

GRADIENTE CONJUGADO PRECONDICIONADO PELA MATRIZ QUASE-NEWTON DE MEMÓRIA LIMITADA

Veranise J. C. Dubeux, Paulo C Mappa, Alfredo Canelas e José Herskovits

Programa de Engenharia Mecânica, COPPE,
Universidade Federal do Rio de Janeiro.
Caixa Postal 68503, 21945 970, Rio de Janeiro - RJ - Brasil
e-mail: veranise@optimize.ufrj.br

***Palavras Chaves:** Programação Não Linear, Algoritmos de Pontos Interiores, Método do Gradiente Conjugado Precondicionado, Técnica Quase-Newton de Memória Limitada.*

***Resumo.** Desenvolvemos uma técnica iterativa e eficiente para a resolução de sistemas lineares grandes e mal condicionados, pensando em sua aplicação em sistemas lineares gerados pelos algoritmos de ponto interior utilizados na programação não linear. As idéias apresentadas estão baseadas no método do Gradiente Conjugado e na técnica de condicionamento quase-Newton de Memória Limitada. A técnica apresentada é integrada ao programa de otimização FAIPA, Feasible Arc Interior Point Algorithm.*

1 INTRODUÇÃO

As técnicas iterativas para a resolução de sistemas lineares grandes e mal condicionados foram desenvolvidas pensando principalmente na sua aplicação nos sistemas lineares que devem ser resolvidos nos algoritmos de ponto interior utilizados na Programação Não Linear.

A solução de problemas de grande porte implica, em particular, na solução de sistemas lineares internos ao FAIPA, com elevado número de equações. Para tal problema é feita uma integração de técnicas de memória limitada a métodos iterativos, como por exemplo, o método do gradiente conjugado.

Nos problemas de otimização da engenharia o custo computacional da sua resolução está fundamentalmente no cálculo das funções associadas, suas derivadas, e também nos sistemas lineares que os algoritmos de Pontos Interiores utilizados devem resolver em cada iteração. Outro aspecto crítico quando o problema é grande é o armazenamento das matrizes e os vetores utilizados.

Várias características dos sistemas que devem ser resolvidos motivam o estudo de métodos iterativos em vez de métodos diretos, em primeiro lugar a matriz quase-Newton que compõe a matriz do sistema é cheia, seu armazenamento requer muita memória. Por outro lado o Método de Memória Limitada permite armazenar informação suficiente para representar a matriz sem ter que armazenar todos os seus coeficientes. Os métodos iterativos não necessitam o conhecimento explícito dos coeficientes da matriz e portanto se apresentam muito convenientes para serem utilizados juntamente com estas técnicas de Memória Limitada. Desta forma pode ser reduzida substancialmente a capacidade necessária para o armazenamento. Em segundo lugar é possível transformar o sistema original num outro sistema simétrico definido positivo o que permite a utilização do método do Gradiente Conjugado Precondicionado, método que para este tipo de problemas se apresenta muito competitivo quando comparado com os métodos diretos.

Em problemas de grande porte os sistemas lineares que devem ser resolvidos são freqüentemente muito mal condicionados, foi portanto necessário utilizar técnicas de precondicionamento para acelerar a convergência do algoritmo. Foi desenvolvida uma técnica de precondicionamento baseada na técnica quase-Newton de Memória Limitada introduzida por Nocedal e Morales¹ esta técnica tem como principal vantagem que pode ser utilizada sem o conhecimento explícito da matriz de coeficientes do sistema original pois utiliza informação obtida durante a resolução de sistemas lineares anteriores. Além disso, ela gera o precondicionador diretamente na forma compacta de Memória Limitada. Outros precondicionadores podem ser utilizados em conjunto a técnica apresentada em Morales, neste trabalho foi utilizado o precondicionador de Jacobi, outros métodos estão sendo implementados e serão apresentados em futuros trabalhos.

As técnicas desenvolvidas foram integradas ao programa FAIPA (Feasible Arc Interior Point Algorithm). FAIPA é um algoritmo de Pontos Interiores que resolve o problema geral de otimização não-linear com restrições, ele tem sido muito utilizado nos últimos anos e tem se mostrado confiável na resolução de problemas de otimização em diversas áreas da Engenharia^{2,3}.

Alguns exemplos de problemas de grande porte serão resolvidos com a técnica proposta neste trabalho.

2 PROBLEMA DE PROGRAMAÇÃO NÃO LINEAR

O problema geral de programação não linear considerado é representado da seguinte maneira:

$$\left. \begin{array}{l} \underset{x}{\text{minimize}} \quad f(x), \quad x \in \mathfrak{R}^n \\ \text{sujeito a} \quad g_i(x) \leq 0; \quad i = 1, \dots, m \\ \text{e} \quad \quad \quad h_i(x) = 0; \quad i = 1, \dots, p \end{array} \right\} \quad (1)$$

Onde: x é o vetor das variáveis do projeto, $f(x): \mathfrak{R}^n \rightarrow \mathfrak{R}$ é a função objetivo, $g(x): \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ é a função que representa as restrições de desigualdade e $h(x): \mathfrak{R}^n \rightarrow \mathfrak{R}^p$ é a função que representa as restrições de igualdade.

As condições de Karush-Kuhn-Tucker, correspondentes ao problema Eq. (1), são as seguintes:

$$\left\{ \begin{array}{l} \nabla f(x) + \nabla g(x)\lambda + \nabla h(x)\mu = 0 \\ \lambda^t g(x) = 0 \\ \lambda \geq 0 \\ g(x) \leq 0 \\ h(x) = 0, \end{array} \right. \quad (2)$$

onde $\lambda \in R^m$ e $\mu \in R^p$ são os multiplicadores de Lagrange que correspondem as restrições de desigualdade e igualdade respectivamente.

2.1 Algoritmo FAIPA

O FAIPA (Feasible Arc Interior Point Algorithm), proposto por Herskovits² é um algoritmo de Pontos Interiores que resolve o problema geral de otimização não-linear dado pela Eq. (1). Ele faz iterações nas variáveis de projeto (variáveis primais) e nos multiplicadores de Lagrange (variáveis duais) até verificar as condições de otimalidade de Karush-Kuhn-Tucker.

O FAIPA parte de um ponto inicial interior à região viável, isto é, interior à região definida pelo conjunto de pontos que verificam as restrições de desigualdade. Logo ele define uma seqüência de pontos dentro desta região. Em cada um destes pontos é feita uma busca linear inexata ao longo de um arco. Para calcular o arco, o FAIPA resolve três sistemas lineares com uma mesma matriz de coeficientes. Os sistemas que são resolvidos em cada iteração do FAIPA e que permitem definir o arco de busca são os seguintes:

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \Lambda \nabla g^t & G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} d_0 \\ \lambda_0 \\ \mu_0 \end{bmatrix} = - \begin{bmatrix} \nabla f \\ 0 \\ h \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \Lambda \nabla g^t & G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ \lambda_1 \\ \mu_1 \end{bmatrix} = - \begin{bmatrix} 0 \\ \lambda \\ \mu \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \Lambda \nabla g^t & G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{d} \\ \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = - \begin{bmatrix} 0 \\ \lambda \tilde{w}^I \\ \mu \tilde{w}^E \end{bmatrix} \quad (5)$$

Os sistemas das Eq. (3) a (5) têm a mesma matriz de coeficientes e têm sempre solução⁴. O uso da técnica da representação por Memória Limitada da matriz Quase-Newton no algoritmo de Pontos Interiores e Arcos Viáveis (FAIPA) possibilitará um ganho computacional, principalmente para problemas de grande porte, pois não será necessário guardar a matriz quase-Newton, B, a cada iteração.

Em forma compacta pode-se expressar Eqs. (3) a (5) pela seguinte equação equivalente:

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \Lambda \nabla g^t & G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} d_0 & d_1 & \tilde{d} \\ \lambda_0 & \lambda_1 & \tilde{\lambda} \\ \mu_0 & \mu_1 & \tilde{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla f & 0 & 0 \\ 0 & \lambda & \lambda \tilde{w}^I \\ h & \mu & \mu \tilde{w}^E \end{bmatrix} \quad (6)$$

A matriz destes sistemas não é simétrica e nem definida positiva, portanto não pode ser aplicado o Método do Gradiente Conjugado. Se multiplicarmos a segunda equação deste sistema pela matriz diagonal Λ^{-1} (de elementos sempre positivos) se obtém o seguinte sistema com a matriz de coeficientes simétrica:

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \nabla g^t & \Lambda^{-1}G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} d_0 & d_1 & \tilde{d} \\ \lambda_0 & \lambda_1 & \tilde{\lambda} \\ \mu_0 & \mu_1 & \tilde{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla f & 0 & 0 \\ 0 & E & \tilde{w}^I \\ h & \mu & \mu \tilde{w}^E \end{bmatrix} \quad (7)$$

Onde E é um vetor com cada componente igual a 1. A matriz do sistema dado pela Eq. (7) continua sendo não definida. Da primeira equação pode-se obter as variáveis d em função das outras variáveis:

$$\begin{bmatrix} B & \nabla g & \nabla h \\ \nabla g^t & \Lambda^{-1}G & 0 \\ \nabla h^t & 0 & 0 \end{bmatrix} \begin{bmatrix} d_0 & d_1 & \tilde{d} \\ \lambda_0 & \lambda_1 & \tilde{\lambda} \\ \mu_0 & \mu_1 & \tilde{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla f & 0 & 0 \\ 0 & E & \tilde{w}^t \\ h & \mu & \lambda \tilde{w}^E \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} d_0 \\ d_1 \\ \tilde{d} \end{bmatrix} = - \begin{bmatrix} B^{-1}(\nabla f + \nabla g \lambda_0 + \nabla h \mu_0) \\ B^{-1}(\nabla g \lambda_1 + \nabla h \mu_1) \\ B^{-1}(\nabla g \tilde{\lambda} + \nabla h \tilde{\mu}) \end{bmatrix} \quad (8)$$

Um sistema com matriz simétrica e definida positiva pode ser obtido substituindo a Eq. (8) na Eq. (7) para de esta forma obter o seguinte sistema linear:

$$\begin{bmatrix} \nabla g^t B^{-1} \nabla g - \Lambda^{-1} G & \nabla g^t B^{-1} \nabla h \\ \nabla h^t B^{-1} \nabla g & \nabla h^t B^{-1} \nabla h \end{bmatrix} \begin{bmatrix} \lambda_0 & \lambda_1 & \tilde{\lambda} \\ \mu_0 & \mu_1 & \tilde{\mu} \end{bmatrix} = \begin{bmatrix} -\nabla g^t B^{-1} \nabla f & E & \tilde{w}_t \\ -\nabla g^t B^{-1} \nabla f + h & \Lambda^{-1} \mu & \Lambda^{-1} \mu \tilde{w}_E \end{bmatrix} \quad (9)$$

Após resolver este sistema pode-se achar as variáveis d utilizando a Eq. (8). Nos casos onde na solução do problema alguns multiplicadores de Lagrange, λ_i , são muito pequenos é de se esperar que o número de condicionamento da matriz simétrica do sistema da Eq. (9) aumente muito na últimas iterações do algoritmo FAIPA, dado que $G\Lambda^{-1}$ aparece na diagonal desta matriz. O número de condicionamento pode também crescer muito quando a matriz Hessiana na solução apresenta autovalores menores ou iguais a zero, fazendo que a matriz B apresente autovalores muito pequenos. Nestes dois casos, em teoria, o número de condicionamento da matriz do sistema tende a infinito. Uma das soluções seria impor uma cota inferior aos valores de λ_i e aos autovalores de B . Utilizar cotas grandes prejudica as propriedades de convergência do algoritmo FAIPA. Optou-se então por utilizar cotas pequenas e preconditionar a matriz para melhorar o número de condicionamento antes de aplicar o método iterativo para resolver o sistema.

3 MÉTODO DO GRADIENTE CONJUGADO PRECONDICIONADO

O método do Gradiente Conjugado é uma técnica iterativa para resolução de sistemas de equações lineares com a matriz de coeficientes simétrica definida positiva.

$$Ax = b \quad (10)$$

A velocidade de convergência do Método do Gradiente Conjugado assim como outros métodos iterativos depende de características da matriz de coeficientes do sistema. Os métodos iterativos com preconditionamento resolvem o seguinte sistema:

$$M^{-1}Ax = M^{-1}b \quad (11)$$

Onde M é denominada preconditionador da matriz de coeficientes. O método iterativo fará um menor número de iterações quanto mais próximo de 1 estiverem os autovalores da matriz de coeficientes resultante $M^{-1}A$. Por exemplo, se na Eq. (11) a matriz M for a própria matriz A o sistema resultante fica com a matriz de coeficientes igual a matriz identidade. De maneira geral, não pode ser tomada M igual a matriz A . M deve ser uma matriz próxima de A e será um melhor preconditionador quanto mais próximo da identidade estiver a matriz $M^{-1}A$.

Abaixo é apresentado o algoritmo do Método do Gradiente Conjugado Precondicionado para a solução de um sistema linear de equações⁵. Dada uma estimativa inicial de x_0 :

Calcule:

$$r_0 = b - Ax_0 \quad (12)$$

$$z_0 = M^{-1}r_0 \quad (13)$$

$$p_0 = z_0 \quad (14)$$

Para i variando de 1 até verificar o critério de convergência adotado faça:

$$\alpha_i = r_i^t z_i / (Ap_i)^t p_i \quad (15)$$

$$x_{i+1} = x_i + \alpha_i p_i \quad (16)$$

$$r_{i+1} = r_i - \alpha_i Ap_i \quad (17)$$

$$z_{i+1} = M^{-1}r_{i+1} \quad (18)$$

$$\beta_i = r_{i+1}^t z_{i+1} / r_i^t z_i \quad (19)$$

$$p_{i+1} = z_{i+1} + \beta_i p_i \quad (20)$$

fim

Além de M preconditionar bem o sistema original, resolver um sistema linear com esta matriz deve ter um custo computacional muito menor do que com a matriz original A pois isto deve ser feito em cada iteração (Eq. 18).

De maneira similar pode-se resolver o seguinte sistema:

$$MAx = Mb \quad (21)$$

Neste caso, a matriz M deve-se aproximar à matriz inversa de A . Neste caso, evita-se resolver um sistema linear em cada iteração.

4 TÉCNICA DE MEMÓRIA LIMITADA

A técnica de memória limitada foi concebida para resolução de problemas de otimização

não linear de grande porte. A mesma, que é baseada no método quase-Newton⁶ permite aproximar a inversa da matriz Hessiana da função que se deseja minimizar sem a necessidade de armazenar a matriz quase-Newton. Morales e Nocedal¹ a apresentaram para preconditionamento de sistemas lineares, eles aplicaram a técnica de Memória Limitada a seguinte função:

$$f(x) = x^t Ax - x^t b \quad (22)$$

Como a Hessiana desta função é matriz A do sistema linear, a matriz quase-Newton servirá como preconditionador.

A técnica de Memória Limitada tem as seguintes vantagens: ela gera um preconditionador que aproxima diretamente a inversa da matriz do sistema linear, este preconditionador é obtido na forma compacta de memória limitada o que viabiliza seu uso para problemas grandes, outra vantagem é que não necessita do conhecimento explícito dos coeficientes da matriz do sistema linear. Esta técnica pode ser empregada quando devem ser resolvidos vários sistemas lineares com a mesma matriz de coeficientes ou com matriz de coeficientes que variam ligeiramente. Este é o caso dos sistemas lineares dos algoritmos para programação não linear. Durante a resolução do primeiro sistema linear, que deve ser feita pelo método do Gradiente Conjugado, é construído o preconditionador que só poderá ser utilizado na resolução dos seguintes sistemas lineares.

Seja H_0 uma matriz simétrica definida positiva. Sejam os pares $\{s_i, y_i\}_{i=1}^{k-1}$, onde s_i são algumas das direções conjugadas obtidas pelo algoritmo do Gradiente Conjugado enquanto os vetores y_i cumprem:

$$y_i = As_i \quad (23)$$

A matriz de preconditionamento gerada pelo método BFGS, é a seguinte:

$$H_k = H_0 + \begin{bmatrix} S_k & H_0 Y_k \end{bmatrix} \begin{bmatrix} R_k^{-t} (D_k + Y_k^t H_0 Y_k) R_k^{-1} & -R_k^{-t} \\ -R_k^{-1} & 0 \end{bmatrix} \begin{bmatrix} S_k^t \\ Y_k^t H_0 \end{bmatrix} \quad (24)$$

onde:

$$\begin{aligned} S_k &= [s_0, \dots, s_k] \\ Y_k &= [y_0, \dots, y_k] \end{aligned} \quad (25)$$

são matrizes ($n \times k$);

$$(R_k)_{i,j} = \begin{cases} s_i^t y_j & \text{se } i \leq j \\ 0 & \text{caso contrário} \end{cases} \quad (26)$$

é uma matriz triangular ($k \times k$);

$$D_k = \text{diag}[s_0^t y_0, \dots, s_{k-1}^t y_{k-1}] \quad (27)$$

é uma matriz diagonal ($k \times k$).

Quando H_0 é um múltiplo da matriz identidade, ou seja $H_0 = \gamma_k I$ se obtém:

$$H_k = \gamma_k + \begin{bmatrix} S_k & \gamma_k Y_k \end{bmatrix} \begin{bmatrix} R_k^{-t} (D_k + \gamma_k Y_k^t Y_k) R_k^{-1} & -R_k^{-t} \\ -R_k^{-1} & 0 \end{bmatrix} \begin{bmatrix} S_k^t \\ \gamma_k Y_k^t \end{bmatrix} \quad (28)$$

Os coeficientes da matriz H_k não são conhecidos, porém pode ser feito o produto desta matriz por um vetor na seguinte forma:

$$H_k v = \gamma_k v + \begin{bmatrix} S_k & \gamma_k Y_k \end{bmatrix} \begin{bmatrix} R_k^{-t} (D_k + \gamma_k Y_k^t Y_k) R_k^{-1} & -R_k^{-t} \\ -R_k^{-1} & 0 \end{bmatrix} \begin{bmatrix} S_k^t v \\ \gamma_k Y_k^t v \end{bmatrix}. \quad (29)$$

5 GRADIENTE CONJUGADO PRECONDICIONADO PELA MATRIZ QUASE-NEWTON DE MEMÓRIA LIMITADA

O algoritmo apresentado neste trabalho foi pensado para resolver o seguinte sistema:

$$HL^{-1}AL^{-t}y = HL^{-1}b \quad (30)$$

onde H é o preconditionador baseado na técnica quase-Newton de Memória Limitada. As matrizes L e L^t são matrizes de condicionamento triangulares que podem ser obtidas por algum outro método de condicionamento como, por exemplo, o método de decomposição incompleta de Cholesky da matriz A . Aplicando o algoritmo dado pelas Eqs. (12) a (20) resulta o seguinte algoritmo:

Calcule

$$r_0 = L^{-1}(b - Ax_0) \quad (31)$$

$$z_0 = Hr_0 \quad (32)$$

$$p_0 = L^{-t}z_0 \quad (33)$$

Para i variando de 1 até verificar o critério de convergência adotado faça:

$$\alpha_i = r_i^t z_i / (Ap_i)^t p_i \quad (34)$$

$$x_{i+1} = x_i + \alpha_i p_i \quad (35)$$

$$r_{i+1} = r_i - \alpha_i L^{-1} Ap_i \quad (36)$$

$$z_{i+1} = Hr_{i+1} \quad (37)$$

$$\beta_i = r_{i+1}^t z_{i+1} / r_i^t z_i \quad (38)$$

$$p_{i+1} = L^{-t} z_{i+1} + \beta_i p_i \quad (39)$$

fim

A matriz H representa a matriz quase-Newton de Memória Limitada que foi proposta em Morales¹, ela não é calculada explicitamente, sua representação compacta de memória limitada é utilizada no algoritmo pra calcular o produto da Eq. (37). Neste trabalho se propõe um método diferente ao proposto em Morales¹ para a escolha dos pares $\{s_i; y_i\}$ que serão empregados para construir o preconditionador H .

Ao se testar o algoritmo proposto percebeu-se que a escolha dos pares $\{s_i; y_i\}$ no processo de memória limitada afeta significativamente a eficácia da matriz de Memória Limitada no preconditionamento do problema. Basicamente a matriz H é igual à proposta por Morales¹ com a diferença na escolha dos pares $\{s_i; y_i\}$ utilizados.

Para que seja feita a escolha dos pares primeiro é feito o seguinte cálculo para cada par $\{s_i; y_i\}$:

$$\alpha_i = s_i^t A s_i / s_i^t s_i \quad (40)$$

É importante observar que o valor de α_i é máximo se a direção s_i coincide com o autovetor da matriz A de autovalor máximo, enquanto é mínimo se coincide com o autovetor de autovalor mínimo. Um dos efeitos da matriz H de memória limitada é corrigir este valor nas direções escolhidas para construí-la.

Cada vez que deve ser descartado um par escolhe-se aquele par $\{s_i; y_i\}$ cujo valor de α_i esteja mais próximo de 1. Isto é feito com o propósito de manter os pares cujo valor de α_i é muito grande ou muito pequeno, tentando corrigir os valores dos α_i nas direções responsáveis pelo mal condicionamento da matriz A .

A escolha proposta dos pares mostrou as seguintes vantagens:

Para um número de pares máximo igual, o número de iterações necessário para a convergência diminui; na medida em que aumentou-se o número máximo de pares $\{s_i; y_i\}$ utilizados o número de iterações diminui, fato que não foi obtido em Morales¹.

6 RESULTADOS NUMÉRICOS

Os resultados de problemas de Programação não Linear com restrições resolvidos com o programa FAIPA implementado com os métodos: Gradiente Conjugado Precondicionado (GCP) e Gradiente Conjugado Precondicionado pela matriz quase-Newton de Memória Limitada (GCP_ML) são apresentados na Tabela 1.

Tabela 1- - Resultados Numéricos

Nome do Problema	Tamanho da matriz do sistema	GCP	GCP_ML (8 pares)	GCP_ML (16 pares)	GCP_ML (32 pares)
Polígono 25	420	87.8	75.7	64.6	48.8
			66.4	57.7	45.2
Polígono 50	1470	139.6	134.6	128,1	117.4
			130.2	103.3	111.8
Polígono 75	3145	152.6	146.0	139.2	126.3
			138.5	128.3	115.2
Polígono 100	5445	178.5	175.1	170.2	164
			169.5	160.8	154.8
Can Shape 100	504	544.1	578.9	540.4	490.0
			437.7	460.7	285.4
Can Shape 200	1004	929.5	918.8	912.3	902.3
			912.3	840.5	814.5
Minsufo 25	1250	7.7	6.4	6.1	6.1
			6.4	6.1	6.1
Elec 25	25	4.0	4.2	4.0	4.0
			4.1	4.0	4.0
Chain 50	51	51.4	45.5	37.7	21.4
			45.5	37.6	21.4
Chain 100	101	101.2	95.2	87.6	71.2
			95.2	87	70.8
Chain 200	201	201.1	197.1	187.3	170.7
			195.4	187	170

Os problemas foram selecionados da coleção de problemas do CUTer^{7,8} (*Constrained and Unconstrained Testing Environment revisited*) e foram rodados em FORTRAN, versão 6.1, num AMD Athlon XP de 2.5 GHz, 1.5 GB de memória RAM e HD de 80 GB.

Os valores apresentados na Tabela 1 referem-se ao número médio de iterações do método do Gradiente Conjugado para resolver o segundo sistema do FAIPA. Para o método GCP_ML são apresentados dois resultados para um número máximo de pares igual a 8, 16 e 32. O primeiro resultado refere-se ao GCP_ML com a escolha dos pares e o segundo ao GCP_ML sem a escolha de pares.

O preconditionador utilizado tanto nos método GCP como no Método GCP_ML foi o preconditionador de Jacobi.

A matriz de preconditionamento de memória limitada foi achada em cada iteração durante a resolução do primeiro sistema do FAIPA e utilizada na resolução do segundo sistema. Por esta razão se apresentam resultados somente para o segundo sistema. Não se obteve bons resultados ao utilizar a matriz de preconditionamento na resolução dos sistemas da seguinte iteração do FAIPA. Acredita-se que isto foi devido à utilização do sistema dual cuja matriz de coeficientes varia muito em cada iteração.

A Figura 1 ilustra uma comparação do número médio de iterações, para resolver o segundo sistema do FAIPA, entre os dois métodos CGP e GCP_ML.

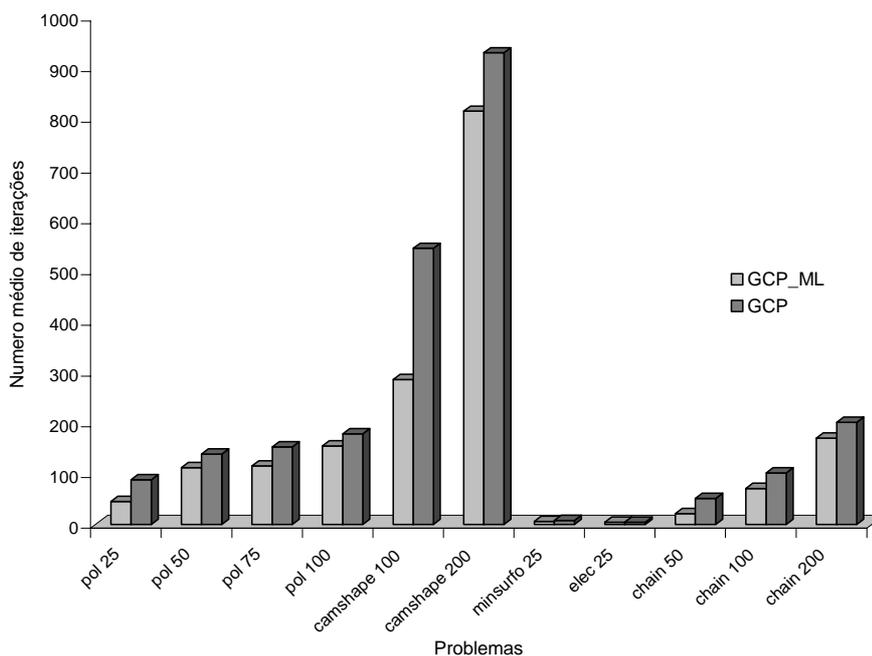


Figura 1 - Número médio de iterações para resolver o sistema dual com os métodos: GCP_ML e GCP

Na Figura 2 é possível visualizar o ganho percentual em número de iterações no segundo sistema do FAIPA do método GCP_ML em GCP.

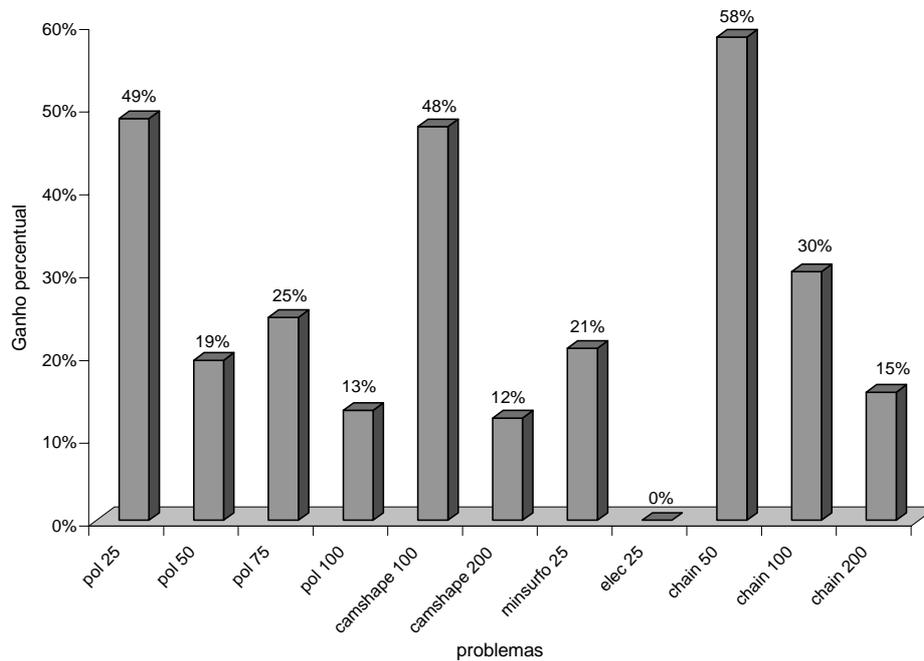


Figura 2 - Ganho percentual do número médio de iterações do GCP_ML em relação GCP

Os resultados apresentados na Tabela 2 mostram que o custo computacional do algoritmo que monta a matriz do sistema dual a cada iteração.

Tabela 2 - Tempo necessário para resolver o problema Polígono (25, 50, 75 e 100) utilizando GCP_ML (32 pares).

Nome do Problema	Tamanho da matriz do sistema	Sem montar a matriz do sistema dual	Montando a matriz do sistema dual
Polígono 25	420	15.26 seg.	98.61 seg.
Polígono 50	1470	89.75 seg.	4114.83 seg.
Polígono 75	3145	422.45 seg.	7311.47 seg.
Polígono 100	5445	840.89 seg.	14611.47seg

7 CONCLUSÕES

O aumento do número máximo de pares mostrou uma redução do número de iterações do GCP_ML na maioria dos casos. Porém em alguns casos se obteve tempos de execução maiores. No futuro será feito um estudo para determinar o número de pares mais adequado em função das características do problema;

Em muitos problemas o número médio de iterações do GCP foi muito alto. Em alguns casos se atingiu o número máximo permitido. Nestes casos, o condicionamento da matriz dos sistemas mostrou-se insuficiente. Em parte, isto pode ser solucionado utilizando um condicionador melhor que o de Jacobi;

No conjunto de problemas resolvidos somente os problemas Minsufo e Elec foram resolvidos de forma satisfatória. Os outros problemas apresentaram um número médio de iterações grande. Acredita-se que a principal causa deste fato seja a resolução do problema por meio do sistema dual;

No futuro serão estudados métodos iterativos para a resolução do sistema primal-dual (Eq. 6). Acredita-se que seja possível obter melhores resultados que os apresentados neste trabalho.

A escolha dos pares apresentada mostrou melhoras em alguns problemas. Em cálculos realizados com matrizes aleatórias bem e mal condicionadas, foram obtidos resultados promissores. Acredita-se que essa é uma técnica que pode ser aplicada com mais sucesso no sistema primal-dual para algoritmos iterativos que possam ser aplicados em sistemas não definidos

A Tabela 2, mostra o bom desempenho da técnica proposta em termos de tempo.

8 REFERÊNCIAS

- [1] Morales, J. L., Nocedal, J., Automatic preconditioning by limited memory quasi-Newton updating, *SIAM Journal on Optimization*, 10(4): 1079-1096, June, 2000
- [2] Herskovits, J., A View on Nonlinear Optimization. Cap. Advances in Structural Optimization, I. Herskovits Ed., - KLUWER Academic publishers, Holland, p. 7-116, June, 1995.
- [3] Herskovits, J., A Feasible Directions Interior Point Technique for Nonlinear Optimization. *JOTA – Journal of Optimization Theory and Applications*, Vol. 99, N1, p. 121-146, October, 1998.
- [4] Herskovits, J., Santos, G., Feasible Arc Interior Point Algorithms for Nonlinear Optimization. *Fourth World Congress on Computational Mechanics*, in CD-ROM, Buenos Aires, Argentina, June, 1998
- [5] Saad, Y., Iterative Methods for Sparse Linear Systems, 2a ed., Siam, 2003.
- [6] Byrd, R. H., Nocedal, J., and Schnabel, R. B., Representations of Quasi-Newton Matrices and Their use in Limited Memory Methods, *Mathematical Programming* 63, 129-156, 1994.
- [7] Dolan, E. D., Moré, J. J., Benchmarking Optimization Software with COPS, Mathematics and Computer Science Division, Technical Report ANL/MCS-246, Argonne National Laboratory, Argonne, Illinois, November, 2000.
- [8] Gould, N. I. M., Orban, D. and Toint, P. L., General CUTER and Sif Dec Documentation, England, 2002, <http://hls.rl.ac.uk/cuter-www>.