lopes (2007)* da seguinte forma:

$$\text{Min } c^t x. \tag{3}$$

$$\text{Sujeito a: } a(t)^t x \geq b(t), \forall t \in B \tag{4}$$

onde B é um conjunto compacto.

Como podemos observar, este problema difere do problema de programação linear padrão, no sentido de que as restrições são indexadas por um parâmetro *"t"*, que por pertencer ao conjunto contínuo B, pode indexar infinitos valores. Desta forma, podemos ter infinitas restrições, fato este gerador de nosso problema semi-infinito.

3 Estado da Arte da Solução

De uma forma geral, os métodos para resolução de problemas semi-infinitos desenvolvidos visam *"converter"* um problema linear semi-infinito em um problema de programação linear na forma padrão, através da transformação do mesmo em vários subproblemas, substituindo o conjunto semi-infinito por um finito. Existem várias formas de fazer isso, sendo as mais comuns através de métodos de discretização, métodos de troca, e métodos baseados em redução local. (ver *Hettich & Kortanek (1993)* para mais detalhes), cujas variações são os métodos primais (ou de direções viáveis) métodos duais (ou métodos KKT) e métodos baseados em planos de corte. (ver *Still & Lopes (2007)* para maiores esclarecimentos).

Os métodos baseados em discretização são evidenciados por *Lin, Fang & Wu (1996)*, utilizando planos de corte e *Anderson, Goberna & Lopez (2001)*, com métodos baseados em simplex. *Leon, Sanmatias & Vercher (2000)* trataram do problema linear semi-infinito com metodologia híbrida, baseada em simplex. *Ito, Wu, Shiu & Teo(2010)* também apresentaram um método baseado em discretização e planos de corte em seu trabalho.

Ito, Liu & Teo (2000) apresentaram um método de redução baseado na dualidade de Wolfe com bons resultados práticos. *Bhattacharjee, Green Jr & Barton (2005)* utilizam métodos intervalares para resolver problemas lineares, acoplado ao CPLEX (baseado em SIMPLEX).

Ferris & Philpott(1998) utilizaram um método de pontos interiores baseado em direções viáveis para resolver alguns problemas. *Sheu, Wu & Fang (1995)* propuseram um algoritmo de pontos interiores primal dual resolver problemas de otimização semi-infinita linear, também baseado em direções viáveis.

Não é de conhecimento dos autores a abordagem de utilizar um método de discretização, ou um método de troca para reduzir o problema linear semi-infinito a um PPL na forma padrão e utilizar um método de pontos interiores preditor-corretor para solucionar o problema de otimização linear gerado, o que justifica o presente trabalho.

4 Algoritmos Propostos

De acordo com a literatura (ver *Hettich & Kortanek (1993)*, e *Still & Lopes (2007)*), as duas formas listadas a seguir podem ser utilizadas para reduzir o problema de programação semi-infinita a um problema de programação linear padrão:

1. Método de Discretização
2. Método de Troca

Os fundamentos matemáticos dessas estratégias são detalhados no item 4.1.

4.1 Fundamentos Matemáticos

De acordo com *Hettich & Kortanek (1993)*, o principal problema na formulação de sistemas semi-infinitos consiste na forma de discretizar e reduzir localmente o espaço de soluções (ou seja, o conjunto B) que é contínuo, de forma a solucionar o problema corretamente.

Dessa forma iremos considerar o *problema discreto* ou a *menor aproximação* do espaço de soluções:

$$Z^P = \{z : g(z,t) \leq 0, t \in B\} \tag{5}$$

do problema semi-infinito (P) pela substituição finita de várias restrições.

Ainda de acordo com *Hettich & Kortanek (1993)*, o caminho mais simples para obter soluções de um sistema semi-infinito é através da discretização ordinária do conjunto B, da seguinte forma:

Escolha $\overline{B} \subset B, |\overline{B}| < \infty$, e troque Z^P por:

$$Z^P(\overline{B}) = \{z : g(z,t) \leq 0, t \in \overline{B}\} \tag{6}$$

e considere o problema aproximado:

$$(P(\overline{B})) = \max\{F(z) : z \in Z^P(\overline{B})\} \tag{7}$$

Nesse sentido, $\overline{B}$ consiste em uma discretização do conjunto contínuo B e é alterado a cada iteração, sendo tipicamente, $\overline{B}$ denominado de *malha ou grade*.

Sem perda de generalidade, o problema *semi-infinito* em sua forma geral, pode ser reescrito da seguinte forma:

$$P(\overline{B}) = \max\{F(z) : z \in Z^P(\overline{B})\} \text{ com,} \tag{8}$$
$$Z^P(\overline{B}) = \{z : g(z,t) \leq 0, t \in \overline{B}, \|z\|_\infty \leq \Gamma\} \tag{9}$$

Onde assumimos que existe um $\Gamma < \infty$ tal que:

$$Z^P \subset \{z : \|z\|_\infty \leq \Gamma\} \tag{10}$$

Isto implica que Z^P é um conjunto compacto contido em uma bola de centro na origem, com raio Γ. A restrição (linear) adicional $\|z\|_\infty \leq \Gamma$ assegura que $P(\overline{B})$ tem uma soluçao para todo $\overline{B} \subset B$.

As condições de otimalidade de primeira e segunda ordem do problema são abordadas por *Hettich & Kortanek (1993)*, *Goberna & Lopez (2002)*, *Goberna & Lopez (2007)* e *Vazquez, Ruckmann, Stein & Still (2008)*, e embasam os resultados apresentados a seguir:

Proposição 1 (Condição Necessária de Otimalidade Primal)**.** *Seja $\overline{z}$ um ponto ótimo local para (P). Então não existe uma direção de subida estritamente viável.*

Definição 1. *As condições de* Karush-Kuhn-Tucker (KKT) *que são necessárias para que exista uma solução* primal dual *do problema linear é caracterizada pelo seguinte sistema:*

$$\begin{aligned} Ax &= b \; com \; x \geq 0 && (11)\\ A^t y + s &= c, \; com \; s \geq 0 && (12)\\ x_i s_i &= 0, \; com \; i = 1,2,\ldots,n && (13) \end{aligned}$$

Teorema 4.1. *Seja $\bar{x}$ uma solução para o problema de programação linear semi-infinito, que satisfaz as condições de KKT, conforme a definição . Então $\bar{x}$ é um minimizador global para o problema.*

Já a dualidade do problema semi-infinito é discutida em *Hettich & Kortanek (1993)*, *Goberna & Lopez (2002)*, *Lopez & Still (2007)* e *Vazquez, Ruckmann, Stein & Still (2008)*, o qual nos diz que no caso de um conjunto finito $B = \{t_1, ...t_r\}$, podemos proceder como no caso do *Problema Dual de Programação Linear*, isto é, determinar o ínfimo de $\sum_{i=1}^{r} b(t_i)\mu(t_i)$ sobre o conjunto de multiplicadores não-negativos $\mu(t_i)$, com $c = \sum_{i=1}^{r} a(t_i)\mu(t_i)$. Dessa forma o problema dual associado pode ser definido como:

$$\text{Procurar } v(D) = \inf\left\{\sum_{t\in B} b(t)\mu(t) : \mu \in Z^D\right\} \text{ onde} \tag{14}$$

$$Z^D = \left\{\mu \in \mathbb{R}_+^{(B)} : \sum_{t\in B} a(t)\mu(t) = c\right\} \tag{15}$$

E cuja afirmação seguinte é válida: se os problemas Primal e Dual tiverem soluções viáveis, então para toda solução temos:

$$c^t z \leq \sum_{t\in B} b(t)\mu(t) \tag{16}$$

Finalmente, as condições de estabilidade do problema são vistas em *Cánovas, Lopez & Parra (2002)*, *Cánovas, Lopez, Parra & Toledo (2007)*, *Hantoute & Lopez (2008)* e *Fan, Cheng & Wang (2011)*.

4.2 Algoritmos

4.2.1 Método de Discretização

De uma forma geral, discretizar pode ser entendido como a conversão de uma grandeza contínua em uma sequência de intervalos discretos, sem perder, contudo, a representatividade desta grandeza. Seguindo esse princípio, o método de discretização funciona baseado no conceito de subdividir o conjunto B do problema semi-infinito (que é contínuo) em um subconjunto $\overline{B} \subset B$, tal que $\overline{B}$ é discreto e contém todas as discretizações efetuadas no conjunto B. A cada iteração, o conjunto $\overline{B}$ é atualizado, através de um processo de refinamento, tornando-o mais subdividido dentro do seu intervalo de definição.

O tamanho da grade pode ser definido como a quantidade de partes que o conjunto B é subdividido, e é um fator de fundamental importância no estudo deste método. A cada iteração, o tamanho da grade aumenta e sua amplitude diminui, dando origem a um conjunto $\overline{B}$ mais "próximo"do conjunto B original, ou seja, quanto mais próximos os intervalos discretizados, mais fidedignamente representamos de forma discreta o conjunto B.

Segundo *Lopez & Still (2007)*, dizemos que $(P) \equiv P(\overline{B})$ é chamado *finitamente reduzível* se existe uma malha $\overline{B} \subset B$ tal que:

$$v(P) \equiv v(P(\overline{B})) \tag{17}$$

e (P) é dito *fracamente discretizável* se existe uma sequência de malhas $\overline{B}_k$, tal que:

$$v(P) \to v(P(\overline{B}_k)) \tag{18}$$

Sejam $\rho(\overline{B}_k)$ a distância de Hausdorff entre B e B_k e S o conjunto dos minimizadores de B_k. Então, (P) é dito discretizável se para cada sequência de malhas $\overline{B}_k$, satisfazendo $\rho(\overline{B}_k) \to 0$, tivermos:

1. $P(\overline{B}_k)$ possui solução para um k suficientemente grande;
2. Para cada sequência de soluções $\bar{z}_k \in S(\overline{B}_k)$, tivermos:

$$d(\bar{z}_k, S(B)) \to 0 \text{ e } v(P(\overline{B}_k)) \to v(P), \text{ se } k \to \infty \tag{19}$$

onde $d(x,S) := \min_{x \in S} \|x - s\|$

Um problema convexo (P) é redutível se existe um minimizador que satisfaça a condição de KKT para um subconjunto $\overline{B} = [t_1, ..., t_n]$ no conjunto formado pelos índices cujas restrições estão ativas, com $k \leq n$. Neste trabalho, consideramos que o problema é discretizável. Uma condição para que o problema seja discretizável pode ser encontrada em *Lopez & Still (2007)*, a qual transcrevemos a seguir:

Proposição 2. *Seja P um problema linear semi-infinito contínuo, com conjunto viável F limitado. Então P é discretizável.*

Tomando o problema de otimização linear semi-infinito, temos:

$$\begin{array}{ll} \text{Min} & c^t x \\ \text{Sujeito a:} & f_i(t)x_i \leq bi \\ & t \in [a,b] = B \end{array} \tag{20}$$

O presente trabalho visa implementar o processo de discretização anteriormente definido, de acordo com a forma descrita no algoritmo 1 a seguir:

Algoritmo 1: Método de Discretização

Dados:	$B = [a,b]$ um conjunto compacto
	n = número limite de iterações
	ε = a precisão adotada
Passo 1:	Faça $k = 0$;
Passo 2:	Faça uma discretização no conjunto B, subdividindo-o de forma ordinária em $2^{k+1} + 1$ partes;
Passo 3:	Encontre uma solução x_k de $P(\overline{B})$ (utilizando método de pontos interiores);
Passo 4:	Verifique se satisfaz os critérios de parada:
4.1.	$k > n$
4.2.	Se $k > 0$, $\lvert x_{k-1} - x_k \rvert < \varepsilon$
Passo 5:	Se satisfaz algum dos critérios de parada, FIM
	Senão, faça $k = k + 1$ e volte para o passo 2.

Observe que a idéia preconizada pela proposição 1 é utilizada para definir o critério de parada 4.2. do algoritmo 1, pois se a solução entre duas iterações consecutivas não melhorar, então encontramos um ótimo local.

4.3 Método de troca

Como é de conhecimento na programação linear, as restrições que não estão ativas no ponto ótimo (ou seja, que possuem folga), não interferem na solução ótima do PPL. Baseado nesse conceito, o método de troca é uma alternativa eficiente na resolução de problemas de otimização semi-infinita linear, quando comparado ao método de discretização tradicional, uma vez que, a cada iteração, as restrições que não estão ativas são desconsideradas do problema gerado na nova iteração, e são criadas novas restrições apenas no entorno dos pontos ativos da solução do problema.

Dessa forma, ao nos referirmos a um *"algoritmo de troca"*estamos nos referindo a um procedimento que discretiza o espaço de busca do PPL no entorno das restrições ativas da melhor solução corrente. Assim sendo, no decurso do processo de solução, é possível desconsiderar as restrições que não estão ativas na solução corrente, ou seja, as que possuem folga.

Conceitualmente, o algoritmo de troca desenvolvido no presente trabalho pode ser definido da seguinte forma:

Algoritmo 2 Método de Troca Conceitual

Dados:	$B = [a,b]$ um conjunto compacto
	n = número limite de iterações
	ε = a precisão adotada
Passo 1:	Faça $k = 0$;
Passo 2:	Faça uma discretização no conjunto B, subdividindo-o de forma ordinária em $2^{n+1}+1$ partes;
Passo 3:	Encontre uma solução x_k de $P(\overline{B})$ (utilizando método de pontos interiores);
Passo 4:	Verifique se satisfaz os critérios de parada:
4.1.	$k > n$
4.2.	Se $k > 0$, $\lvert x_{k-1} - x_k \rvert < \varepsilon$
Passo 5:	Se satisfaz algum dos critérios de parada, FIM
	Senão, escolha $\overline{B}_{k+1} = \overline{B}_k \cup t_k$ onde k é uma restrição ativa (ou seja, faça uma discretização apenas em torno dos índices cujas restrições estão ativas) dividindo o entorno do ponto em uma grade de tamanho $\frac{b-a}{2^{k+1}+1}$
	faça $k = k+1$ e volte para o passo (3)

O algoritmo 2, descrito anteriormente, supostamente possui um potencial de melhor desempenho, uma vez que, em relação ao método de discretização tradicional representado no algoritmo 1, empregará um menor número de restrições decorrentes do método de discretização.

Desta forma, segundo *Hettich & Kortanek (1993)*, denotando por $v(z)$ o valor ótimo, temos o seguinte resultado:

Teorema 4.2. *Assuma que exista um $\Gamma > 0$ com (10) e cada passo $v(z^i) = g(z^i, t^j)$ ocorre para ao menos um $j \in [1,\ldots,r_i]$. Então ou o Algoritmo 2 para após um número finito de passos com a solução de (P) ou a sequência $\{z^v\}$ tem ao menos um ponto de acumulação, e este soluciona (P).*

4.4 Exemplos Numéricos

Com a finalidade de exemplificar o funcionamento dos algoritmos 1 e 2 propostos, foram implementados no presente estudo os problemas originados de *Leon, Sanmatias & Vercher (2000)*, que

reproduzimos a seguir:

CC. minimize x_2 s.t. $x_1\cos(s)+x_2\sin(s)\leq 1, s\in[0,2\pi]$,

PT. minimize x_1 s.t. $x_1+(1-s-s^2)x_2\geq s-s^2, s\in[0,1]$,

GL. minimize $2x_1+x_2$ s.t. $sx_1+(1-s)x_2\geq s-s^2, s\in[0,1]$,

GU2. minimize $x_1+(1+\varepsilon)x_2$ s.t. $x_1+sx_2\geq -1/(1+s), s\in[0,1]$,

GU3. minimize $x_1+(1/2)x_2+((1/2)+\varepsilon/3)x_3$ s.t. $x_1+sx_2+(s+\varepsilon s^2)x_3\geq e^x, s\in[0,1]$,

WBI. minimize $x_1-2x_2-x_3$ s.t. $x_1+sx_2+s^2x_3\geq s^5, s\in[1,2], x_j\geq 0, j=1,2,3$

TAN. minimize $\sum_{i=1,\ldots,n}x_i$ s.t. $\sum_{i=1,\ldots,n}s^{i-1}x_i, s\in[0,1]$,

4.5 Resultados obtidos na literatura

Originalmente, os problemas contantes em *Leon, Sanmatias & Vercher (2000)* foram solucionados utilizando-se dois métodos distintos, denominados *A1* e *A2*. Na tabela a seguir, reproduzimos os resultados obtidos pelos autores do referido artigo, onde n denota o número de variáveis originais do problema, *ALGO* denota qual o algoritmo utilizado, *IT* o número de iterações e o valor ótimo obtido em cada um dos métodos; o símbolo (!) quando apresentado ao lado da nomenclatura do algoritmo indica que para esta situação, o algoritmo não foi capaz de achar uma solução ótima para o problema.

Tabela 1 - Resultados da Literatura

Problema	n	ALGO	IT	Valor Ótimo
CC	2	A1	6	-1
		A2	14	-0.99999999999996
PT	2	A1	2	0.2360679779
		A2	13	0.2360679775
GL	2	A1	1	0.666666667
		A2	14	0.666666667
GU2	2	A1	4	-0.6666663856
$\varepsilon=-0.5$		A2	15	-0.6666666667
GU3	3	A1	4	1.7262807025
$\varepsilon=1,5x10^{-6}$		A2(!)	2	1.7394272761
WBI	3	A1,A2	1	-2
TAN	3	A1	3	0.6490424215
		A2	24	0.6490420934
	6	A1	9	0.6160851913
		A2(!)	76	0.6160862020
	8	A1	24	0.6156532268
	10	A1	31	0.6156280585
	12	A1	17	0.6156265660

Como podemos observar na tabela 1, o algoritmo A2 não foi capaz de encontrar uma solução ótima para os problemas $GU2$ e TAN, quando $n\geq 6$. Já o algoritmo A1 conseguiu encontrar soluções ótimas em todas as instâncias.

5 Experimento Computacional

Com a finalidade de testar o potencial de desempenho dos algoritmos propostos no presente trabalho, foi realizado um experimento computacional considerando os problemas propostos no item 4.4. Os algoritmos foram implementados em linguagem C++, na plataforma *Windows 7*, em um PC com processador *intel Core 2 Duo*, com 2 GB de memória RAM.

O *problema linear semi-infinito* original é transformado, a cada iteração, em um *PPL*, o qual é solucionado por um algoritmo para resolução de problemas lineares. Na resolução deste último, foi utilizado um método de pontos interiores preditor-corretor, que se mostrou adequado na resolução dos problemas.

A discretização do conjunto B ocorre em ambos os métodos, e é realizada a uma taxa de $2^{n+1}+1$, onde n é o número da iteração considerada. O critério de parada para ambos os métodos, foi o número máximo de iterações limitado a 50 e também a variação da solução ótima entre as iterações, onde foi estabelecido como suficiente uma aproximação mínima de 1×10^{-5} entre as soluções ótimas de iterações consecutivas.

O método de discretização, como o próprio nome sugere, considera que em cada iteração, o conjunto B é apenas subdividido em várias partes, ou seja, discretizado, dando origem ao conjunto $\overline{B}$. Este $\overline{B}$ é uma família de conjuntos $\overline{B}_1 \subset \overline{B}_2 \subset \ldots \subset \overline{B}_n$ que representam de uma forma discreta o conjunto B que é contínuo.

Já no método de troca, apenas as restrições que não apresentaram folga, continuarão a fazer parte do problema gerado. Logo, o número de novas restrições acrescentadas ao problema é variável e depende das características individuais da solução do mesmo, ou seja, estamos subdividindo o conjunto B várias vezes a cada iteração, e a cada iteração um número de novas restrições é acrescentado em torno das que não apresentaram folga, enquanto as restrições que apresentam folga são simplesmente desconsideradas do PPL gerado na iteração.

O método utilizado para solucionar o problema finito linear gerado foi o *método de pontos interiores preditor-corretor*, considerando os seguintes parâmetros, válidos para ambos os métodos.

- Precisão de $1x10^{-5}$
- Número limite de iterações: 40
- Solução dos problemas primal e dual, cujo critério de parada adotado foi:

$$\max\left\{x^t s, 10^{-2}\left\|Ax-b\right\|, 10^{-2}\left\|A^t t+s-c\right\|\right\} \leq 10^{-5} \tag{21}$$

Ressaltamos que foi utilizada, por questões de simplicidade de implementação, a inversa da matriz abaixo para determinar a solução do sistema:

$$\begin{bmatrix} A & 0 \\ -SX^{-1} & A^t \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = -\begin{bmatrix} Ax-b \\ A^t y - c + \mu X^{-1} e \end{bmatrix}$$

Convém salientar que este solver de otimização linear foi testado com problemas de pequeno porte e também problemas da biblioteca NETLIB, mostrando-se eficiente e preciso na resolução dessas instâncias e, portanto, sendo capaz de atender de forma confiável aos problemas propostos no presente trabalho.

Na Tabela 2 - Resultados do Experimento apresentada a seguir, temos os resultados obtidos para cada um dos problemas a partir da utilização de cada método, onde destacamos o nome do problema, o algoritmo utilizado, o número de iterações semi-infinitas, o número de iterações lineares na resolução dos problemas gerados, o número de restrições na última iteração, o tempo gasto, e o valor ótimo alcançado.

Tabela 2 - Resultados do Experimento

Problema	ALGO	IT	IT LINEAR	Restrições	Tempo(ms)	Valor ótimo
CC	Discretização	2	11	9	39	-0.99999999939
	Troca	2	10	5	39	-0.99999998657
PT	Discretização	5	49	65	1547	0.23601402408
	Troca	7	7	5	86	0.23606569229
GL	Discretização	7	73	257	85516	0.66665649771
	Troca	8	6	5	85	0.66666489656
GU2	Discretização	2	14	9	47	-0.66666666585
$\varepsilon = -0.5$	Troca	2	11	5	43	-0.66666580604
GU3	Discretização	2	13	9	52	1.85914124227
$\varepsilon = 0$	Troca	2	13	5	44	1.85914091289
WBI	Discretização	2	48	9	83	-2.00000131035
	Troca	2	13	5	82	-2.00000929593
TAN	Discretização	7	51	257	69940	0.64901557375
$n = 3$	Troca	7	8	5	78	0.64904163853
$n = 6$	Discretização	3	32	17	110	0.61673333136
	Troca	12	9	5	150	0.61343440276
$n = 8$	Discretização	4	42	33	347	0.61581080251
	Troca	13	121	5	197	0.60162992794
$n = 10$	Discretização	4	47	33	421	0.61565473442
	Troca	13	126	5	232	0.59204677535

Como podemos observar na Tabela 2 - Resultados do Experimento verifica-se que o número de iterações semi-infinitas difere para cada método, de acordo com o problema abordado. O mecanismo do método de discretização determina que o número de restrições que devem ser consideradas sempre aumenta a cada iteração, o que vai tornando o problema linear mais restrito sob a penalidade de uma resolução mais custosa e, eventualmente, proibitiva, enquanto o método de troca acaba geralmente tendo um número maior de iterações, mas que são mais "leves"pois possuem um número menor de restrições. Outra variável que corrobora tal tese é o tempo gasto na resolução de cada problema, onde o método de troca possui grande vantagem nos casos em que o problema de discretização passa da terceira iteração. Conclui-se, portanto, que o método de discretização tradicional pode se tornar pouco eficiente, dependendo da instância solucionada. Podemos verificar também que no caso do uso do método de troca, o número de restrições geradas em uma mesma iteração é menor ou, no pior caso, igual ao número de restrições geradas pelo método de discretização.

Em relação aos valores alcançados para a função do problema, observamos que o método de troca proposto apresenta resultados qualitativamente semelhantes aos da literatura. Nesse caso a vantagem

do método de troca proposto sobre o de discretização tradicional se caracteriza por necessitar um menor número de restrições e, consequentemente, um menor numero de iterações para solucionar o mesmo problema. Tal característica concede ao método proposto um potencial para demandar menos tempo computacional na resolução dos problemas examinados, o que foi confirmado nos testes realizados.

Finalmente, a partir dos resultados numéricos obtidos, podemos inferir que a utilização de um método de pontos interiores preditor-corretor foi adequada para a finalidade proposta, que foi a de se obter a solução para o PPL gerado a cada iteração semi-infinita. Dessa forma, este método se coloca como uma alternativa viável aos métodos que se utilizam do SIMPLEX para a resolução do PPL gerado.

6 Referências

1. Anderson, Edward J.; Goberna, Miguel A. & López, Marco A. (2001) *Simplex like trajectories on quasi-polyedral sets*, Mathematics of Operations Research, Vol 26, n.1, pp.147-162.

2. Bhattacharjee, Binita; Green Jr, William H. & Barton,Paul I. (2005) *Interval Methods for Semi-Infinite Programs*, Computational Optimization and Applications, n.30, pp.63-93.

3. Bunch, J.R. & Parlett, B.N. (1971) *Direct Methods for solving Symetric Indefinit Systems of linear equations*, SIAM J. Numer. Anal. n.8, 639-655.

4. Cánovas, Maria J.; Lopez, Marco A. & Parra, Juan. (2002) *Stability in the Discretization of a Parametric Semi-Infinite Convex Inequality System*, Mathematics of Operations Research, vol.27, n.4, pp.755-774.

5. Cánovas, Maria J.; Lopez, Marco A.; Parra, Juan & Toledo, F. J. (2007) *Sufficient Conditions for Total Ill-Posedness in Linear Semi-Infinite Optimization*, European Journal of Operational Research, n.181, pp.1126-1136.

6. Fan, Xiaodong; Cheng, Caozong & Wang, Haijin. (2011) *Stability of Semi-Infinite Vector Optimization Problems Without Compact Constraints*, Nonlinear Analysis, n.74, pp.2087-2093.

7. Ferris, R.L. & Philpott, A.B. (1998) *An Interior Point Algorithm for Semi-Infinite Linear Programming*, Mathematical Program, n.43, pp.257-276.

8. Goberna, M. & Lopez, Marco A. (2002) *Linear semi-infinite programming theory: An updated survey*, European Journal of Operational Research, n.143, pp.390-405.

9. Hantoute, A. & Lopez, Marco A. (2008) *Caracterization of Total Ill-Posedness in Linear Semi-Infinite Optimization*, Journal of Computational and Applied Mathematics, n.217, pp.350-364.

10. Hettich, R & Kortanek, K.O. (1993) *Semi-Infinite Programming; theory, methods and applications*. SIAM Review, vol.35, n.3,pp.380-349.

11. Ito, Satoshi; Wu, Soon-Yi; Shiu, Ting-Jang & Teo, Kok Lay (2010) *A Numerical Approach to infinite-dimensional Linear Programming in L_1-Spaces*, Journal of Industrial and Management Optimization, Vol.6, n.1, pp.15-28.

12. Ito, Satoshi & Teo, Kok Lay (2000) *A reduction method for FM-Continuos Linear Semi-Infinite Problems*, Anals of Operational Research, pp.189-213.

13. Leon, T; Sanmatias, S.& Vercher, E. (2000) *On the Numerical Treatment of Linearly Constrained Semi-Infinite Optmization Problems*, European Journal of Operational Research, n.121, pp.78-91.

14. Lin, C.J.; Fang, S.C.& Wu, Soon-Yi. (1996) *Parametric Linear Semi-Infinite Programming*, Appl. Math. Lett., Vol.9, n.3, pp.89-96.

15. Lopez, Marco & Still, George. (2007) *Semi-Infinite Programming*,European Journal of Operational Research, n.180, pp.491-518.

16. Mészaros, C. (1996) The Efficient Implementation of Interior Point Methods for solving Linear Programming and this Applications, Ph.D. Thesis. Eötus Loránd University of Sciences, Budapest.

17. Sheu, R.L.; Wu, Soon-Yi & Fang, S.C. (1995) *A Primal-Dual Infeasible-Interior-Point Algorithm for Linear Semi-Infinite Programming*, Computers Math. Applic., Vol.29, n.8, pp.7-18.

18. Still,G.(1999) *Generalized Semi-Infinite Programming: theory and methods*. European Journal of Operational Research, n.119, pp.301-313.

19. Vazquez, F. Guerra; Ruckmann, J.J.; Stein, O. & Still, O. (2008) *Generalized Semi-Infinite Programming: A Tutorial*, Journal of Computational and Applied Mathematics, n.217, pp.394-419.