

UM MÉTODO DE PROJEÇÃO PARA PROBLEMAS DE COMPLEMENTARIDADE LINEAR

Cristiane Maria Alves Pissarra

Petronio Pulino

Instituto de Matemática, Estatística e Computação
Científica – IMECC

UNICAMP – Campinas – SP – Brasil

Resumo

Este artigo apresenta um algoritmo de dois passos para a resolução numérica de problemas de complementaridade linear (LCP). O algoritmo que apresentaremos combina a estratégia de conjunto ativo aliada ao método dos gradientes conjugados quadrático para a resolução do sistema linear reduzido. O desenvolvimento deste método de dois passos foi baseada em um outro, também de dois passos, desenvolvido por Kocvara & Zowe (1994), que combina o método de super relaxação sucessiva simétrica com projeção e o método dos gradientes conjugados pré condicionado. Na seção Experiência Numérica, utilizaremos o método para resolver LCPs com matrizes não singulares pertencentes às classes de matrizes semi positivas definidas, P-matrizes e P_0 -matrizes. Será feita ainda uma comparação entre o desempenho numérico do algoritmo apresentado neste texto e o método desenvolvido por Pardalos, Ye, Han & Kaliski (1993).

Palavras-chave: problema de complementaridade linear, método de conjuntos ativos, redução potencial.

Abstract

In this article we present an iterative two-step algorithm for the numerical solution of linear complementarity problems (LCP). The algorithm presented combine the active set strategy with the square conjugated gradient method for the linear system solution. This iterative two-step algorithm was based in a similar one developed by Kocvara & Zowe (1994), that combine the symmetric successive over relaxation with projection method and the preconditioned conjugated gradient method. In the Numerical Experience section, we have tested the method for solve some LCP with non-singular matrices. Such matrices belong to semi positive definite matrix class, P-matrix class and P_0 -matrix class. Furthermore we show a comparision between our method and the one developed by Pardalos, Ye, Han & Kaliski (1993).

Keywords: linear complementarity problems, actives sets methods, potential reduction.

1. Introdução

O núcleo deste trabalho consiste em se apresentar um método iterativo com estratégia de conjunto ativo, para resolução numérica de problemas de complementaridade linear com matrizes específicas, e que apresentou resultados superiores, quando comparados aos testes realizados com o método iterativo de pontos interiores de Pardalos, Ye, Han & Kaliski (1993), para as mesmas matrizes. A idéia central do método combina a estratégia de conjunto ativo aliada ao método dos Gradientes Conjugados Quadrático, para a resolução do sistema linear resultante, que se espera ser menor do que a dimensão do problema inicial, a cada passo.

O método apresentado aqui foi desenvolvido a partir de um outro método de dois passos apresentado por Kocvara & Zowe (1994) (que chamaremos de 2PASSOS), método este que combina o método da Super Relaxação Sucessiva Simétrica com Projeção (SSORP) (para determinar uma aproximação do conjunto ativo ótimo) e método dos Gradientes Conjugados com Pré condicionamento (PCG) (para a resolução dos sistemas lineares reduzidos resultantes). Detalhes deste método podem ser encontrados em Pissarra (1997).

Considere o Problema de Complementaridade Linear (LCP(A,b)) não singular, definido da seguinte forma:

Dados $A \in M_{n \times n}(\mathfrak{R})$ não singular e $b, c \in \mathfrak{R}^n$, encontrar $x \in \mathfrak{R}^n$ de modo que as seguintes condições sejam satisfeitas:

$$Ax - b \geq 0 \quad (1)$$

$$(P1) \quad x \geq 0 \quad (2)$$

$$x^t (Ax - b) = 0 \quad (3)$$

ou mostrar que tal vetor x não existe.

O problema de complementaridade linear não singular acima forma as condições necessárias de otimalidade de Karush-Kuhn-Tucker para o problema de programação quadrática, com $A \in M_{n \times n}(\mathfrak{R})$ simétrica e $b \in \mathfrak{R}^n$

$$\begin{aligned} \min \quad & \left\{ f(x) = \frac{1}{2} x^t A x - x^t b \right\} \\ \text{s.a} \quad & S = \{x \in \mathfrak{R}^n / x_i \geq 0, i = 1, \dots, n\}, \end{aligned}$$

com isto temos a equivalência entre o problema de programação quadrática acima e o problema de complementaridade linear. Os detalhes desta equivalência podem ser vistos em Pissarra (1997).

Iremos abordar LCPs onde $A \in M_{n \times n}(\mathfrak{R})$ pertence às seguintes classes de matrizes:

- *matrizes semi positivas definidas*: uma matriz $A \in M_{n \times n}(\mathfrak{R})$ é semi positiva definida (spd) se $x^t A x \geq 0$, para qualquer $x \in \mathfrak{R}^n$;
- *P-matrizes*: uma matriz $A \in M_{n \times n}(\mathfrak{R})$ é P-matriz se todos os seus menores principais são positivos;
- *P₀-matrizes*: uma matriz $A \in M_{n \times n}(\mathfrak{R})$ é P₀-matriz, se todos os seus menores principais são não negativos.

Os resultados que garantem a solução para cada um destes LCPs de matrizes pertencentes à estas classes, não serão mostrados aqui mas poderão ser encontrados em Cottle, Pang & Stone (1992).

Utilizaremos, no trabalho, a seguinte notação: $\alpha, \beta, \rho, \dots$ são constantes reais; x, y, \dots denotam vetores; x_i significa a i -ésima componente do vetor x . $x^{(k)}$ é a k -ésima iteração sucessiva de x , no método em particular. Portanto, $x_i^{(k)}$ é o i -ésimo componente da k -ésima iteração do vetor x . A_{ij} é a (i, j) -ésima componente da matriz A .

Iremos precisar ainda da seguinte definição:

Definição 1.1: Para cada $x \in S = \{x \in \mathbb{R}^n / x_i \geq c_i, i = 1, \dots, n\}$, chamaremos de **variáveis ativas** às componentes x_i para as quais $x_i = c_i$. Analogamente, chamaremos de **variáveis não ativas** as componentes x_i tais que $x_i < c_i$.

2. Métodos e Algoritmos

2.1 O Método dos Gradientes Conjugados Quadrático com Projeções (CGSP)

Este algoritmo consiste de uma combinação do método dos Gradientes Conjugados Quadrático – CGS, com a estratégia de conjunto ativo (onde, geralmente, os métodos baseados em estratégias de conjunto ativo só permitem que uma restrição seja adicionada ou retirada do conjunto ativo à cada busca unidimensional) e será aplicado em LCPs descritos na forma de (PI) cujas matrizes $A \in M_{n \times n}(\mathbb{R})$ sejam não singulares pertencentes às classes já definidas na *Seção 1*.

O método utiliza ainda a definição de um operador, \mathbf{P} , que guardará a informação sobre quais variáveis estão ativas ou não ativas no estágio k do algoritmo:

Definição 2.1: O operador $\mathbf{P}_{I(x)} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-p(x)}$, onde $p(x)$ denota a cardinalidade do conjunto ativo $I(x) = \{i / x_i = 0\}$, que leva um vetor $x \in \mathbb{R}^n$ ao vetor de dimensão $\mathbb{R}^{n-p(x)}$, obtido pela omissão das componentes de x com índices em $I(x)$ é definido como **operador de projeção vinculado ao conjunto ativo $I(x)$** .

O que fazemos para resolver o problema (PI) é, começando no estágio $k=0$, iterar $x^{(k)} \geq 0$, declarar *fixas* as variáveis i tais que $x_i^{(k)} = 0$ e $r_i^{(k)} = Ax_i^{(k)} - b \geq 0$. Declara-se *livres* as variáveis restantes e com isto tem-se definido o operador de projeção vinculado ao conjunto ativo $I(x)$.

As variáveis que estão não ativas são projetadas na fronteira do conjunto de restrição $S = \{x \in \mathbb{R}^n / x \geq 0\}$, Com isso estamos resolvendo o item (2) do problema (PI). Ainda nesse estágio, dentre as variáveis projetadas, procura-se quais estão violando a restrição (1) do problema (PI).

O próximo estágio é resolver o sistema resultante $\mathbf{P}_{I(x)} A \mathbf{P}_{I(x)} x = \mathbf{P}_{I(x)} b$ pelo método dos Gradientes Conjugados Quadrático (CGS), obtendo-se $x^{(k+1)}$. A matriz do novo sistema é obtida retirando-se a i -ésima linha e i -ésima coluna da matriz A , para $i \in I(x)$. O que se espera, é que durante o processo, o sistema linear a ser resolvido tenha dimensão menor

do que o do problema original, a cada estágio de resolução. A cada iteração k do método CGSP, é utilizada a aproximação adquirida pelo método CGS na iteração $(k-1)$ e é feita uma nova verificação das condições das variáveis, até que se consiga chegar à solução do problema (PI).

Para as classes de matrizes simétricas e positivas definidas e P-matrizes (incluindo matriz de Murty e matriz de Fathi) a matriz do sistema resultante $\mathbf{P}_{I(x)} A \mathbf{P}_{I(x)} x = \mathbf{P}_{I(x)} b$ é não singular. Para os exemplos de P_0 -matriz, não houve ocorrência de singularidade do sistema.

2.2 Algoritmo para o Método CGSP

Dado $x^{(0)} \in \mathfrak{R}^n$ uma aproximação inicial;
 $k = 0, 1, 2, \dots$

```

Enquanto  $\|r^{(k)}\| > \text{tolerância}$  faça
  para  $i = 0, 1, 2, \dots, n$  faça
    if  $x_i^{(k)} < 0$  then
       $x_i^{(k)} = 0$ 
    endif
    definir  $I(x^{(k)})$ 
     $r^{(k)} = Ax^{(k)} - b$ 
  para  $i = 0, 1, 2, \dots, n$  faça
    if  $i \in I(x^{(k)})$  e  $r_i^{(k)} < 0$  then
       $i$  é retirado de  $I(x^{(k)})$ 
    endif

```

Passo 2: Cálculo da nova aproximação

$$x^{(k+1)} = \text{CGS}^m(x^{(k)}, P_I A P_I, P_I b)$$

O método CGS é uma variação do método dos Gradientes Bi-Conjugados (BCG), aplicado para matrizes $A \in M_{n \times n}(\mathfrak{R})$ não singulares (Greenbaum (1997) e Fernandes (1993)). Para matrizes simétricas e positivas definidas, este método se comporta como o método dos Gradientes Conjugados, isto é, converge em n passos (Golub (1985)).

2.2.1 Algoritmo para o Método CGS

Dado $x^{(0)} \in \mathfrak{R}^n$ uma aproximação inicial;
 $k = 0, 1, 2, \dots$

$$g^{(0)} = b - Ax^{(0)}; \quad p^{(-1)} = 0; \quad q^{(0)} = 0; \quad \rho^{(-1)} = 1$$

$$\begin{array}{l}
 \text{Enquanto} \quad \|g^{(k)}\| > \text{tolerância} \quad \text{faça} \\
 \rho^{(k)} = g_0^t g^{(k)} \\
 \beta^{(k)} = \frac{\rho^{(k)}}{\rho^{(k-1)}} \\
 u^{(k)} = g^{(k)} + \beta^{(k)} q^{(k)} \\
 d^{(k)} = u^{(k)} + \beta^{(k)} (q^{(k)} + \beta^{(k)} d^{(k-1)}) \\
 v^{(k)} = A d^{(k)} \\
 \sigma = g^{(0)} v^{(k)} \\
 \alpha^{(k)} = \frac{\rho^{(k)}}{\sigma^{(k)}} \\
 q^{(k+1)} = u^{(k)} - \alpha^{(k)} v^{(k)} \\
 g^{(k+1)} = g^{(k)} - \alpha^{(k)} A(u^{(k)} + q^{(k+1)}) \\
 x^{(k+1)} = x^{(k)} + \alpha^{(k)} (u^{(k)} + q^{(k+1)}) \\
 k = k + 1
 \end{array}$$

2.3 O Método de Pardalos, Ye, Han e Kaliski (RP)

Para testarmos a eficiência numérica do método CGSP, vamos compará-lo com um método apresentado por Pardalos, Ye, Han & Kaliski (1993). O método de Redução Potencial (RP) utiliza a estratégia de reduzir uma determinada função potencial $\phi(x, y) = \mathfrak{R}_+^n \times \mathfrak{R}_+^n \mapsto \mathfrak{R}$, de pontos interiores, dada por:

$$\phi(x, y) = \rho \ln(x^t y) - \sum_{j=1}^n \ln(x_j y_j), \quad \rho = \rho(n) > n. \quad (4)$$

Para alcançar uma redução potencial, utiliza-se o método da projeção dos gradientes escalados, resolvendo o seguinte problema de programação linear, sujeito à restrição elipsoidal, na k -ésima iteração:

$$\min (\nabla_x \phi(x^{(k)}, y^{(k)})^t \delta x + \nabla_y \phi(x^{(k)}, y^{(k)})^t \delta y)$$

s.a

$$\delta y = A \delta x$$

$$\|(X^{(k)})^{-1} \delta x\|^2 + \|(Y^{(k)})^{-1} \delta y\|^2 \leq \beta^2 < 1$$

3. Experiência Numérica

É importante salientar que, para o método RP, utilizamos um vetor $(x^{(0)}, y^{(0)})$, que contém a aproximação inicial, pertencente ao interior da região factível definida em cada problema. Já no método dos Gradientes Conjugados Quadrático com Projeções, $(x^{(0)}, y^{(0)})$ pode ser inicializado em qualquer vetor. Ainda no método RP, há a necessidade de uma variável

($0 < \beta < 1$) que controla o tamanho do passo a cada iteração de descida e assegura a factibilidade estrita em cada iteração. Nos testes realizados, para valores de β próximos de 1 (mais especificamente, $\beta = 0.99$), o algoritmo converge mais rapidamente.

Nos testes que utilizaram subrotinas para criar valores aleatórios, a partir de uma semente, foram utilizados 5 exemplos diferentes, e na comparação de tabelas, utilizou-se portanto, a média de iterações e de tempos obtidos. A máquina utilizada para executar esses exemplos foi um microcomputador *Pentium 133*. A tolerância, quando não citada explicitamente, foi considerada como sendo $eps = 1.0e-07$. O tempo de CPU, para a execução do algoritmo principal do método CGSP, mais externo (*Passo 1*), foi dado em segundos, para todos os casos. Também foram analisados o número de iterações do algoritmo mais interno (*Passo 2*) do algoritmo CGSP. Todas as matrizes testadas nesta seção foram extraídas de Pardalos, Ye, Han & Kaliski (1993).

3.1 Matriz de Murty e Matriz de Fathi

A matriz de Murty (indicada com o sub índice M) e a matriz de Fathi (indicada com o sub índice F), quadradas de ordem n e o vetor $b \in \mathfrak{R}^n$ correspondente a cada um dos LCPs com as respectivas matrizes, são definidos por

$$A_M = \begin{bmatrix} 1 & & & & \\ 2 & 1 & & & 0 \\ 2 & 2 & 1 & & \\ \vdots & & \ddots & \ddots & \\ 2 & & \dots & 2 & 1 \end{bmatrix}, \quad A_F = \begin{bmatrix} 1 & 2 & 2 & \dots & 2 \\ 2 & 5 & 6 & \dots & 6 \\ 2 & 6 & 1 & & \\ \vdots & \vdots & & \ddots & \vdots \\ 2 & 6 & \dots & & 4(n-1)+1 \end{bmatrix}$$

e $b_M = b_F = (1, 1, \dots, 1)^t \in \mathfrak{R}^n$.

A solução do LCP(A,b) encontrada, para cada um dos problemas, é dada por $x_M^* = x_F^* = (1, 0, 0, \dots, 0)^t \in \mathfrak{R}^n$ e $y_M^* = y_F^* = (0, 1, 1, \dots, 1)^t \in \mathfrak{R}^n$.

Tabela 1 – Comparação entre os métodos RP implementados em diferentes compiladores, com $\beta = 0.99$

n	RP ₁ - Murty		RP ₂ - Murty	
	Iterações	Tempo	Iterações	Tempo
100	14.0	0.35	14.0	14.55
200	15.0	2.19	15.0	128.63
n	RP ₁ - Fathi		RP ₂ - Fathi	
	Iterações	Tempo	Iterações	Tempo
100	16.0	0.57	16.0	16.64
200	18.0	3.96	18.0	168.96

RP₁ – versão implementada em *IBM3090-600S*, utilizando o *VSFortran Compiler*, realizado pela referência Pardalos, Ye, Han & Kaliski (1993) e

RP₂ – versão implementada em *Pentium 133*, utilizando o *Fortran77*

Tabela 2 – Comparação entre o Método RP e o Método CGSP

n	RP – Murty ($\beta = 0.99$)		CGSP	
	Iterações	Tempo	Iterações	Tempo
100	14.0	14.55	6.0	3.52
200	15.0	128.63	7.0	6.97
n	RP – Fathi ($\beta = 0.99$)		CGSP	
	Iterações	Tempo	Iterações	Tempo
100	16.0	16.64	8.0	8.39
200	18.0	168.96	8.0	11.32

Tabela 3 – Número de iterações para o *Passo 2* (CGS) do algoritmo CGSP

n	Iterações para algoritmo CGS	
	Murty	Fathi
100	200, 200, 25, 16, 7, 1	200, 35, 17, 12, 5, 3, 2, 1
200	200, 74, 56, 15, 4, 2, 1	200, 56, 34, 18, 9, 4, 2, 1

onde a seqüência 200,200,25,...,1, para a matriz de Murty de ordem 100, na Tabela 3, significa que foram realizadas 200 iterações do algoritmo CGS, para a iteração 1 do método CGSP; 200 iterações do algoritmo CGS, para a iteração 2 do método CGSP; 25 iterações do algoritmo CGS, para a iteração 3 do método CGSP, e assim por diante.

3.2 Matriz Semi Positiva Definida Aleatória

Nesta seção vamos gerar LCP com matriz $A \in M_{n \times n}(\mathfrak{R})$ semi positiva definida, gerada a partir de números aleatórios. A matriz A tem a forma

$$A = U' \Lambda U$$

onde a matriz $U \in M_{n \times n}(\mathfrak{R})$ é a matriz de *Householder*, gerada a partir do vetor aleatório r , tal que $0 \leq r_i \leq 1$, $i = 1, \dots, n$. $\Lambda \in M_{n \times n}(\mathfrak{R})$ é matriz diagonal, representada por $\Lambda = \text{Diag}(\lambda_i)$, onde $0 \leq \lambda_i \leq 11$, $i = 1, \dots, n$. A solução de complementaridade (x^*, y^*) , é também gerada a partir de vetores aleatórios e que deverá satisfazer:

$$y^* > 0, (y^* + Ae)_i > 0, 0 \leq x_i^* \leq 10 \text{ e } (x^*)^t y^* = 0.$$

O vetor $b \in \mathfrak{R}^n$ será dado por $b = Ax^* - y^*$.

Tabela 4 – Comparação entre o Método RP e o Método CGSP

n	RP ($\beta = 0.99$)		CGSP	
	Iterações	Tempo	Iterações	Tempo
100	18.8	16.04	3.7	1.09
200	26.8	290.97	4.2	3.84

Tabela 5 – Número médio de iterações para o *Passo 2* (CGS) do algoritmo CGSP

Matriz Semi Positiva Definida Aleatória	
n	Iterações para algoritmo CGS
100	200, 24.6, 18.2, 0.6
200	200, 59.6, 21.8, 0.6

3.3 P-matrizes

Nesta seção os LCPs, a serem resolvidos, terão matriz $A \in M_{n \times n}(\mathfrak{R})$ pertencente à classe das P-matrizes com a seguinte estrutura:

$$A = \begin{bmatrix} P_1 & \alpha Q \\ 0 & P_2 \end{bmatrix}$$

onde $P_1 \in M_{n_1 \times n_1}(\mathfrak{R})$ e $P_2 \in M_{n_2 \times n_2}(\mathfrak{R})$ são matrizes positivas definidas, $Q \in M_{n_1 \times n_2}(\mathfrak{R})$, como definidas em Pardalos, Ye, Han & Kaliski (1993) e α é escalar positivo. Nos testes que realizamos, tomamos $n_1 = 2n/5$ e $n_2 = 3n/5$.

Os vetores que contém a solução de complementaridade (x^*, y^*) são gerados tais que:

$$0 \leq x_i^*, y_i^* \leq 10, \quad i = 1, \dots, n, \quad e \quad (x^*)^t y^* = 0$$

e o vetor $b \in \mathfrak{R}^n$ será dado por $b = Ax^* - y^*$.

3.3.1 Escalamento Matricial

Iremos utilizar escalamento matricial para podermos amenizar o problema de deterioração do número de condição do LCP, quando o parâmetro α cresce. Nos exemplos numéricos realizados, percebemos que este procedimento tornou mais viável a utilização do método RP. Isso concorda com a abordagem teórica mostrada por Pardalos, Ye, Han & Kaliski (1993). Um estudo mais detalhado sobre essas afirmações pode ser encontrado em Pissarra (1997).

O escalamento que se utiliza com frequência é o do tipo $\tilde{A} = EAD$ onde $E, D \in M_{n \times n}(\mathfrak{R})$ são matrizes diagonais positivas definidas. Dado este escalamento e um LCP(A,b), resolvemos um LCP(\tilde{A}, \tilde{b}) equivalente, onde $\tilde{b} = Eb$.

Encontrada a solução \tilde{x} de LCP(\tilde{A}, \tilde{b}), a solução equivalente para o LCP(A,b), é dada por $x = D\tilde{x}$ e $y = E^{-1}\tilde{y}$.

Para os problemas teste, adotamos uma versão do escalamento EAD, conhecido como escalamento balanceado, ver Pardalos, Ye, Han & Kaliski (1993). Seja

$$r(i) = \{ \max_j |A_{ij}|, \quad i = 1, \dots, n \} \quad e \quad \bar{A} = R^{-1}A$$

onde $R \in M_{n \times n}(\mathfrak{R})$ e $R = \text{Diag}(r)$,

$$c(j) = \{\max_i |\bar{A}_{ij}|, j = 1, \dots, n\} \quad e \quad \tilde{A} = R^{-1}AC^{-1}$$

onde $C \in M_{n \times n}(\mathcal{R})$, $C = \text{Diag}(c)$ e $\tilde{b} = R^{-1}b$.

Se $(\tilde{x}^*, \tilde{y}^*)$ é uma solução para o LCP(\tilde{A}, \tilde{b}), então $x^* = C^{-1}\tilde{x}^*$ é uma solução do LCP(A,b).

Tabela 6 – P-matriz (n = 100, $\beta = 0.99$)

Comparação entre escalamento e não escalamento no Método RP, para diferentes valores de α .

RP α	SEM ESCALAMENTO		COM ESCALAMENTO	
	Iter. média	Tempo	Iter. Média	Tempo
50	190.0	128.05	30.6	22.27
100	199.4	154.52	31.6	23.02

Tabela 7 – P-matriz (n = 100, n = 200)

Comparação entre o Método RP (com escalamento) e o Método CGSP, para diferentes valores de α .

n = 100 α	RP ($\beta = 0.99$)		CGSP	
	Iterações	Tempo	Iterações	Tempo
50	30.6	22.27	7.2	0.45
100	31.6	23.02	6.8	0.42
n = 200 α	RP ($\beta = 0.99$)		CGSP	
	Iterações	Tempo	Iterações	Tempo
50	38.8	362.78	7.0	0.99
100	41.6	381.14	7.2	1.02

Tabela 8 – Número médio de iterações para o *Passo 2* (CGS) do algoritmo CGSP

n	P-matriz
	Iterações para algoritmo CGS
100 ($\alpha=50$)	200, 80.6, 61.0, 51.0, 36.0, 11.6, 0.8
100 ($\alpha=100$)	200, 109.8, 61.4, 43.0, 35.4, 18.0, 0.8
200 ($\alpha=50$)	200, 96.6, 55.0, 39.6, 36.4, 20.6, 1.0
200 ($\alpha=100$)	200, 80.6, 67.2, 58.8, 46.2, 21.4, 3.2

3.4 P_0 - matrizes

Em várias aplicações de LCPs, a matriz A é uma P_0 - matriz. Nos exemplos teste, utilizaremos a matriz $A \in M_{n \times n}(\mathcal{R})$, dada por

$$A = \begin{bmatrix} P + D_1 & P + D_2 \\ -I_{n/2 \times n/2} & 0 \end{bmatrix}$$

onde $P \in M_{n/2 \times n/2}(\mathfrak{R})$, as matrizes diagonais $D_1, D_2 \in M_{n/2 \times n/2}(\mathfrak{R})$ todas positivas definidas.

Os vetores solução $x = (x_1 \ x_2)^t$ e $y = (y_1 \ y_2)^t$, onde $x_1, y_1, x_2, y_2 \in \mathfrak{R}^{n/2}$. Gerar $x_1^*, y_2^* \in \mathfrak{R}^{n/2}$ tal que $0 \leq (x_1^*)_i, (y_2^*)_i < 10$ e $(x_1^*)_i + (y_2^*)_i > 0$, onde $i = 1, \dots, n/2$. Tome y_1^* complementar a x_1^* e x_2^* complementar a y_2^* , tal que $0 \leq (x_2^*)_i, (y_1^*)_i < 10$, onde $i = 1, \dots, n/2$.

Tome $b = Ax^* - y^* = (b_1 \ b_2)^t$, onde $b_1, b_2 \in \mathfrak{R}^{n/2}$.

Tabela 9 – P_0 - matriz

Comparação entre o Método RP (com escalamento) e o Método CGSP, para diferentes dimensões

n	RP ($\beta = 0.99$)		CGSP	
	Iter. Média	Tempo	Iter. média	Tempo
100	246.4	256.38	11.3	2.46
200	762.5	7095.83	16.5	13.40

Tabela 10 – Número médio de iterações para o *Passo 2* (CGS) do algoritmo CGSP

n	P_0 - matriz									
	Iterações para algoritmo CGS									
100	200, 84.4, 74.6, 49.4, 56.8, 42.4, 38.0, 29.6, 22.2, 17.6, 4.4									
200	200, 121.0, 65.0, 71.8, 60.0, 46.4, 41.8, 67.0, 37.8, 48.4, 33.6, 31.2, 31.8, 32.8, 4.8, 2.0									

4. Conclusões

Apresentamos neste trabalho, a experiência computacional aplicada a diferentes problemas de complementaridade linear. As matrizes escolhidas, $A \in M_{n \times n}(\mathfrak{R})$, bem como os vetores $b \in \mathfrak{R}^n$ correspondentes a cada um dos LCPs, foram extraídos exemplos de Pardalos, Ye, Han & Kaliski (1993) e estão subdivididos em classes: (i) matrizes semi positivas definidas (*semi pd*), incluindo os exemplos de Murty e Fathi, matrizes spd geradas aleatoriamente e P-matrizes e (ii) P_0 - matrizes.

O que fizemos neste trabalho foi a aplicação da estratégia desenvolvida com sucesso em Pissarra (1997), para resolver LCPs de matrizes não singulares. Chamamos este novo método de Gradientes Conjugados Quadrático com Projeções (CGSP). Comparamos o desenvolvimento deste método com o método de Redução Potencial (RP), desenvolvido em Pardalos, Ye, Han & Kaliski (1993). Com isto, podemos fazer as seguintes observações, com base na experiência numérica mostrada aqui: analisando a aplicação do método RP, para cada uma dessas classes de matrizes, podemos ver que para os exemplos das matrizes de Murty e Fathi e para o exemplo da matriz semi positiva definida gerada aleatoriamente, o método se mostrou eficiente quanto ao número de iterações, que permaneceu praticamente inalterado, quando houve um aumento na dimensão dos problemas. Isto confirma o resultado teórico mostrado em Pissarra (1997). Outro fator importante é a escolha correta do tamanho do passo (β) para todos os exemplos, que em testes confirmaram o valor $\beta = 0.99$. Estes resultados são condizentes com a referência Pardalos, Ye, Han & Kaliski (1993) na qual tomamos como base.

Já para os exemplos de LCPs com P-matrizes e P_0 - matrizes os resultados obtidos não foram tão satisfatórios, em termos práticos, quando utilizado o método RP. Isto se deve ao fato de, como visto pelo desenvolvimento teórico, para um aumento do parâmetro α para P-matriz, o número de condição do LCP deteriora, a convergência do algoritmo é mais lenta, levando um maior número de iterações para convergir, e por conseguinte a um tempo de execução maior. No entanto, a técnica de escalamento utilizada ajudou muito no balanceamento da matriz, melhorando assim a velocidade de convergência, como mostra a tabela 6. Para os LCPs com P_0 - matrizes, no entanto, percebemos que o método RP está longe de ser uma técnica adequada para sua resolução. O algoritmo convergiu, mas o tempo gasto e o número de iterações realizadas não nos permite utilizar o método para resolvê-los com eficiência, como mostra a tabela 9. Estas conclusões são condizentes com Pardalos, Ye, Han & Kaliski (1993).

Para as classes de LCPs em que aplicamos o método RP, aplicamos também o método CGSP e analisando os resultados obtidos nas tabelas 2, 4, 7 e 9, este método obteve melhores resultados, tanto no número de iterações, quanto no tempo gasto para a realização. Outra observação importante é o número de iterações não se modificar significativamente, quando aumentamos a dimensão do problema. Este método também se mostrou muito eficiente por não necessitar de um conjunto ativo inicial ótimo pré estabelecido, o que facilita ainda mais a resolução de problemas. Conclui-se que o método CGSP obteve resultados superiores aos do método RP, conclusão esta obtida através de análises dos experimentos numéricos para a classe de matrizes desenvolvida neste trabalho.

Referências Bibliográficas

- (1) Cottle, R.W.; Pang, J.S. & Stone, R.E. (1992). *The Linear Complementarity Problem*. Academic Press, Inc.
- (2) Fernandes, M.R. (1993). Métodos Variacionais para Sistemas Lineares Esparsos: Uma Aplicação à Superfícies Livres de Capilaridade. Tese de Mestrado, UNICAMP.
- (3) Golub, G.H. & Van Loan, C. (1985). *Matrix Computations*. The Johns Hopkins University Press.
- (4) Greenbaum, A. (1997). *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia.
- (5) Kocvara, M. & Zowe, J. (1994). An Iterative two-step Algorithm for Linear Complementarity Problems. *Numerische Mathematik*, **68**(01), 95-106.
- (6) Pardalos, P.M.; Ye, Y.; Han, C.G. & Kaliski (1993). Solution of P_0 -Matrix Linear Complementarity Problems Using a Potential Reduction Algorithm. *SIAM Journal on Matrix Analysis and Applications*, **14**(04), 1048-1060.
- (7) Pissarra, C.M.A. (1997). Problemas de Complementaridade Linear: Aspectos Teóricos, Computacionais e Aplicações. Tese de Mestrado, UNICAMP.