UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas – ICEx
Programa de Pós-Graduação em Matemática (PGMAT)

Lucas Almeida Portela

O Método do Gradiente Conjugado de Hestenes e Stiefel Modificado para Otimização Não-Linear em Alta Dimensão

Belo Horizonte

2022

Lucas Almeida Portela

O Método do Gradiente Conjugado de Hestenes e Stiefel Modificado para Otimização Não-Linear em Alta Dimensão

Dissertação apresentada ao Programa de Pós-Graduação em Matemática do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Matemática.

Orientador: Prof. Dr Ricardo Hiroshi Caldeira Takahashi (ICEX-UFMG)
Coorientador: Prof. Dr Luis Carlos de Castro Santos (IME-USP)

Belo Horizonte

2022

Portela, Lucas Almeida.

P843m O Método do gradiente conjugado de Hestenes e Stiefel modificado para otimização não-linear em alta dimensão [recurso eletrônico] / Lucas Almeida Portela – 2022.
81 f. il.

Orientador: Ricardo Hiroshi Caldeira Takahashi.
Coorientador: Luís Carlos de Castro Santos.
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Matemática.
Referências: f.69-71.

1. Matemática – Teses. 2. Otimização matemática – Teses. 3. Métodos do gradiente conjugado – Teses. I. Takahashi, Ricardo Hiroshi Caldeira. II. Santos, Luís Carlos de Castro. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Matemática. IV. Título.

CDU 51(043)

Ficha catalográfica elaborada pela bibliotecária Irenquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais – ICEx

Universidade Federal de Minas Gerais
Departamento de Matemática
Programa de Pós-Graduação em Matemática

FOLHA DE APROVAÇÃO

O Método do Gradiente Conjugado de Hestenes e Stiefel Modificado para Otimização Não-Linear em Alta Dimensão

LUCAS ALMEIDA PORTELA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Prof. Ricardo Hiroshi Caldeira Takahashi
UFMG

Prof. Luís Carlos de Castro Santos
IME/USP e Embraer

Profa. Denise Bulgarelli Duczmal
UFMG

Prof. Nelson Mugayar Kuhl
IME/USP

Belo Horizonte, 28 de julho de 2022.

Av. Antônio Carlos, 6627 – Campus Pampulha - Caixa Postal: 702
CEP-31270-901 - Belo Horizonte – Minas Gerais - Fone (31) 3409-5963
e-mail: pgmat@mat.ufmg.br - home page: http://www.mat.ufmg.br/pgmat

Agradecimentos

Primeiramente, gostaria de agradecer a Deus, por essa nova conquista. Pois, é Ele a minha fonte e sempre será. Posteriormente, os agradecimentos são direcionados ao orientador, professor Ricardo Hiroshi Caldeira Takahashi e ao coorientador, professor Luis Carlos de Castro Santos. Tenho muita sorte e privilégio de tê-los como mestres e referências.

Seguidamente, à minha mãe, Sandra, à irmã, Sarah e à namorada (noiva), Jordana, pelo tempo cedido a mim. À Jordana, meus sinceros agradecimentos pelo tempo dispendido para correção ortográfica desta dissertação, longas conversas, companheirismo, etc.

Não poderia me esquecer de todos os professores dos quais tive a honra de ser aluno: prof. Bernardo Nunes Borges Lima (Probabilidade I); prof. Marcelo Hilário (Probabilidade II); prof. Henrique Versieux (Equações Diferenciais Parciais I); prof. Bernardo Melo (Geometria Diferêncial); prof. Arturo Pérez (Análise no $\mathbb{R}^n$); prof. Marcelo Marchesin (Topologia); prof. Silas Luiz de Carvalho (Equações Diferenciais Ordinárias I); prof. Karina Daniela Marin (Análise Funcional).

Por último, mas não menos importante, gostaria de agradecer a todos os colegas que fiz no programa de pós graduação em matemática da UFMG, em especial, ao Célio Terra e ao Genílson Soares de Santana, que nunca deixaram de sanar as minhas dúvidas e me auxiliar durante o mestrado. Gostaria que soubessem que vocês foram fundamentais nessa conquista. Muito obrigado.

Resumo

Apresenta-se uma proposta de alteração no método do Gradiente Conjugado HS (*Hestenes-Stiefel*), denominando-o de GC-HS*, com a finalidade de assegurar que o método esteja sempre bem-definido, independentemente da condição que a busca em linha deva satisfazer, a condição suficiente de descida sempre seja atendida e se tenha, sob determinadas hipóteses, convergência global. Resultados numéricos sugerem que o método proposto é promissor quando comparado aos métodos HS (*Hestenes-Stiefel*), PR (*Polak-Ribière*), DL (Dai-Liao) e GY (Gonglin Yuan).

Palavras-chave: otimização em alta dimensão; gradiente conjugado; gradiente conjugado de *Hestenes-Stiefel* modificado.

Abstract

In this dissertation, we propose modifications in the Conjugate Gradient HS (*Hestenes-Stiefel*) method, naming it GC-HS*, for the purpose of making the method always well defined, regardless of the search line conditions that is being used, ensure that the sufficient descent condition is always attended and there is, under certain hypotheses, global convergence. Numerical results indicate that the proposed method is promising when compared to the methods: HS (*Hestenes-Stiefel*), PR (*Polak-Ribière*), DL (Dai-Liao) and GY (Gonglin Yuan).

Keywords: large scale optimiztion; conjugate gradient; modified *Hestenes-Stiefel* conjugate gradient.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

GC-HS	gradiente conjugado de Hestenes e Stiefel
GC-PR	gradiente conjugado de Polak e Ribière
GC-DL	gradiente conjugado de Dai e Liao
GC-GY	gradiente conjugado de Gonglin Yuan
GC-HS*	gradiente conjugado HS modificado
WWP	condição fraca de Wolfe e Powell
SWP	condição forte de Wolfe e Powell

Lista de símbolos

$\mathbb{N}$	o conjunto dos números reais
$\mathbb{R}^n$	o espaço euclidiano de dimensão n (com produto interno euclidiano e a norma correspondente)
x_i	a coordenada i do elemento $x \in \mathbb{R}^n$ na base canônica
$(x_1, ..., x_n)$	as coordenadas do elemento $x \in \mathbb{R}^n$ na base canônica
$\|x\|$	a norma euclidiana de $x \in \mathbb{R}^n$
$CO\ X$	o envelope convexo do conjunto $X \subset \mathbb{R}^n$
$\overline{X}$	o fecho do conjunto $X \subset \mathbb{R}^n$
A^T	a matriz transposta da matriz A
$f'(x)$	a derivada da função f no ponto x
$\nabla f(x)$	o gradiente da função f no ponto x
g_k	o gradiente da função f no iterando k, $g_k = \nabla f(x_k)$
$\{x_k\}$	uma sequência de pontos no $\mathbb{R}^n$
$\liminf t_k$	o limite inferior da sequência de números reais t_k
$\mathbb{R}^{nm}$	o espaço de matrizes reais de dimensão $n \times m$
β_{k+1}^M	o coeficiente da conjugação do método M;
p_k	a direção de busca em $\mathbb{R}^n$
α_k	o comprimento do passo na direção p_k
t	a constante do método de Dai-Liao
s_k	a diferença vetorial entre pontos consecutivos, $s_k = x_{k+1} - x_k$
$\nabla^2 f(x_k)$	a matriz Hessiana da função f no ponto x;
H_k	aproximação da matriz Hessiana $\nabla^2 f(x_k)$ no ponto x_k
y_k	a subtração vetorial entre os gradientes consecutivos dos iterandos, $y_k = g_{k+1} - g_k$

$F(\alpha)$	é definido por $f(x_k + \alpha p_k)$
$B(x, \varepsilon)$	é o conjunto $\{y \in \mathbb{R}^n;\ \|x - y\| < \epsilon\}$
$\overline{B(x, \varepsilon)}$	é o conjunto $\{y \in \mathbb{R}^n;\ \|x - y\| \leq \epsilon\}$
L	é a constante de Lipschitz

Sumário

1 Introdução

A lei de Moore (ALENCAR, 2019), que indiretamente diz que o desempenho computacional duplica a cada dois anos, mantendo-se o custo operacional, não justifica inteiramente a evolução da computação nas últimas décadas. Segundo Viana (VIANA, 2022), na área da otimização, entre 1990 e 2014, os métodos matemáticos ficaram 870 mil vezes mais rápidos, segundo dados da revista *Operations Research*. Enquanto isso, o desempenho computacional, ligado às questões de *hardware* (lei de Moore), melhorou 6500 vezes. A consequência dessa evolução conjunta é de 5,6 bilhões de vezes, significando que um cálculo que levaria 180 anos caiu para apenas um segundo. Nesse ponto, Philippe Toint, professor da universidade de Namur, opina: "*Preferiria ter os algoritmos matemáticos de hoje nos computadores de ontem, do que o contrário*".

Essa necessidade de evolução computacional, tanto de hardware quanto de métodos matemáticos, é justificada pela demanda de solução de problemas aplicados à engenharia, bioengenharia, medicina, manufatura, economia (YUAN; SHENG; LIU, 2016), dentre outras áreas, que almejam que essa solução seja a melhor possível dentre todas as possibilidades viáveis. Desse modo, estamos lidando com problemas de otimização, em que as respectivas soluções acarretarão o melhor desenvolvimento do meio em que estão inseridas.

Então, busca-se, no presente trabalho, solucionar problemas de otimização irrestrita, formulados matematicamente por:

$$\begin{array}{l} \min \ f(x) \\ \text{s.t. } x \in \mathbb{R}^n \end{array}; \tag{1.1}$$

em que $f : \mathbb{R}^n \to \mathbb{R}$ é denominada função objetivo. Como possíveis métodos de otimização, usados para solucionar numericamente o problema (1.1), cita-se:

- gradiente descendente (RAO, 2019);
- métodos quasi-Newton (BROYDEN, 1967);
- gradiente conjugado (NOCEDAL; WRIGHT, 2006a);
- região de confiança (COLEMAN; LI, 1996).

Entretanto, ao aumentar a dimensão n do espaço de busca $\mathbb{R}^n$, pode-se deparar com obstáculos nos âmbitos:

- computacional, e.g. memória física insuficiente;
- temporal, i.e. o tempo de processamento computacional necessário para solução do problema é impraticável.

Essas inconveniências, causadas pelo aumento da dimensão do problema de otimização (1.1), originaram um novo ramo, denominado: otimização em alta dimensão, que necessita de métodos numéricos dedicados para encontrar a solução de problemas que pertençam a essa classe de otimização.

Dentre um conjunto de problemas de otimização em alta dimensão, serão aqui tratados os problemas de otimização suave e irrestrita, i.e. a função objetivo f, presente no problema (1.1), é continuamente diferenciável. Para essa classe de problemas de otimização escolhida, os métodos de primeira ordem (NOCEDAL; WRIGHT, 2006a) são preferíveis, pois utilizam informações da função f e de seu gradiente (∇f) apenas, não requerendo a construção e armazenamento de matrizes de elevada dimensão. Como consequências destas características, destacam-se como vantagens dos métodos de primeira ordem:

- a necessidade de pouco espaço de armazenamento no computador quando comparados a outros métodos, por exemplo, os métodos de região de confiança (COLEMAN; LI, 1996), que necessitam da informação da Hessiana da função f;
- o tempo de processamento computacional, em geral, é menor que os métodos que exigem manipulações e operações matriciais, principalmente, quando as matrizes são densas, pois requerem menos operações aritméticas em ponto flutuante. Ademais, não há necessidade de utilizar técnicas específicas para lidar com essas matrizes em alta dimensão, e.g. técnicas de pré-condicionamento de matrizes e técnicas de manipulação de matrizes esparsas, quando estas estão presentes.

Visto isso, este trabalho busca fazer uma breve revisão teórica sobre os métodos de primeira ordem (NOCEDAL; WRIGHT, 2006b), em especial os métodos de gradiente conjugado, desenvolvidos a priori para otimização linear (SHEWCHUK, 1994) e, posteriormente, adaptados para a otimização não-linear (NOCEDAL; WRIGHT, 2006a). Dentre esses métodos de gradiente conjugado, são abordados aqui:

- gradiente conjugado Hestenes-Stiefel (GC-HS) (HESTENES; STIEFEL, 1952);
- gradiente conjugado Polak-Ribière (GC-PR) (POLYAK, 1969);
- gradiente conjugado Dai-Liao (GC-DL) (DAI; LIAO, 2001);
- gradiente conjugado de Gonglin Yuan (GC-GY) (YUAN, 2009).

Para um relato histórico dos principais artigos desenvolvidos em otimização não-linear envolvendo os métodos de gradiente conjugado entre os anos de 1946 a 1976, cita-se (GOLUB; O'LEARY, 1989).

Neste trabalho, é proposto um novo método de gradiente conjugado, baseado nos métodos citados acima. Quando se utiliza uma busca em linha baseada nas condições fortes de Wolfe-Powell (NOCEDAL; WRIGHT, 2006a), assumindo ainda determinadas hipóteses sobre a função objetivo, é aqui demonstrado que:

- a condição suficiente de descida (NOCEDAL; WRIGHT, 2006b);
- a condição de Zoutendijk (ZOUTENDIJK, 1970);
- a convergência global (NOCEDAL; WRIGHT, 2006a);

são sempre atendidas. Além disso, os resultados dos experimentos numéricos, apresentados no capítulo 4, sugerem que o método possui bom desempenho computacional quando comparado aos métodos citados.

Contudo, destaca-se que essas exigências para análise de convergência global, realizadas, em geral, para os métodos de gradiente conjugado, limitam sua aplicabilidade a problemas práticos, em que não há muita informação sobre a função objetivo f, presente no problema (1.1), i.e. se f é continuamente diferenciável, se atende todas a hipóteses para a análise de convergência. Por outro lado, se a função objetivo f satisfizer estas condições, e.g. ser uma função estritamente convexa, então esse tipo de problema de otimização irá se beneficiar, convergindo certamente para a solução do problema (1.1). A título de curiosidade, os funcionais de energia em análise estrutural irão se beneficiar ao utilizarem o método do gradiente conjugado proposto neste trabalho.

Estrutura do Trabalho

No capítulo 2 é realizada uma recapitulação dos métodos de busca em linha (seção 2.1): exatos, subseção 2.1.1, e inexatos, subseção 2.1.2. Nesta subseção, são tratadas as principais condições que a busca em linha deve satisfazer: condição de Armijo (subseção 2.1.2.1); condição fraca de Wolfe-Powell (subseção 2.1.2.2); condição forte de Wolfe-Powell (subseção 2.1.2.3). Posteriormente, na seção 2.2, é realizada uma revisão biliográfica sobre os métodos de gradiente conjugado de Hestenes e Stiefel (seção 2.2.1), Polak e Ribière (seção 2.2.2) e Dai e Liao (seção 2.2.3), denominando-os de métodos clássicos.

No capítulo 3, seção 3.1, destacam-se os principais resultados do artigo (YUAN, 2009), utilizado como referência para a construção do método do gradiente conjugado HS modificado (GC-HS*) na seção 3.2. As principais propriedades deste método são descritas na subseção 3.2.1, tendo como ápice a propriedade **(P8)**, que demonstra, sob determinadas hipóteses, a convergência global do método proposto.

Dispondo dos resultados teóricos, há a necessidade de validar o desempenho computacional do novo método em problemas de otimização irrestrita em alta dimensão. Essa tarefa é realizada no capítulo 4.

Por último, no capítulo 5, são apresentadas as conclusões finais do trabalho, além das propostas de continuidade futura com algumas sugestões de como atacá-las.

2 Métodos de Gradiente Conjugado Não-Lineares

Os métodos de gradiente conjugado foram desenvolvidos inicialmente para encontrar a solução de sistemas lineares e, posteriormente, adaptados para solucionar problemas de otimização não lineares (GOLUB; O'LEARY, 1989) formulados por:

$$\begin{array}{l} \min\ f(x) \\ \text{s.t. } x \in \mathbb{R}^n \end{array} ; \tag{2.1}$$

em que $f : \mathbb{R}^n \to \mathbb{R}$ é uma função suave: continuamente diferenciável. Esses métodos possuem estrutura geral definida por:

$$\begin{aligned} x_{k+1} &= \ x_k + \alpha_k p_k; \\ p_k &= \begin{cases} -g_k, & se\ k = 0 \\ -g_k + \beta_k p_{k-1}, & se\ k \geq 1 \end{cases} ; \end{aligned} \tag{2.2}$$

em que:

- $x_k,\ x_{k+1} \in \mathbb{R}^n$ são os iterandos corrente e próximo, respectivamente;
- $\alpha_k \in \mathbb{R}_+$ é o comprimento do passo, determinado conforme técnicas explanadas na seção 2.1, na direção de busca p_k;
- $g_k \in \mathbb{R}^n$ é o gradiente da função f no ponto x_k, $g_k = \nabla f(x_k)$;
- $\beta_k \in \mathbb{R}$ é o coeficiente de conjugação, cuja a expressão depende do método de gradiente conjugado a ser escolhido.

Segundo (SHEWCHUK, 1994), os métodos de gradiente conjugado são caracterizados por:

- serem métodos de primeira ordem, ou seja, utilizam informações da função objetivo f e do seu gradiente ∇f;
- possuírem convergência para a solução do problema quadrático (2.12) em, no máximo, n iterações, em que n é a dimensão do problema;
- serem indicados para a solução de problemas de otimização em alta dimensão.

Essa última característica decorre, principalmente, do fato de os métodos de gradiente conjugado utilizarem regras de cômputo dos passos exigindo, relativamente, pouco espaço de armazenamento quando comparados a outros métodos, por exemplo aqueles baseados em técnicas quasi-Newton (BROYDEN, 1967).

Contudo, antes de serem retratados os métodos clássicos de gradiente conjugado utilizados como referência para a elaboração de uma nova proposta de algoritmo para essa classe de métodos, é apresentada, na próxima seção, uma revisão sobre as técnicas de busca linear.

2.1 Métodos de Busca em Linha

É necessário, nos métodos de gradiente conjugado, estrutura apresentada em (2.2), utilizados para solucionar problemas de otimização não-linear, encontrar o tamanho (comprimento) do passo α_k, na direção p_k, a fim de garantir $f(x_k + \alpha_k p_k) < f(x_k)$ a cada iteração. A análise de convergência dos métodos de otimização em geral dependerá da estratégia adotada para definir o valor de α_k (NOCEDAL; WRIGHT, 2006b). Por exemplo, em (YUAN, 2009) é mencionado que a prova de convergência do método GC-PR para funções não-lineares em geral, com a busca linear atendendo as condições de Wolfe-Powell continua em aberto.

Entretanto, é clara a conveniência de encontrar α_k sem que haja a necessidade de se avaliar a função objetivo f e seu gradiente ∇f muitas vezes. Pois, caso isso ocorra, implicará no aumento do tempo de convergência do método de otimização, tendo em vista que o problema de otimização é em alta dimensão e, muitas vezes, o gradiente da função f é aproximado pelo método das diferenças finitas (ANDREI, 2009). Isto é, não é conhecida a expressão analítica do gradiente da função f.

Dessa forma, existem dois métodos de busca em linha: exato e inexato. Porém, antes de explanarmos estes métodos, é apresentada a seguir uma importante definição que será utilizada nas próximas subseções.

Definição 1. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função continuamente diferenciável. Dizemos que $p \in \mathbb{R}^n$ é uma direção de descida de f no ponto x se:*

$$\nabla f(x)^\intercal p < 0. \tag{2.3}$$

2.1.1 Busca em Linha Exata

Na busca em linha exata, para todo iterando $k \geq 0$, α_k é a solução do problema de otimização unidimensional, em relação a variável α e na direção de descida p_k, definido por:

$$\alpha_k = \arg\min \ F(\alpha); \tag{2.4}$$

em que $F(\alpha) = f(x_k + \alpha p_k)$.

Dentre os métodos numéricos para encontrar a solução do problema unidimensional (2.4), podem ser citados:

- métodos diretos (utilizam apenas os valores de $F(\alpha)$) (RAO, 2019):
  - seção áurea;
  - interpolação quadrática;
  - interpolação cúbica.
- métodos indiretos (utilizam os valores das derivadas de primeira e segunda ordem de $F(\alpha)$) (NOCEDAL; WRIGHT, 2006b):
  - método de Newton;
  - método quasi-Newton;
  - método da secante.

Apesar de serem métodos exatos, tais métodos possuem como principal desvantagem o elevado número de avaliações da função objetivo f e/ou suas derivadas f', tornando o seu uso prático em algoritmos computacionais questionável (NOCEDAL; WRIGHT, 2006b). Principalmente, quando se lida com otimização em alta dimensão e otimização com elevado custo computacional, ou seja, gasta-se um tempo elevado para avaliar a função objetivo no ponto desejado.

Logo, a fim de sanar esse problema do excesso do número de avaliações, utiliza-se a busca em linha inexata.

2.1.2 Busca em Linha Inexata

A busca em linha inexata é a mais indicada para encontrar o comprimento de passo que alcance a redução da função f adequadamente, sem despender um alto número de avaliações da função objetivo f (NOCEDAL; WRIGHT, 2006b).

Em linhas gerais, esse tipo de busca inexata gera valores para α até que certas condições de $F(\alpha)$ e $F'(\alpha)$ sejam satisfeitas, exigindo um número de avaliações da função f menor que a busca em linha exata.

Portanto, são abordadas nas subseções seguintes as principais condições que a busca em linha inexata deve satisfazer, bem como suas características essenciais. Tais condições são utilizadas na próxima seção 2.2 e nos capítulos 3 e 4.

2.1.2.1 Condição de Armijo

A condição de Armijo é encontrar α tal que:

$$F(\alpha) = f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^{\intercal} p_k, \tag{2.5}$$

com $c_1 \in (0,1)$ constante. (NOCEDAL; WRIGHT, 2006b) cita que, na prática (para fins de implementação computacional), c_1, para os métodos de gradiente conjugado, pode ser igual a 10^{-4}.

Contudo, a desigualdade (2.5) não é suficiente para assegurar uma boa evolução do algoritmo de otimização, pois permite que α assuma valores excessivamente próximos a 0, sem sequer investigar tamanhos maiores de passo.

Para implementação da condição de Armijo e visando evitar que α seja um número inicializado excessivamente próximo a zero, utiliza-se o algoritmo *Backtracking Line Search* 1.

Algorithm 1 Algoritmo *Backtracking* - Condição Armijo

Entrada: $x_k \in \mathbb{R}^n$, $p_k \in \mathbb{R}^n$, $\bar{\alpha} \in (0, \infty)$, $\rho \in (0,1)$ e $c_1 \in (0,1)$.
Saída: α_k
  Faça $\alpha = \bar{\alpha}$.
  enquanto $f(x_k + \alpha p_k) > f(x_k) + c_1 \alpha \nabla f(x_k)^\intercal p_k$ **faça** ▷ *xrepetição*
    $\alpha = \rho\alpha$;
  fim enquanto
  $\alpha_k = \alpha$ ▷ *final*

Em (NOCEDAL; WRIGHT, 2006b) menciona-se que é comum $\bar{\alpha} = 1$ em métodos de Newton e quasi-Newton. E superior a 1 para métodos como gradiente conjugado e maior direção de descida (*method of steepest descent*) (RAO, 2019).

O seguinte lema assegura a existência de um intervalo no qual a condição de Armijo é satisfeita, desde que p_k seja uma direção de descida.

Lema 1) Se $c_1 \in (0,1)$ e $f : \mathbb{R}^n \to \mathbb{R}$ continuamente diferenciável. Seja $x \in \mathbb{R}^n$ e $p \in \mathbb{R}^n$ uma direção de descida de f neste ponto, então existe $\alpha^* > 0$ tal que:

$$f(x + \alpha p) \leq f(x) + c_1 \alpha \nabla f(x)^\intercal p \quad \forall \alpha \in [0, \alpha^*]. \tag{2.6}$$

Demonstração: Lema 4.6, página 50 de (KRESSNER, 2015).

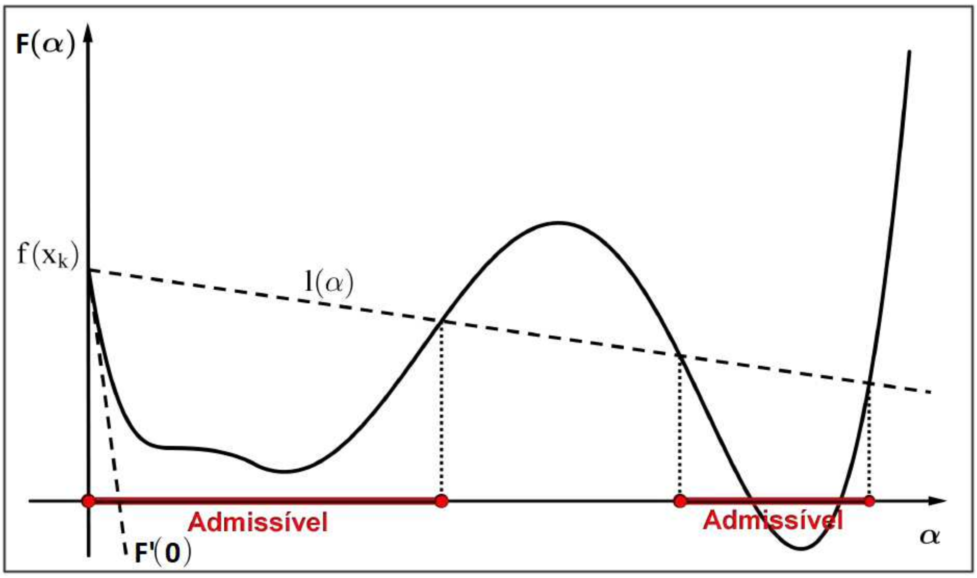
Uma consequência direta do **Lema 1** é que o algoritmo 1 gera uma sequência finita de valores de α, tendo seu último elemento satisfazendo condição de Armijo (2.5). A titulo de ilustração, a figura 1 apresenta o intervalo de α no qual é satisfeita a condição de Armijo (2.5), sendo válidas as hipóteses do **Lema 1**.

2.1.2.2 Condição Fraca de Wolfe-Powell

Visando garantir que α_k não seja escolhido excessivamente próximo a zero, e, consequentemente, possibilite aos métodos de otimização um melhor progresso, por meio de menor número de iterações para convergência, foi estabelecida a condição de curvatura, exigindo que α_k satisfaça:

$$\nabla f(x_k + \alpha_k p_k)^\intercal p_k \geq c_2 \nabla f(x_k)^\intercal p_k \tag{2.7}$$

Figura 1 – Intervalo admissível para α satisfazendo a condição de Armijo, equação (2.5). Sendo $l(\alpha) = f(x_k) + c_1 \alpha \nabla f(x_k)^\intercal p_k$.



Fonte: figura do autor.

com $c_2 \in (c_1, 1)$. As condições (2.5) e (2.7) são denominadas juntas condição fraca de Wolfe-Powell (Powell) ou condição padrão de Wolfe-Powell (Powell).

No algoritmo 2 é adotado o método da bisseção como critério para geração do próximo ponto a ser examinado, a fim de atender as condições fracas de Wolfe-Powell, garantido, sob determinadas hipóteses, pelo corolário 2.1.

Algorithm 2 Algoritmo da Bisseção - Condição Fraca de Wolfe-Powell

Entrada: $x_k \in \mathbb{R}^n,\ p_k \in \mathbb{R}^n$, $c_1 \in (0, 1)$ e $c_2 \in (c_1, 1)$.
Saída: α_k

Faça $\alpha = 0$, $t = 1$ e $\beta = +\infty$.

enquanto $f(x_k + tp_k) > f(x_k) + c_1 t \nabla f(x_k)^\intercal p_k$ **faça** ▷ *xrepetição*

$$\beta = t \quad e \quad t = \frac{1}{2}(\alpha + \beta); \tag{2.8}$$

fim enquanto

enquanto $\nabla f(x_k + tp_k)^\intercal p_k < c_2 \nabla f(x_k)^\intercal p_k$ **faça** ▷ *xrepetição*

$$\alpha = t \quad e \quad t = \begin{cases} 2\alpha, & se \ \ \beta = +\infty \\ \dfrac{1}{2}(\alpha + \beta), & \text{caso contrário} \end{cases}. \tag{2.9}$$

fim enquanto

$\alpha_k = t$ ▷ *final*

Lema 2: Seja $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ uma função continuamente diferenciável, $x \in \mathbb{R}^n$ e $p \in \mathbb{R}^n$ uma direção de descida de f nesse ponto. Então, apenas uma das seguintes possibilidades para o algoritmo da bisseção 2 deve ser satisfeita.

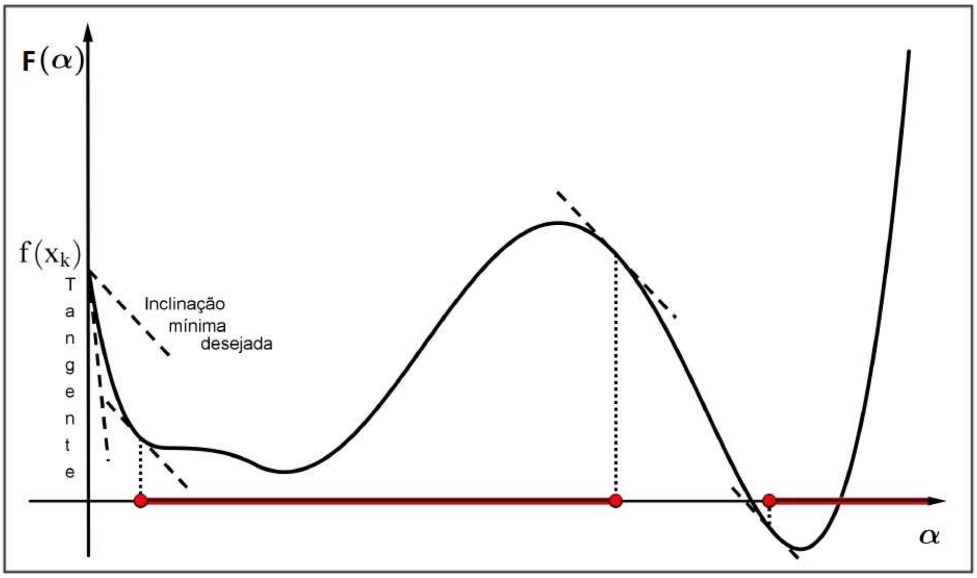
i) O conjunto de valores, gerados pelo algoritmo 2, para t é finito, sendo que o último valor de t satisfaz as condições fracas de Wolfe-Powell.

ii) O algoritmo 2 gera uma sequência infinita de valores para t, o parâmetro β nunca possui um valor finito, α torna-se positivo na primeira iteração e dobra de valor a cada iteração, consequentemente, $f(x+tp) \longrightarrow -\infty$.

Demonstração: Lema 6.4, página 80 de (BURKE, 2020).

Corolário 2.1: Sejam válidas as hipóteses do **Lema 2** com f limitada ao longo do caminho retilíneo $\{x + \alpha p; \alpha > 0\}$. Então, é válida a condição *(i)* do **Lema 2**.

Figura 2 – Intervalo admissível para α satisfazendo a condição de curvatura, equação (2.7).

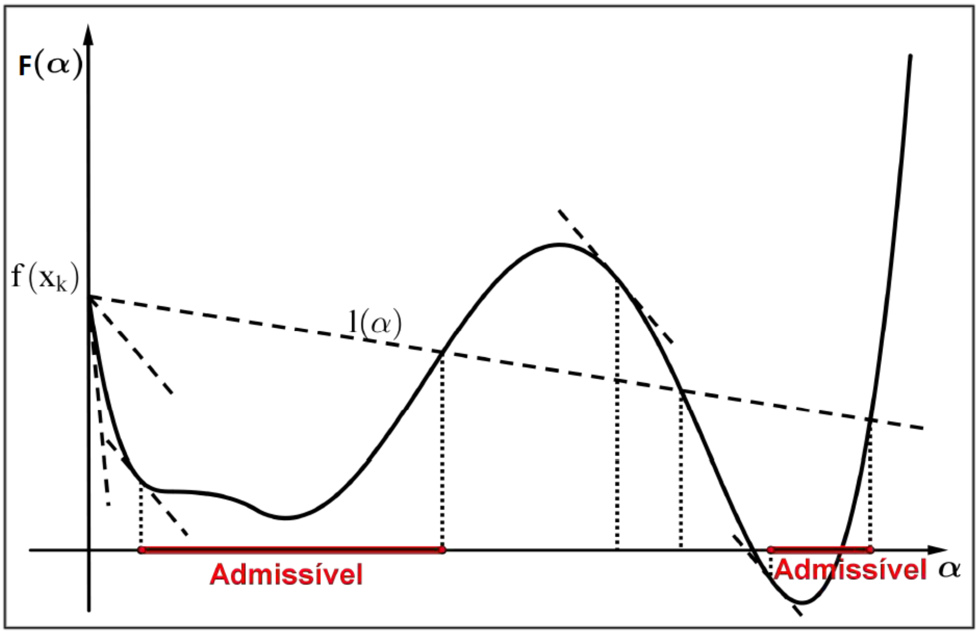


Fonte: figura do autor.

Além disso, sendo válidas as hipóteses do **Lema 2**, as figuras 2 e 3 apresentam, respectivamente, os intervalos no qual α atende somente a condição de curvatura, equação (2.7), e a condição fraca de Wolfe-Powell. É notório que o intervalo desta condição de Wolfe-Powell está contido naquele intervalo.

Contudo, satisfazer as condições fracas de Wolfe-Powell não implica necessariamente que α_k esteja perto de um minimizador ou ponto estacionário de f na direção p (NOCEDAL; WRIGHT, 2006b). Então, visando sanar este problema, restringe-se a condição de curvatura a fim de forçar α_k a pertencer a essa vizinhança de interesse.

Figura 3 – Intervalo admissível para α satisfazendo a condição fraca de Wolfe-Powel. Sendo $l(\alpha) = f(x_k) + c_1\alpha\nabla f(x_k)^\intercal p_k$.



Fonte: figura do autor.

2.1.2.3 Condição Forte de Powell

Uma das possibilidades para restringir a condição de curvatura é impor que α_k satisfaça:

$$|\nabla f(x_k + \alpha_k p_k)^\intercal p_k| \leq c_2 |\nabla f(x_k)^\intercal p_k|, \tag{2.10}$$

com $c_2 \in (c_1, 1)$.

A condição forte de Wolfe-Powel é satisfeita quando α_k verifica (2.5) e (2.10). A única diferença entre (2.7) e (2.10) é que esta, limita superiormente $F'(\alpha_k)$, não permitindo que assuma valores positivos elevados quando se toma o produto interno com a direção de descida p_k. O lema a seguir afirma, sob certas hipóteses, a existência de um intervalo em que as condições fracas e fortes de Wolfe-Powell são satisfeitas. Tal intervalo é importante para o desenvolvimento do algoritmo de busca que atenda as estas condições. Sob as hipóteses do **Lema 3**, a figura 4 ilustra o intervalo da reta que atende a condição forte de Wolfe-Powell.

Lema 3: Seja $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ uma função continuamente diferenciável. Considere-se ainda, para $x \in \mathbb{R}^n$, $p \in \mathbb{R}^n$, uma direção de descida de f nesse ponto, com f inferiormente limitada ao longo do caminho retilíneo $\{x + \alpha p; \alpha > 0\}$. Então, se $0 < c_1 < c_2 < 1$, existe um intervalo satisfazendo as condições fracas e fortes de Wolfe-Powell.

Demonstração: Lema 3, página 35 de (NOCEDAL; WRIGHT, 2006b).

Podem ser utilizados os algoritmos 3 e 4, transcritos das páginas 60 e 61 de (NOCEDAL; WRIGHT, 2006b) a fim de encontrar α_k que atenda a condição forte de Wolfe-Powell. Percebe-se que encontrar o comprimento de passo que atenda a esta condição envolve duas etapas. A primeira, tem o objetivo de construir uma sequência de valores α_i tal que $f(x_k + \alpha_i p_k)_i$ seja monótona, sendo finalizada quando o comprimento de passo encontrado satisfazer a condição forte de Wolfe-Powell.

A segunda etapa do algoritmo 3, por sua vez, tem a finalidade de definir um intervalo que contenha os comprimentos de passo α_i desejáveis, sendo denominada de função *zoom* e descrita no algoritmo 4. Esta função, no decorrer das iterações, diminui o intervalo $(\alpha_{\text{min}}, \alpha_{\text{max}})$, presente como entrada no algoritmo 3, até que seja encontrado um comprimento de passo desejável. Por fim, sob as hipóteses do **Lema 3**, demonstrasse em (AL-BAALI; FLETCHER, 1986) que o algoritmo 3 termina após um número finito de iterações.

Algorithm 3 Algoritmo de Busca para a Condição Forte de Wolfe-Powell

Entrada: $x_k \in \mathbb{R}^n$, $p_k \in \mathbb{R}^n$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $\alpha_{\text{min}} > 0$, $\alpha_{\text{max}} > \alpha_{\text{min}}$.
Saída: α_k
  Escolha $\alpha_0 \in (\alpha_{\text{min}}, \alpha_{\text{max}})$ e faça $i = 0$.
  enquanto não parar **faça** ▷ *xrepetição*
    se $f(x_k + \alpha_i p_k) > f(x_k) + c_1 \alpha_i \nabla f(x_k)^\intercal p_k$ ou $[f(x_k + \alpha_i p_k) \geq f(x_k + \alpha_{i-1} p_k)$ e $i > 0]$ **então**
      Obtenha α_k pelo processo *zoom*(α_{i-1}, α_i) e **pare**;
    fim se
    se $|\nabla f(x_k + \alpha_i p_k)^\intercal p_k| \leq -c_2 \nabla f(x_k)^\intercal p_k$ **então**
      $\alpha_k = \alpha_i$ e **pare**;
    fim se
    se $\nabla f(x_k + \alpha_i p_k)^\intercal p_k \geq 0$ **então**
      Obtenha α_k pelo processo *zoom*(α_i, α_{i-1}) e **pare**;
    fim se
    Escolha $\alpha_{i+1} \in (\alpha_{i+1}, \alpha_{\text{max}})$.
    Faça $i = i + 1$.
  fim enquanto

A medida que é alterado o algoritmo de busca com a finalidade de atender as condições de Armijo, fraca de Wolf-Powell e forte de Wolf-Powell são aumentadas as restrições para a busca de α_k. Tal fato implica em intervalos viáveis mais restritos e, consequentemente, demandando maior número de avaliações da função objetivo f e de seu respectivo gradiente. Esse fato pode ser visualizado ao analisar as figuras 1, 3 e 4.

Destaca-se que a condição forte de Wolfe-Powell foi utilizada no capítulo 3 para a prova de convergência global. Por sua vez, para os experimentos numéricos apresentados no capítulo 4 foi utilizada a condição fraca de Wolfe-Powell, implementada por meio do algoritmo da bisseção 2.

Algorithm 4 Algoritmo *ZOOM*

Entrada: $x_k \in \mathbb{R}^n,\ p_k \in \mathbb{R}^n$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $0 \leq \alpha_\text{l} < \alpha_\text{u}$
Saída: α
  Faça $j = 0$.
  enquanto não parar **faça** ▷ *xrepetição*
    Escolha um passo $\alpha_j \in (\alpha_\text{l}, \alpha_\text{u})$.
    Calcule $f(x_k + \alpha_j p_k)$.
    se $f(x_k + \alpha_j p_k) > f(x_k) + c_1 \alpha_j \nabla f(x_k)^\intercal p_k$ ou $f(x_k + \alpha_j p_k) \geq f(x_k + \alpha_\text{l} p_k)$ **então**
      $\alpha_\text{u} = \alpha_j$
    senão
      Calcule $\nabla f(x_k + \alpha_j p_k)^\intercal p_k$.
      se $|\nabla f(x_k + \alpha_j p_k)^\intercal p_k| \leq -c_2 \nabla f(x_k)^\intercal p_k$ **então**
        $\alpha = \alpha_j$ e **pare**;
      fim se
      se $(\nabla f(x_k + \alpha_j p_k)^\intercal p_k)(\alpha_\text{u} - \alpha_\text{l}) \geq 0$ **então**
        $\alpha_\text{u} = \alpha_\text{l}$;
      fim se
      $\alpha_\text{l} = \alpha_\text{j}$;
    fim se
    Faça $j = j + 1$.
  fim enquanto

2.2 Os Métodos de Gradiente Conjugado Clássicos

Nessa seção são apresentadas a principais características dos métodos de gradiente conjugado clássicos utilizados como referência para a construção do método do gradiente conjugado proposto nesse trabalho, o GC-HS*.

2.2.1 O Método do Gradiente Conjugado de Hestenes e Stiefel (GC-HS)

Desenvolvido por Hestenes-Stiefel (HESTENES; STIEFEL, 1952), o método do gradiente conjugado HS é um algoritmo que busca a solução de sistemas lineares:

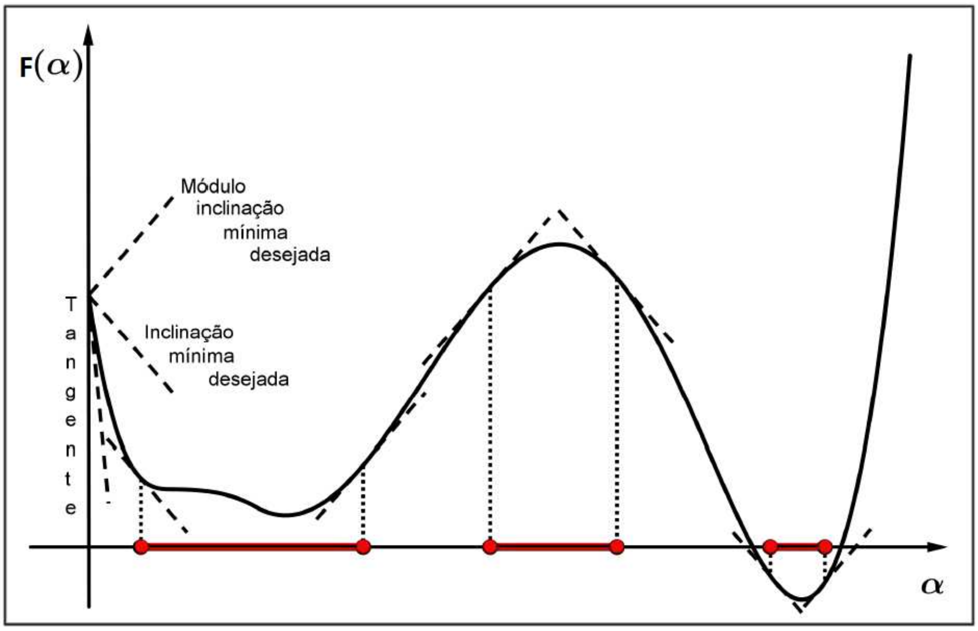
$$Ax = b; \tag{2.11}$$

em que:

- A $\in \mathbb{R}^{n \times n}$, simétrica ($A^\intercal = A$) e definida positiva ($A > 0$);
- $x, b \in \mathbb{R}^n$.

Este método pode ser preferível em relação aos métodos diretos, e.g. método de Eliminação de Gauss-Jordan (SAFF; SNIDER, 2015) e método de Crout (SAFF; SNIDER, 2015), quando os

Figura 4 – Intervalo admissível para α satisfazendo a condição forte de Wolfe-Powel.



Fonte: figura do autor.

problemas são em alta dimensão. Isto devido, por exemplo, a menor necessidade de armazenamento de matrizes de grandes dimensões.

Solucionar (2.11) é equivalente a encontrar a solução do problema de otimização quadrática (SHEWCHUK, 1994):

$$\begin{array}{l} \min \ \frac{1}{2}x^{\intercal}Ax - b^{\intercal}x \\ \text{s.t. } x \in \mathbb{R}^n \end{array} . \tag{2.12}$$

Pois, A definida positiva assegura que $f(x) = \frac{1}{2}x^{\top}Ax - b^{\top}x$ seja uma função estritamente convexa e, consequentemente, possua um único ponto de mínimo local (ISMAILOV; SOLODOV, 2005). Ou seja, o problema (2.12) possui solução única.

O método do gradiente conjugado HS possui estrutura definida em (2.2), com β_k, o coeficiente de conjugação, sendo fixado por:

$$\beta_k^{HS} = \frac{g_k^{\intercal}y_{k-1}}{p_{k-1}^{\intercal}y_{k-1}}; \tag{2.13}$$

em que: $y_k = g_k - g_{k-1}$.

Para os problemas quadráticos (2.12), α_k é calculado de forma exata, sendo definido por:

$$\alpha_k = \frac{p_k^{\intercal}(b - Ax_k)}{p_k^{\intercal}Ap_k} = \frac{p_k^{\intercal}r_k}{p_k^{\intercal}Ap_k}; \tag{2.14}$$

em que: $r_k = b - Ax_k$ é o resíduo. A norma deste, $\|r_k\|$, indica a que distância Ax_k encontra-se de b no subespaço imagem de A. Enfatiza-se o comportamento monotônico da sequência $(\|r_k\|)_{k\in\mathbb{N}}$ em sua aproximação a 0, realizada em no máximo n iterações. Porém, em problemas de otimização em alta dimensão, considerando a precisão finita dos computadores, pode-se extrapolar o número de n iterações devido ao acumulo de erro numérico ao longo dessas iterações.

Contudo, para problemas de otimização em que a função objetivo f, indicada na equação (2.1), não tem uma forma quadrática, tal como na expressão (2.12), há necessidade de ser realizada uma busca em linha, apresentada na seção 2.1, a fim de permitir a localização de um bom valor de α_k. Deve-se notar que o resíduo r_k pode ser interpretado como $r_k = -\nabla f(x_k)$, quando a função $f(x)$ é definida como a função objetivo quadrática em (2.12). Desta forma, no caso de problemas com função objetivo não-quadrática, é possível aplicar este método fazendo r_k igual a menos o gradiente da função.

Destaca-se para o método GC-HS que:

- a condição de conjugação:

$$p_{k+1}^{\intercal} y_k = 0; \tag{2.15}$$

  é válida para todo $k > 0$, independentemente de f, presente no problema de otimização irrestrito (2.1), e da busca linear que está sendo utilizada (ZHANG, 2009);

- há necessidade que a busca linear satisfaça, no mínimo, a condição fraca de Wolfe-Powell, apresentada na subseção 2.1.2.2, para que β_k^{HS} esteja bem definido ou seja: $p_k^{\intercal} y_k > 0$;

- esse método possui bom desempenho computacional (YUAN, 2009), no sentido que o número de iterações e o tempo de processamento computacional até a solução do problema são menores quando comparados aos métodos: Fletcher e Reeves (FLETCHER; REEVES, 1964); CD (FLETCHER, 1987); DY (DAI; YUAN, 2000); LS (LIU; STOREY, 1991).

2.2.2 O Método do Gradiente Conjugado Polak e Ribière (GC-PR)

Proposto por Polak e Ribière (POLAK; RIBIÈRE, 1969), o método do gradiente conjugado desenvolvido, o GC-PR, representa uma importante contribuição para a otimização não-linear irrestrita (GOLUB; O'LEARY, 1989). Este método define o coeficiente de conjugação β_k^{PR} como:

$$\beta_k^{PR} = \frac{g_k^{\intercal} y_{k-1}}{g_{k-1}^{\intercal} g_{k-1}}; \tag{2.16}$$

tendo como principais características:

- para problemas de otimização quadráticas, quando se usa a busca em linha exata, subseção 2.1.1, é provado a sua convergência global em (ZOUTENDIJK, 1970);

- para funções não-lineares e utilizando a busca linear exata, subseção 2.1.1, não há garantia de convergência global. Powell (POWELL, 1984) apresentou um exemplo em $\mathbb{R}^3$ para o qual o método PR é infinitamente cíclico, não convergindo a um ponto estacionário de f;

- para funções não-lineares e com busca em linha inexata, subseção 2.1.2, o método PR, quando converge, apresenta bom desempenho computacional em relação aos métodos: Fletcher e Reeves; CD; DY; LS (YUAN, 2009). Já em relação ao método GC-HS, (NOCEDAL; WRIGHT, 2006a) menciona que não há distinção evidente de desempenho computacional, sendo esses dois métodos, GC-HS e GC-PR, preferíveis aos demais métodos (Fletcher e Reeves; CD; DY; LS);

- quando o método PR é utilizado com a busca em linha satisfazendo a condição forte de Wolfe-Powell, abordada na subseção 2.1.2.3, não há garantia que a condição de descida ($g_k^{\intercal} p_k < 0$) seja sempre satisfeita (NOCEDAL; WRIGHT, 2006a). Então, para evitar esse problema é realizada uma pequena alteração no coeficiente de conjugação (2.16), proposta em (GILBERT; NOCEDAL, 1992), definindo:

$$\beta_k^{PR^*} = \max\{\beta_k^{PR}, 0\}. \tag{2.17}$$

Assim, ao computar o coeficiente de conjugação como definido em (2.17) juntamente com a busca linear, satisfazendo a condição forte de Wolfe-Powell, tem-se a condição de descida sempre satisfeita. Também em (GILBERT; NOCEDAL, 1992) prova-se, sob um conjunto de hipóteses e utilizando $\beta_k^{PR^*}$, a convergência global.

2.2.3 O Método do Gradiente Conjugado Dai e Liao (GC-DL)

O método proposto por Dai e Liao, em (DAI; LIAO, 2001), é indicado para problemas de otimização irrestrita em alta dimensão. Além disso, possui estrutura definida em (2.2) com o coeficiente de conjugação sendo determinado por:

$$\beta_{k+1}^{DL} = \beta_{k+1}^{HS} - t\frac{g_{k+1}^{\intercal} s_k}{p_k^{\intercal} y_k}, \quad t \geq 0, \quad \forall k > 0. \tag{2.18}$$

em que: $s_k = x_{k+1} - x_k$.

É conhecido que os métodos de gradiente conjugado produzem uma sequência de direções de busca p_k que satisfazem a condição:

$$p_j^{\intercal} Q p_i = 0, \quad i \neq j; \tag{2.19}$$

denominada condição de conjugação, em que Q é a hessiana de uma função quadrática convexa.

Porém, para uma função f não-linear no geral, utilizando a definição de $y_k = g_{k+1} - g_k$, em que $g_k = \nabla f(x_k)$, e aplicando o teorema do valor médio, tem-se:

$$p_{k+1}^{\intercal} y_k = \alpha_k p_{k+1}^{\intercal} \nabla^2 f(x_k + \theta\alpha_k p_k) p_k, \tag{2.20}$$

com $\theta \in (0, 1)$. De (2.19) e (2.20) é razoável considerar a condição de conjugação para funções não-lineares no geral identicamente nula, ou seja:

$$p_{k+1}^{\intercal} y_k = 0. \tag{2.21}$$

Por outro lado, os métodos quasi-Newton (BROYDEN, 1967) realizam a construção iterativa de uma matriz H_{k+1}, possuindo as seguintes características:

- H_{k+1} é uma aproximação de $\nabla^2 f(x_{k+1})$;
- se a condição de curvatura é satisfeita, $s_k^\intercal y_k > 0$, então $H_{k+1}s_k = y_k$, satisfazendo a equação secante;
- a direção de busca é definida nesses métodos como:

$$p_{k+1} = -H_{k+1}^{-1} g_{k+1}. \tag{2.22}$$

Da equação secante e (2.22), tem-se:

$$p_{k+1}^\intercal y_k = p_{k+1}^\intercal (H_{k+1}s_k) = -g_{k+1}^\intercal s_k. \tag{2.23}$$

Mantendo esta relação como âncora, Dai e Liao (DAI; LIAO, 2001) propuseram a condição de conjugação sendo computada por:

$$p_{k+1}^\intercal y_k = -t g_{k+1}^\intercal s_k. \tag{2.24}$$

Em seguida, fazendo o produto interno de $p_{k+1} = -g_{k+1} - \beta_k p_k$ com y_k e isolando β_k, obtém-se (2.18).

Esse método possui as seguintes particularidades de acordo com (DAI; LIAO, 2001):

- se adotado o método de busca linear exato, então $\beta_{k+1}^{DL} = \beta_{k+1}^{HS}$;
- há necessidade que a busca linear satisfaça, no mínimo, a condição fraca de Wolfe-Powell, apresentada na subseção 2.1.2.2, para que β_{k+1}^{DL} esteja bem definido ou seja: $p_k^\intercal y_k > 0$;
- se a direção gerada pelo método HS, abordado na seção 2.2.1, for de descida e a busca linear fornecer uma relação $p_k^\intercal y_k > 0$, que é verificada ao satisfazer a condição fraca de Wolfe-Powell, então, o método de Dai-Liao gera direções que são de descida;
- há convergência global para funções uniformemente convexas, ver definição 10.5 da página 144 de (BAUSCHKE; COMBETTES et al., 2011), que incluem as funções fortemente convexas;
- assim como Gilbert e Nocedal (GILBERT; NOCEDAL, 1992) fizeram para o método PR, definindo $\beta_k^{PR^*}$ (2.17) para a demonstração de convergência global, Dai e Liao definiram:

$$\beta_{k+1}^{DL^*} = \max\{\beta_{k+1}^{HS}, 0\} - t\frac{g_{k+1}^\intercal s_k}{p_k^\intercal y_k}. \tag{2.25}$$

  Com essa definição, é possível provar a convergência global (teorema 3.6 de (DAI; LIAO, 2001)) se: a busca linear satisfizer a condição forte de Wolfe-Powell; a condição suficiente de descida for atendida, i.e. $\exists\ c > 0$ tal que $g_k^\intercal p_k \leq -c\|g_k\|^2\ \forall k \geq 0$;

- experimentos numéricos indicaram que $t = 0,1$ é uma boa escolha de valor de parâmetro para problemas de otimização em alta dimensão.

3 O Método do Gradiente Conjugado Não-Linear com a Condição de Descida Suficiente

São apresentados, na seção seguinte, 3.1, os pontos principais do artigo *modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems* (YUAN, 2009). O citado artigo foi utilizado como referência para o desenvolvimento do método do gradiente conjugado HS modificado (GC-HS*), descrito na seção 3.2. Na subseção 3.2.1 são apresentados suas principais propriedades.

3.1 Gradiente Conjugado - Gonglin Yuan (YUAN, 2009)

A fim de solucionar:

$$\begin{array}{c} \min \ f(x) \\ \text{s.t. } x \in \mathbb{R}^n \end{array} ; \tag{3.1}$$

em que $f : \mathbb{R}^n \to \mathbb{R}$ é uma função suave, continuamente diferenciável, e utilizando, como base, o método GC-PR, seção 2.2.2, Yuan propôs o algoritmo 5.

Deve-se destacar que dada a sequência $\{d_k\}$, $\{g_k\}$ e $\{\alpha_k\}$, gerada pelo algoritmo 5, tem-se que a condição suficiente de descida é atendida (Teorema 3.1 de (YUAN, 2009)). Portanto, define-se:

Definição 2. *Dizemos que a condição suficiente de descida é atendida se existir uma constante $c > 0$ tal que para cada direção p_k, tem-se:*

$$g_k^\intercal p_k \leq -c\|g_k\|^2 \qquad \forall k \geq 0. \tag{3.2}$$

Além disso, pressupõe-se como hipóteses:

H1) O conjunto $\Omega = \{x \in \mathbb{R}^n; f(x) \leq f(x_0)\}$ é limitado, ou seja:

$$\|x\| \leq M, \quad \forall\, x \in \Omega, M > 0.$$

H2) No conjunto $\Omega_0 \subset \mathbb{R}^n$, aberto e convexo, com $\Omega \subset \Omega_0$, tem-se:

H2.1) f é inferiormente limitado;

H2.2) f é diferenciável;

H2.3) f' é Lipschitz contínua, ou seja, existe $L > 0$ tal que

$$\|f'(x) - f'(y)\| \leq L\|x - y\| \quad \forall\, x, y \in \Omega_0.$$

Algorithm 5 Gradiente Conjugado de Gonglin Yuan (GC-GY) - (YUAN, 2009)

Entrada: $x_0 \in \mathbb{R}^n$, $\varepsilon \in (0,1)$, $c_1 \in \left(0, \frac{1}{2}\right)$, $c_2 \in (c_1, 1)$ e $\mu > 0$.
Saída: $\bar{x}$ (solução de (3.1)).
Faça $p_0 = -g_0 = -\nabla f(x_0)$ e $k = 0$.
enquanto $\|g_k\| > \varepsilon$ **faça** ▷ *xrepetição*

Y.1.2 Encontre α_k tal que:

$$\left.\begin{array}{l} f(x_k + \alpha_k p_k) - f(x_k) \leq c_1 \alpha_k g_k^T p_k \\ g(x_k + \alpha_k p_k)^\intercal p_k \geq c_2 g_k^\intercal p_k \end{array}\right\} \text{Condição Fraca de Wolfe-Powell;}$$

Y.1.3 Defina $x_{k+1} = x_k + \alpha_k p_k$.
Y.1.4 Calcule a nova direção de busca:

$$p_{k+1} = -g_{k+1} + \beta_{k+1}^{\text{MPRP}} p_k;$$

tal que:

$$\beta_{k+1}^{\text{MPRP}} = B_k^{PR} - \min\left\{B_k^{PR}, \frac{\mu\|y_k\|^2}{\|g_k\|^4} g_{k+1}^\intercal p_k\right\},$$

com $B_k^{PR} = \beta_{k+1}^{PR}$, em que β_{k+1}^{PR} é definido conforme (2.16).
Y.1.5 Faça $k = k + 1$.
fim enquanto
Faça $\bar{x} = x_k$ ▷ *final*

Yuan demonstra que:

- a condição de Zountendijk (ZOUTENDIJK, 1970) é satisfeita, ou seja:

$$\sum_{k=0}^{\infty} \frac{(g_k^\intercal p_k)^2}{\|p_k\|^2} < \infty; \tag{3.3}$$

- se existir um $\alpha > 0$, tal que $\alpha_k \geq \alpha$ para todo $k \geq 0$, então o algoritmo 5 possui convergência global, equivalentemente:

$$\lim_{k \to \infty} \|g_k\| = 0; \tag{3.4}$$

Na seção 4 de (YUAN, 2009) é apresentada uma generalização do algoritmo proposto, com base no método PR, para os demais métodos clássicos: FR (FLETCHER; REEVES, 1964); DY (DAI; YUAN, 1999); LS (LIU; STOREY, 1991); HS, descrito na seção 2.2.1 do presente trabalho. Para todos eles Yuan afirma que as propriedades apresentadas acima são satisfeitas sob as hipóteses citadas.

O tema central desta dissertação consiste em examinar algumas possibilidades de extensões do artigo (YUAN, 2009). São investigadas aqui:

- qual seria a consequência de retirar a hipótese de α_k ser inferiormente limitada por uma constante positiva e supor que essa sequência seja positiva;

- ao adotar GC-HS, apresentado na seção 2.2.1, como método clássico base não haveria situações em que o valor de B_k^{MHS}, definido por:

$$B_k^{\text{MHS}} = B_k^{HS} - \min\left\{B_k^{HS}, \frac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k\right\}; \tag{3.5}$$

  estaria numericamente mal dimensionado (condicionado), fazendo com que a norma de p_{k+1} apresente valores muito grandes e, assim, exija um número muito alto de iterações para a busca em linha, seção 2.1.

Além disso,

- seria possível utilizar a condição de conjugação da função f não identicamente nula, assim como o Método do Gradiente conjugado não-linear de Dai-Liao (DL) (seção 2.2.3);
- qual seria o impacto em se utilizar a condição de curvatura, representada por meio da equação (2.7), para reiniciar o cálculo das direções conjugadas.

Então, com o objetivo de atacar esses pontos, é apresentado na próxima seção o Método do Gradiente Conjugado HS Modificado.

3.2 O Método do Gradiente Conjugado HS Modificado

Nessa seção são apresentados os algoritmos 6 e 7 com suas principais propriedades descritas na próxima subseção. Considera-se o problema de otimização definido em (3.1), com f continuamente diferenciável.

Para fins da análise de convergência global, realizada na propriedade **(P8)**, são alteradas as condições de busca em linha, substituindo **A.1.2** do algoritmo 6 pelo passo **A.1.2*** no algoritmo 7.

3.2.1 Propriedades do Método do Gradiente Conjugado HS Modificado

São demonstradas as principais propriedades do método do gradiente conjugado HS modificado (GC-HS*), provando-se, sob determinadas hipóteses, a convergência global desse método.

Previamente, é relembrado a seguir uma importante definição, apresentada na seção 2.1, para os métodos de otimização, que buscam solucionar o problema (3.1), em que f é continuamente diferenciável.

Definição 3. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função continuamente diferenciável. Dizemos que $p \in \mathbb{R}^n$ é uma direção de descida de f no ponto x se:*

$$\nabla f(x)^\intercal p < 0. \tag{3.6}$$

Algorithm 6 Algoritmo HS Modificado satisfazendo a condição fraca de Wolfe-Powell

Entrada: $x_0 \in \mathbb{R}^n$, $\varepsilon \in (0,1)$, $c_1 \in \left(0, \frac{1}{2}\right)$, $c_2 \in (c_1, 1)$, $\delta > 0$, $t \in [0, \infty)$ e $\mu \in \left(\frac{1}{2}, 1\right)$.
Saída: $\bar{x}$ (solução de (3.1))
Faça $k = 0$ e $p_0 = -\nabla f(x_0)$.
enquanto $\|g_k\| > \varepsilon$ **faça** ▷ *xrepetição*

A.1.2 Encontre α_k tal que:

$$\left.\begin{array}{l} f(x_k + \alpha_k p_k) - f(x_k) \leq c_1 \alpha_k g_k^T p_k \\ g(x_k + \alpha_k p_k)^\intercal p_k \geq c_2 g_k^\intercal p_k \end{array}\right\} \text{Condição Fraca de Wolfe-Powell}$$

A.1.3 Defina $x_{k+1} = x_k + \alpha_k p_k$.
A.1.4 Calcule a direção de busca:

$$p_{k+1} = -g_{k+1} + \beta_k p_k,$$

tal que:

$$\beta_k = \begin{cases} 0, \ se \ \dfrac{s_k^\intercal y_k}{\|p_k\|^2} \leq \delta \\ B_k^{HS} - \min\left\{B_k^{HS}, \dfrac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k\right\} - \dfrac{t g_{k+1}^\intercal s_k}{p_k^\intercal y_k}, \ se \ \dfrac{s_k^\intercal y_k}{\|p_k\|^2} > \delta \end{cases},$$

em que: $B_k^{HS} = \dfrac{g_{k+1}^\intercal y_k}{p_k^\intercal y_k}$, $s_k = x_{k+1} - x_k$ e $y_k = g_{k+1} - g_k$.
A.1.5 Faça $k = k + 1$.
fim enquanto
Faça $\bar{x} = x_k$ ▷ *final* .

Inicialmente, é provado que β_k, definido nos algoritmos 6 e 7, está bem definido, implicando em $\|p_{k+1}\| < +\infty$ para todo $k \geq 0$. Então, formula-se a primeira propriedade:

P1) O valor de β_k está bem definido sempre que a condição de descida for satisfeita.

Prova:

$$\beta_k = \begin{cases} 0, \ se \ \dfrac{s_k^\intercal y_k}{\|p_k\|^2} \leq \delta \\ B_k^{HS} - \min\left\{B_k^{HS}, \dfrac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k\right\} - \dfrac{t g_{k+1}^\intercal s_k}{p_k^\intercal y_k}, \ se \ \dfrac{S_k^\intercal y_k}{\|p_k\|^2} > \delta \end{cases}.$$

Prova-se que $p_k^\intercal y_k \neq 0$.

Primeiramente, observamos que é satisfeita a condição de descida ($p_k^\intercal g_k < 0$), então, temos que $p_k \neq 0 \ \ \forall k$.

Como $\dfrac{s_k^\intercal y_k}{\|p_k\|^2} > \delta > 0 \Rightarrow s_k^\intercal y_k > 0$ (ou seja, a condição de curvatura para os métodos quasi-Newton é satisfeita).

Tendo que $s_k = x_{k+1} - x_k = \alpha_k p_k$ e $\alpha_k > 0$, então $\alpha_k p_k^\intercal y_k > 0$, resultando em $p_k^\intercal y_k > 0$, pois $\alpha_k > 0$. Logo, β_k está bem definida $\forall \ k > 0$. $\square$

Algorithm 7 Algoritmo HS Modificado satisfazendo a condição forte de Wolfe-Powell

Entrada: $x_0 \in \mathbb{R}^n$, $\varepsilon \in (0,1)$, $c_1 \in \left(0, \frac{1}{2}\right)$, $c_2 \in (c_1, 1)$, $\delta > 0$, $t \in [0, \infty)$ e $\mu \in \left(\frac{1}{2}, 1\right)$.
Saída: $\bar{x}$ (solução de (3.1))
Faça $k = 0$, $g_0 = \nabla f(x_0)$ e $p_0 = -g_0$.
enquanto $\|g_k\| > \varepsilon$ **faça** ▷ *xrepetição*

A.1.2* Encontre α_k tal que:

$$\left.\begin{array}{l} f(x_k + \alpha_k p_k) - f(x_k) \leq c_1 \alpha_k g_k^\intercal p_k \\ |g(x_k + \alpha_k p_k)^\intercal p_k| \leq c_2 |g_k^\intercal p_k| \end{array}\right\} \text{Condição Forte de Wolfe-Powell.}$$

A.1.3 Defina $x_{k+1} = x_k + \alpha_k p_k$.
A.1.4 Calcule a direção de busca:

$$p_{k+1} = -g_{k+1} + \beta_k p_k,$$

tal que:

$$\beta_k = \begin{cases} 0, \; se \; \dfrac{s_k^\intercal y_k}{\|p_k\|^2} \leq \delta \\ B_k^{HS} - \min\left\{B_k^{HS}, \dfrac{\mu \|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k\right\} - \dfrac{t g_{k+1}^\intercal s_k}{p_k^\intercal y_k}, \; se \; \dfrac{s_k^\intercal y_k}{\|p_k\|^2} > \delta \end{cases},$$

em que: $B_k^{HS} = \dfrac{g_{k+1}^\intercal y_k}{p_k^\intercal y_k}$, $s_k = x_{k+1} - x_k$ e $y_k = g_{k+1} - g_k$.
A.1.5 Faça $k = k + 1$.
fim enquanto
Faça $\bar{x} = x_k$ ▷ *final* .

Repare que é possível demonstrar a propriedade **(P1)** simplesmente utilizando a condição fraca de Wolfe-Powell, subseção 2.1.2.2. Entretanto, apresenta-se a demonstração acima para enfatizar que o coeficiente de conjugação β_k está bem definido, independente da condição que a busca em linha deve satisfazer.

Posteriormente, assim como provado para o método proposto em (YUAN, 2009), prova-se que a direção de descida, determinada no passo **A.1.4**, atende a condição suficiente de descida.

P2) Seja p_k determinado no passo **A.1.4** pelos algoritmos 6 e 7. Então, existe $c > 0$ tal que

$$g_k^\intercal p_k \leq -c\|g_k\|^2 \quad \forall\, k \geq 0, \tag{3.7}$$

ou seja, a condição suficiente de descida é sempre satisfeita.

Prova:

Se $k = 0$, então $g_0^\intercal d_0 = -\|g_0\|^2$ e (3.7) é satisfeita. Em seguida, prova-se que $g_{k+1}^\intercal p_{k+1}$ satisfaz (3.7) $\forall\, k \geq 0$.

1) Se $\beta_k = 0$, então:

$$g_{k+1}^{\intercal} p_{k+1} = g_{k+1}^{\intercal}(-g_{k+1}) = -g_{k+1}^{\intercal} g_{k+1} = -\|g_{k+1}\|^2.$$

2) Se $\beta_k \neq 0$ e $B_k^{HS} < \dfrac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2} g_{k+1}^{\intercal} p_k$, então:

$$\beta_k = \frac{-t g_{k+1}^{\intercal} s_k}{p_k^{\intercal} y_k}$$

e

$$\begin{aligned} g_{k+1}^{\intercal} p_{k+1} = g_{k+1}^{\intercal} &\left(-g_{k+1} - \frac{t g_{k+1}^{\intercal} s_k}{p_k^{\intercal} y_k} p_k\right) = \\ &= -\|g_{k+1}\|^2 - \frac{t g_{k+1}^{\intercal} s_k}{p_k^{\intercal} y_k}(g_{k+1}^{\intercal} p_k), \qquad (s_k = \alpha_k p_k) \\ &= -\|g_{k+1}\|^2 - \frac{t\alpha_k (g_{k+1}^{\intercal} p_k)^2}{p_k^{\intercal} y_k}, \qquad \left(\alpha_k \in \mathbb{R}^+ \text{ e } p_k^{\intercal} y_k > 0\right) \\ &\leq -\|g_{k+1}\|^2. \end{aligned}$$

3) Se $\beta_k \neq 0$ e $B_k^{HS} \geq \dfrac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2} g_{k+1}^{\intercal} p_k$, então:

$$\begin{aligned} g_{k+1}^{\intercal} p_{k+1} &= -\|g_{k+1}\|^2 + \left[B_k^{HS} - \frac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2}(g_{k+1}^{\intercal} p_k) - t\frac{(g_{k+1} s_k)}{p_k^{\intercal} y_y}\right](p_k^{\intercal} g_{k+1}) = \\ &= -\|g_{k+1}\|^2 + \frac{(g_{k+1}^{\intercal} y_k)(p_k^{\intercal} g_{k+1})}{p_k^{\intercal} y_k} - \frac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2}(g_{k+1}^{\intercal} p_k)^2 - t\frac{(g_{k+1}^{\intercal} s_k)}{p_k^{\intercal} y_k}(p_k^{\intercal} g_{k+1}) = \\ &= -\|g_{k+1}\|^2 + \frac{(g_{k+1}^{\intercal} y_k)(p_k^{\intercal} g_{k+1})}{p_k^{\intercal} y_k} - \frac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2}(g_{k+1}^{\intercal} p_k)^2 - t\frac{\alpha_k (g_{k+1}^{\intercal} p_k)^2}{p_k^{\intercal} y_k} \leq \\ &\leq -\|g_{k+1}\|^2 + \frac{(g_{k+1}^{\intercal} y_k)(p_k^{\intercal} g_{k+1})}{p_k^{\intercal} y_k} - \frac{\mu\|y_k\|^2}{(p_k^{\intercal} y_k)^2}(g_{k+1}^{\intercal} p_k)^2, \qquad \textbf{(*)} \end{aligned}$$

definindo:

$$\tilde{\mu} = \frac{\sqrt{p_k^{\intercal} y_k}}{\sqrt{2\mu}} g_{k+1} \quad \text{e} \quad \sigma = \frac{\sqrt{2\mu}}{\sqrt{p_k^{\intercal} y_k}}(g_{k+1}^{\intercal} p_k) y_k;$$

e por **(*)**, tem-se:

$$g_{k+1}^{\intercal} p_{k+1} \overset{(*)}{\leq} \frac{\tilde{\mu}\sigma - \frac{1}{2}(\|\tilde{\mu}\|^2 + \|\sigma\|^2) + \frac{1}{4}\left(\frac{p_k^{\intercal} y_k \|g_{k+1}\|^2}{\mu}\right) - p_k^{\intercal} y_k \|g_{k+1}\|^2}{p_k^{\intercal} y_k} \leq$$

$$\leq \frac{-\left(1 - \frac{1}{4\mu}\right) p_k^{\intercal} y_k \|g'_{k+1}\|^2}{p_k^{\intercal} y_k} = -\left(1 - \frac{1}{4\mu}\right)\|g_{k+1}\|^2.$$

Já que $\mu \in \left(\frac{1}{2}, 1\right)$, então $c = \left(1 - \frac{1}{4\mu}\right)$.

Observação: Utiliza-se a desigualdade: $\frac{1}{2}(\|\mu\|^2 + \|\sigma\|^2) \geq \mu^\intercal \sigma, \ \forall\, (\mu, \sigma) \in \mathbb{R}^n \times \mathbb{R}^n$.

$\square$

Para as demais propriedades, são adotadas as seguintes hipóteses:

H1) O conjunto $\Omega = \{x \in \mathbb{R}^n; f(x) \leq f(x_0)\}$ é limitado, ou seja:

$$\|x\| \leq M \quad \forall\, x \in \Omega \ \text{ e } \ M > 0.$$

H2) No conjunto $\Omega_0 \subset \mathbb{R}^n$, aberto e convexo, tal que $\Omega \subset \Omega_0$, tem-se:

H2.1) f é inferiormente limitada;

H2.2) f é diferenciável;

H2.3) f' é Lipschitz contínua, ou seja, existe $L > 0$ tal que:

$$\|f'(x) - f'(y)\| \leq L\|x - y\| \quad \forall\, x, y \in \Omega_0.$$

Denota-se $CO\,\Omega$ o envelope convexo de Ω. E, dadas essas hipóteses, tem-se como consequências imediatas:

C1) O fecho de $CO\,\Omega$, denotado por $\overline{CO}\,\Omega$, é fechado e limitado (Proposição 3.2.14 na página 79 de (ISMAILOV; SOLODOV, 2005)), logo compacto.

C2) Se f satisfaz **(H1)** e f' é localmente lipschtziana, então existe um conjunto $\Omega_0 \subset \mathbb{R}^n$, aberto e convexo, tal que $\Omega \subset \Omega_0$, no qual a função f atende a **(H2.1)**, **(H2.2)** e **(H2.3)**. Esta é uma consequência direta de **(C1)**, cuja demonstração é encontrada no Apêndice A.

C3) Supondo **(H1)** e **(H2)** válidas, então existe $\Lambda > 0$ tal que:

$$\|g(x)\| \leq \Lambda \quad \forall\, x \in \Omega.$$

Sendo que ao anunciar **(C3)**, importante para a prova de convergência global da propriedade **(P8)**, definimos $g : \mathbb{R}^n \to \mathbb{R}^n$ como $g(x) = \nabla f(x)$ para todo $x \in \mathbb{R}^n$.

*Prova de **(C3)**:*

Se g não fosse limitada em Ω, então para cada $k > 0$ existiria $x_k \in \Omega$ tal que $\|g(x)\| \geq k$. Fixemos $x \in \Omega$ e definamos $k = (3ML + \|g(x)\|)$. Logo, existirá $y \in \Omega$ satisfazendo:

$$\|g(y)\| \geq k = (3ML) + \|g(x)\|$$

e

$$\begin{aligned} 3ML &\leq \|g(y)\| - \|g(x)\| \\ &\leq |(\|g(y)\| - \|g(x)\|)| \end{aligned}$$

$$
\begin{aligned}
&\leq L\|x-y\| \\
&\leq L(\|x\|+\|y\|) \\
&\leq 2ML
\end{aligned}
$$

$$\Rightarrow 3ML \leq 2ML \Rightarrow 3 \leq 2 \quad \text{(Absurdo!)}$$

□

A propriedade **(P3)** afirma que, sob as hipóteses **(H1)** e **(H2)** a condição de Zoutendijk (ZOUTENDIJK, 1970) é satisfeita, sendo importante para a demonstração de convergência global do método proposto. Essa condição relaciona o comprimento de passo com a direção de busca, pois é necessário que, além de comprimentos de passo bem computados, estejam disponíveis boas direções de busca a fim de obter sucesso na convergência do método de otimização.

P3) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelo algoritmo 6 ou 7. Então, a condição de Zoutendijk é satisfeita

$$\sum_{k=0}^{\infty} \frac{(g_k^{\intercal} p_k)^2}{\|p_k\|^2} < \infty.$$

Prova:

Dos algoritmos propostos, tem-se que:

$$p_k^{\intercal} g_k \leq -c\|g_k\|^2, \quad \forall\, k \geq 0, \ \textit{com}\ c = \left(1 - \frac{1}{4\mu}\right).$$

Então, utilizando o passo **(A.1.2.1)**, temos:

$$f_{k+1} \leq f_k + c_1 \alpha_k p_k^{\intercal} g_k \leq f_k \leq f_{k+1} + c_1 \alpha_{k-1} p_{k-1}^{\intercal} g_{k-1} \leq f_{k-1} \leq \ldots \leq f_0.$$

Ou seja, $\{f_k\} \subset \mathbb{R}$ é limitada uma vez que estamos supondo **(H2.1)** válida.

Dos passos **(A.1.2.2)** ou **(A.1.2*.2)**:

$$g_{k+1}^{\intercal} p_k \geq c_2 g_k^{\intercal} p_k.$$

Subtraindo $g_k^{\intercal} p_k$ dos dois lados:

$$
\begin{aligned}
-(1-c_2) g_k^{\intercal} p_k &\leq (g_{k+1} - g_k)^{\intercal} p_k \\
&\leq \|g_{k+1} - g_k\| \cdot \|p_k\| \\
&\leq L\alpha_k \|p_k\|^2
\end{aligned}
$$

e

$$\alpha_k \geq -\frac{(1-c_2)(g_k^{\intercal} p_k)}{L\|p_k\|^2}$$

Novamente, pelo passo **(A.1.2.2)**

$$f_{k+1} - f_k \leq c_1 \alpha_k g_k^\intercal p_k$$

e

$$f_k - f_{k+1} \geq -c_1 \alpha_k g_k^\intercal p_k \geq \frac{c_1}{L}(1-c_2)\frac{(g_k^\intercal p_k)^2}{\|p_k\|^2}$$

pois $g_k^\intercal p_k < 0$.

De **(H2.1)**, $\exists\, l \in \mathbb{R}$ tal que $l \leq f(x)\ \forall\, x \in \Omega$, e tido que $c_2 \in (c_1, 1)$, temos:

$$f_0 - l \geq f_0 - f_n = \sum_{k=0}^{m-1} f_k - f_{k+1} \geq \frac{c_1(1-c_2)}{L}\sum_{k=0}^{m-1}\frac{(g_k^\intercal p_k)^2}{\|p_k\|^2} \quad \forall\, m \in \mathbb{N}$$

$$\Rightarrow f_0 - l \geq \left(\frac{c_1(1-c_2)}{L}\sum_{k=0}^{m-1}\frac{(g_k^\intercal p_k)^2}{\|p_k\|^2}\right) \quad \forall\, m \in \mathbb{N}$$

Fazendo $m \longrightarrow \infty$, concluímos:

$$\sum_{k=0}^{\infty}\frac{(g_k^\intercal p_k)^2}{\|p_k\|^2} < \infty.$$

□

A seguir é apresentado um resultado análogo ao Teorema 3.2 presente em (YUAN, 2009), certificando que, sob as hipóteses **(H1)**, **(H2)** e a existência de $\alpha* > 0$, tem-se x_k convergindo para um minimizador local de f.

P4) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelos algoritmos 6 ou 7. Se $\alpha_k \geq \alpha* > 0$, então:

$$\lim_{k\to\infty} \|g_k\| = 0.$$

Prova:

Defina $\theta = \dfrac{2M}{\alpha*}$. Logo, temos que:

$$0 < \|\alpha_k p_k\| = \|s_k\| = \|x_{k+1} - x_k\| \leq \|x_{k+1}\| + \|x_k\| \leq 2M$$

e, sendo $\alpha_k \geq \alpha* > 0$, então:

$$0 < \|p_k\| \leq \frac{2M}{\alpha*} = \theta < +\infty \quad \forall\, k > 0. \quad \textbf{(*)}$$

De **(P2)** (Condição suficiente):

$$c\|g_k\|^2 \leq -g_k^\intercal p_k$$

temos que:

$$\frac{c^2\|g_k\|^4}{\|p_k\|^2} \leq \frac{(g_k^\intercal p_k)^2}{\|p_k\|^2}.$$

Além disso,

$$\sum_{k\geq 0} c^2\frac{\|g_k\|^4}{\|p_k\|^2} \leq \sum_{k\geq 0}\frac{(g_k^\intercal p_k)^2}{\|g_k\|^2} < +\infty. \quad \textbf{(**)}$$

De **(*)** e **(**)**, temos

$$\frac{c^2}{\theta^2}\sum_{k\geq 0}\|g_k\|^4 < \infty,$$

e, consequentemente,

$$\lim_{k\to\infty}\|g_k\|^4 = 0 \Rightarrow \lim_{k\to\infty}\|g_k\| = 0.$$

□

As propriedades **(P5)** e **(P6)** mostradas a seguir são lemas técnicos, no sentindo que serão utilizados como alicerce principal para as demonstrações das propriedades **(P7)** e **(P8)**.

P5) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelos algoritmos 6 ou 7. Se $\|g_k\| \geq \delta > 0 \,\forall\, k > 0$, então $\exists K > 0$ tal que:

$$\|p_k\| \geq K\delta.$$

Além disso, independente de $\|g_k\|$ ser limitada inferiormente por uma constante positiva, temos:

$$\beta_k^2 \leq \frac{\|p_{k+1}\|^2}{\|p_k\|^2}.$$

Prova:

$$p_{k+1} = -g_{k+1} + \beta_k p_k \implies p_{k+1} + g_{k+1} = \beta_k p_k$$

e

$$(p_{k+1} + g_{k+1})^\intercal (p_{k+1} + g_{k+1}) = \beta_k^2\|p_k\|^2$$

$$\begin{aligned}
\|p_{k+1}\|^2 &= -\|g_{k+1}\|^2 - 2g_{k+1}^\intercal p_{k+1} + \beta_k^2\|p_k\|^2 \\
&\overset{(1)}{\geq} -\|g_{k+1}\|^2 + 2c\|g_{k+1}\|^2 + \beta_k^2\|p_k\|^2 \\
&\geq (2c-1)\|g_{k+1}\|^2 + \beta_k^2\|p_k\|^2 \\
&\geq (2 - \frac{1}{2\mu} - 1)\|g_{k+1}\|^2 + \beta_k^2\|p_k\|^2. \quad (*)
\end{aligned}$$

Em (1) utilizamos a condição suficiente de descida. Como $\mu \in \left(\frac{1}{2}, 1\right)$:

$$\|p_{k+1}\|^2 \geq \left(1 - \frac{1}{2\mu}\right)\|g_{k+1}\|^2 \geq \left(1 - \frac{1}{2\mu}\right)\delta^2,$$

e definindo $K = \left(1 - \dfrac{1}{2\mu}\right)^{\frac{1}{2}}$ implica em:

$$\|p_{k+1}\| \geq K\delta.$$

Por outro lado, por (*) temos:

$$\|p_{k+1}\|^2 \geq \beta_k^2 \|p_k\|^2 \implies \beta_k^2 \leq \frac{\|p_{k+1}\|^2}{\|p_k\|^2} \quad \text{P.5.1.}$$

□

P6) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelo algoritmo 7. Então, temos:

$$\liminf \|g_k\| = 0$$

ou

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|p_k\|^2} < \infty.$$

Prova:

Suponhamos que:

$$\liminf \|g_k\| \neq 0; \tag{3.8}$$

e

$$\sum_{k \geq 0} \frac{\|g_k\|^4}{\|p_k\|^2} = \infty. \tag{3.9}$$

De (3.8), existe $\delta > 0$ tal que

$$\|g_k\| \geq \delta > 0 \quad \forall\, k \in \mathbb{N}$$

e, com base no passo **(A.1.4)**, $p_{k+1} = g_{k+1} + \beta_k p_k$ assim,

$$\begin{aligned}
\|g_{k+1}\|^2 &= |p_{k+1}^\intercal p_{k+1} - \beta_k p_k^\intercal g_{k+1}| \\
&\leq |g_{k+1}^\intercal p_{k+1}| + |\beta||p_k^\intercal g_{k+1}| \\
&\leq |g_{k+1}^\intercal p_{k+1}| - c_2|\beta_k|p_k^\intercal g_k \\
&\leq |g_{k+1}^\intercal p_{k+1}| + c_2|\beta_k|p_k^\intercal g_k
\end{aligned}$$

Da desigualdade $(m + \delta n)^2 \leq (1+\delta)^2(m^2 + n^2) \quad \forall\, a, b, \delta \geq 0$, resulta:

$$\begin{aligned}
\|g_{k+1}\|^4 &\leq (|g_{k+1}^\intercal p_{k+1}| + c_2|\beta_k||g_{k+1}^\intercal p_k|)^2 \\
&\leq (1+c_2)^2(|g_{k+1}^\intercal p_{k+1}|^2 + |\beta_k|^2|g_k^\intercal p_k|^2) \\
&\overset{(*)}{\leq} (1+c_2)^2\left(|g_{k+1}^\intercal p_{k+1}|^2 + \frac{\|p_{k+1}\|^2}{\|p_k\|^2}|g_k^\intercal p_k|^2\right) \\
&\leq (1+c_2)^2\left(\frac{|g_{k+1}^\intercal p_{k+1}|^2}{\|p_{k+1}\|^2} + \frac{|g_k^\intercal p_k|^2}{\|p_k\|^2}\right)\|p_{k+1}\|^2.
\end{aligned}$$

Em (*), presente na desigualdade acima, utilizamos a segunda afirmação da Propriedade **(P5)**. Assim,

$$\frac{\|g_{k+1}\|^4}{\|p_{k+1}\|^2} \leq (1+c_2)^2 \left(\frac{|g_{k+1}^\intercal p_{k+1}|^2}{\|p_{k+1}\|^2} + \frac{|g_k^\intercal p_k|^2}{\|p_k\|^2} \right)$$

e, tomando o somatório juntamente com a hipótese (3.9), obtemos:

$$+\infty = \sum_{k\geq 1} \frac{\|g_{k+1}\|^4}{\|p_{k+1}\|^2} \leq 2(1+c_2)^2 \sum_{k\geq 1} \frac{|g_k^\intercal p_k|^2}{\|p_k\|^2}$$

o que implica:

$$\sum_{k\geq 1} \frac{|g_k^\intercal p_k|^2}{\|p_k\|^2} = \infty.$$

o que contraria **(P3)**.

□

A propriedade **(P7)** a seguir é simplesmente um corolário da propriedade **(P6)**.

P7) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelo algoritmo 7. Se

$$\sum_{k\geq 1} \frac{1}{\|p_k\|^2} = \infty$$

então,

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Prova: Se $\liminf_{k\to\infty} \|g_k\| \neq 0$, então existe $\varepsilon > 0$ tal que $\|g_k\| \geq \varepsilon$, para todo $k \geq 0$. Da Propriedade 6, resulta:

$$\frac{1}{\|p_k\|^2} \leq \frac{1}{\varepsilon^4} \frac{\|g_k\|^4}{\|p_k\|^2} \quad \forall\, k \geq 0, \text{ pois } \frac{\|g_k\|}{\varepsilon} \geq 1$$

e

$$\sum_{k\geq 0} \frac{1}{\|p_k\|} \leq \frac{1}{\varepsilon^4} \sum_{k\geq 0} \frac{\|g_k\|^4}{\|p_k\|^2} < +\infty.$$

Logo, pela contra-positiva, provamos:

$$\sum_{k\geq 0} \frac{1}{\|p_k\|^2} = +\infty.$$

□

Por último, a propriedade seguinte afirma que existirá uma subsequência de g_k convergindo para o vetor nulo.

P8) Suponha-se que **(H1)** e **(H2)** sejam satisfeitas e que as sequências $\{g_k\}$, $\{p_k\}$ e $\{\alpha_k\}$ sejam geradas pelo algoritmo 7. Então,

$$\sum_{k\geq 1} \frac{1}{\|p_k\|^2} = \infty.$$

Por consequência, da Propriedade **(P7)**:

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Prova: Suponha $\sum_{k\geq 1} \frac{1}{\|p_k\|^2} < \infty$, então:

$$\begin{aligned}\lim_{k\to\infty} \frac{1}{\|p_k\|^2} = 0 &\implies \lim_{k\to\infty} \|p_k\|^2 = +\infty \\ &\implies \lim_{k\to\infty} \|p_k\| = +\infty.\end{aligned}$$

ou seja, $\{p_k\}$ é uma sequência ilimitada.
Utilizando, na passagem (*), o fato de $\|g(x)\|$ ser limitado e de $\alpha_k \leq \alpha_{\max} < \infty\ \forall k \geq 0$, devido as hipóteses **(H1)** e **(H2.1)**, no passo (**), segue:

$$\begin{aligned} 0 \leq \frac{|s_k^\intercal y_k|}{\|p_k\|^2} &= \frac{|s_k^\intercal (g_{k+1} - g_k)}{\|p_k\|^2} \\ &\leq \frac{\|s_k\|\|g_{k+1} - g_k\|}{\|p_k\|^2} \\ &\overset{(*)}{\leq} \frac{2\Lambda\|s_k\|}{\|p_k\|^2} \\ &\leq \frac{2\alpha_k\Lambda\|p_k\|}{\|p_k\|^2} \\ &\overset{(**)}{\leq} \frac{2\alpha_{\max}\Lambda}{\|p_k\|}, \end{aligned}$$

e para $\|p_k\|$ suficientemente grande, temos

$$\frac{s_k^\intercal y_k}{\|p_k\|^2} \leq \frac{2\alpha_{\max}\Lambda}{\|p_k\|} < \delta \implies p_{k+1} = -g_{k+1}$$

e

$$\|p_{k+1}\| = \|g_{k+1}\| < \Lambda.$$

Portanto, elementos de $\{p_k\}$ com norma maior que Λ estariam sendo gerados por

$$p_{k+1} = -g_{k+1} + \beta_k p_k,$$

com

$$\beta_k = B_k^{HS} - \min\left\{B_k^{HS}, \frac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k\right\} - t\frac{g_{k+1}^\intercal s_k}{p_k^\intercal \gamma_k}$$

e

$$\frac{s_k^\intercal y_k}{\|p_k\|^2} > \delta,$$

entretanto, satisfazer a desigualdade acima implica em:

$$\frac{2\alpha_{\max}\Lambda}{\|p_k\|} > \delta \implies \|p_k\| < \frac{2\alpha_{\max}\Lambda}{\delta}.$$

Assim sendo, de $p_{k+1} = g_{k+1} + \beta_k p_k$ tem-se:

$$\|p_k\| < \Lambda + |\beta_k|\|p_k\|. \tag{3.10}$$

Se $B_k^{HS} < \dfrac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k$, da desigualdade acima e:

$$\begin{cases} s_k^\intercal y_k > \delta\|p_k\|^2; \\ 0 < c\|g_k\| \leq \|p_k\|; \qquad \text{(condição suficiente de descida (P3))} \\ \alpha_k \in (0, \alpha_{\max}); \end{cases}$$

obtemos:

$$\alpha_k p_k^\intercal y_k > \delta\|p_k\|^2 \implies p_k^\intercal y_k > \delta\frac{\|p_k\|^2}{\alpha_k} \implies \frac{1}{p_k^\intercal y_k} \leq \frac{\alpha_k}{\delta\|p_k\|^2}$$

e

$$|\beta_k| = \frac{t|s_k^\intercal g_{k+1}|}{|p_k^\intercal y_k|} \leq \frac{t\alpha_k}{\delta\|p_k\|^2}|s_k^\intercal g_{k+1}| \leq \frac{t\alpha_k^2\Lambda}{\delta\|p_k\|} \leq \frac{t\alpha_{\max}^2\Lambda}{\delta\|p_k\|}.$$

Substituindo em (3.10) e sabendo que $t < \infty$ é uma constante positiva:

$$\|p_{k+1}\| \leq \Lambda + \frac{t\alpha_{\max}^2\Lambda}{\delta\|p_k\|}\|p_k\| \leq \Lambda + \frac{t\alpha_{\max}^2\Lambda}{\delta} < +\infty.$$

Se $B_k^{HS} \geq \mu\dfrac{\|y_k\|^2}{(p_k^\intercal y_k)^2} g_{k+1}^\intercal p_k$, então:

$$|B_k^{HS}| = \frac{|g_{k+1}^\intercal y_k|}{p_k^\intercal y_k} \leq \frac{\alpha_k}{\delta\|p_k\|^2}\|g_{k+1}^\intercal\|\|y_k\| \leq \frac{\alpha_k^2 L\Lambda\|p_k\|}{\delta\|p_k\|^2} \leq \frac{\alpha_k^2 L\Lambda}{\delta\|p_k\|}$$

e já é sabido que:

$$\frac{|tg_{k+1}^\intercal s_k|}{p_k^\intercal y_k} \leq \frac{t\alpha_{\max}^2\Lambda}{\delta\|p_k\|} < +\infty.$$

Por último, tem-se:

$$\frac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2}|g_{k+1}^\intercal p_k| \leq \frac{\alpha_k^2}{\delta^2\|p_k\|^4}\mu L^2\alpha_k^2\|p_k\|^2 M\|p_k\| = \frac{\alpha_k^4\mu L^2 M}{\delta^2\|p_k\|}$$

Portanto:

$$\begin{aligned} \|p_{k+1}\| &\leq \Lambda + \left(|B_k^{HS}| + \frac{\mu\|y_k\|^2}{(p_k^\intercal y_k)^2}|g_{k+1}^\intercal p_k| + \frac{t|g_{k+1}^\intercal s_k|}{(p_k y_k)}\right)\|p_k\| \\ &\leq \Lambda + \frac{\alpha_{\max}^2 L\Lambda}{\delta} + \frac{\alpha_{\max}^4\mu L^2 M}{\delta^2} + \frac{t\alpha_{\max}^2\Lambda}{\delta} < +\infty, \end{aligned}$$

isto é, $\|p_{k+1}\|$ é sempre limitado, contradizendo o fato de $\{p_k\}$ ser ilimitado. Logo:

$$\sum \frac{1}{\|p_k\|^2} = \infty \implies \liminf_{t\to\infty}\|g_k\| = 0.$$

□

Apresentadas as principais propriedades dos algoritmos 6 e 7 propostos, é realizada, na próxima seção 3.3, uma comparação entre os métodos de gradiente conjugado abordados neste trabalho, de forma a evidenciar as semelhanças e diferenças entre eles.

Tabela 1 – Expressões para o coeficiente de conjugação β_k.

Método	Coeficiente de Conjugação
GC-HS	$\beta_k^{HS} = \dfrac{g_k^\intercal y_{k-1}}{p_{k-1}^\intercal y_{k-1}}$
GC-PR	$\beta_k^{PR} = \dfrac{g_k^\intercal y_{k-1}}{g_{k-1}^\intercal g_{k-1}}$
GC-DL	$\beta_k^{DL} = \beta_k^{HS} - t\dfrac{g_k^\intercal s_{k-1}}{p_{k-1}^\intercal y_{k-1}}$
GC-GY	$\beta_k^{\mathrm{GY}} = \beta_k^{PR} - \min\left\{\beta_k^{PR}, \dfrac{\mu\|y_{k-1}\|^2}{\|g_{k-1}\|^4} g_k^\intercal p_{k-1}\right\}$
GC-HS*	$\beta_k^{HS^*} = \begin{cases} 0, \; se \; \dfrac{s_{k-1}^\intercal y_{k-1}}{\|p_{k-1}\|^2} \leq \delta \\ \beta_k^{HS} - \min\left\{\beta_k^{HS}, \dfrac{\mu\|y_{k-1}\|^2}{(p_{k-1}^\intercal y_{k-1})^2} g_k^\intercal p_{k-1}\right\} - \dfrac{t g_k^\intercal s_{k-1}}{p_{k-1}^\intercal y_{k-1}}, \; se \; \dfrac{S_{k-1}^\intercal y_{k-1}}{\|p_{k-1}\|^2} > \delta \end{cases}$

3.3 Discussão Teórica

Lembrado que a estrutura geral dos métodos de gradiente conjugado é:

$$
\begin{aligned}
x_{k+1} &= x_k + \alpha_k p_k; \\
p_k &= \begin{cases} -g_k, & se \; k = 0 \\ -g_k + \beta_k p_{k-1}, & se \; k \geq 1 \end{cases};
\end{aligned}
\tag{3.11}
$$

para todo $k \geq 0$, com o coeficiente de conjugação β_k sendo definido de acordo com a tabela 1.

De imediato, ao analisar a tabela 1, percebe-se que as fórmulas dos coeficientes de conjugação são acrescidas com termos e expressões matemáticas que buscam agregar características específicas ao métodos bases, neste caso: GC-HS (seção 2.2.1) e GC-PR (seção 2.2.2).

Deve-se pontuar que, para o otimização em alta dimensão, acrescentar expressões matemáticas é sinônimo de incremento considerável no tempo de processamento computacional, pois mais operações aritméticas são necessárias. Além disso, deve-se tomar cuidado com os termos que estão sendo adicionados às expressões matemáticas, uma vez que podem ocasionar problemas de falta de espaço de memória física do computador. Portanto, há uma relação de equilíbrio que deve ser, a todo momento, levada em consideração na construção de métodos numéricos para otimização em alta dimensão.

Seguidamente, é observado que os métodos GC-HS e GC-DL só estão bem-definidos ($|\beta_k| < \infty$) quando a busca linear satisfizer, no mínimo, as condições fracas de Wolfe-Powell, subseção 2.1.2.2. Entretanto, como visto na propriedade **(P1)**, o método proposto, o método do gradiente conjugado HS modificado, GC-HS*, está sempre bem definido, independentemente da condição que a busca linear deva satisfazer. Por causa disso, utiliza-se no método desenvolvido, assim como no método do GC-DL, um termo extra o qual considera que a condição de conjugação do método GC-HS não é identicamente nula.

Outra importante vantagem do método do GC-HS*, em relação aos métodos GC-HS, GC-DL e GC-PR, é que o método proposto sempre irá satisfazer a condição suficiente de descida, propriedade **(P2)**, assim como o método do GC-GY, descrito na seção 3.1.

Por fim, como ponto principal deste trabalho, prova-se, sob as hipóteses **(H1)**, **(H2)** e utilizando a busca linear satisfazendo a condição forte de Wolfe-Powell (subseção 2.1.2.3), na propriedade **(P8)**, a convergência global do método GC-HS*. É necessário enfatizar que a convergência global demonstrada para o método apresentado no presente trabalho, GC-HS*, é independente da condição imposta por (YUAN, 2009) de $\inf\{\alpha_k\} > 0$, necessária para convergência global do método GC-GY. Destaca-se ainda, que não há, sob as hipóteses referidas, prova de convergência global para os métodos GC-HS, GC-PR e GC-DL.

Portanto, buscou-se com o método do gradiente conjugado HS modificado, GC-HS*, aperfeiçoar o método base do GC-HS, adicionando as principais características citadas acima, oriundas dos outros métodos de gradiente conjugado estudados.

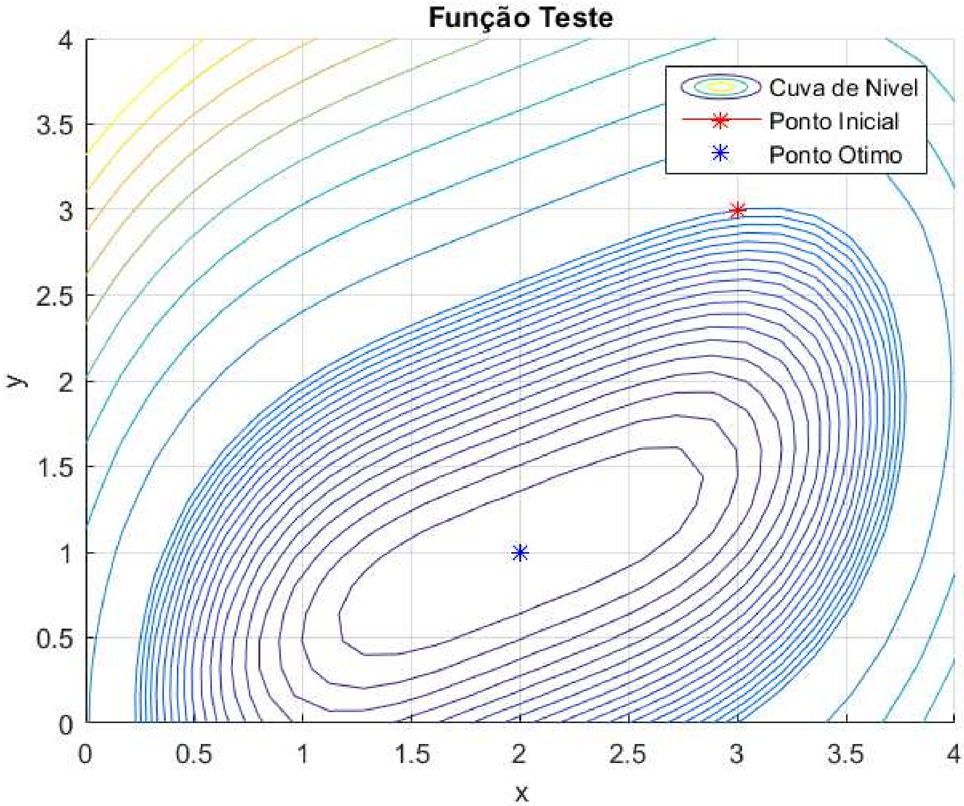
Interpretação Geométrica dos Métodos de Gradiente Conjugado

A fim de aferir uma interpretação geométrica do método do gradiente conjugado HS modificado (CG-HS*) em relação aos demais métodos: GC-HS, GC-PR, GC-DL e GC-GY; foi criada uma função teste f, definida por:

$$f = (x-2)^4 + (x-2y)^2; \tag{3.12}$$

em que é pretendido encontrar o seu mínimo local. As curvas de nível desta função teste são apresentadas na figura 5, assim como o seu ponto de mínimo (2, 1). Para fins de comparação, utilizou-se o mesmo ponto inicial (3, 3) para todos os métodos de gradiente conjugado, sendo também destacado nessa figura 5, salientando que as configurações paramétricas dos métodos são tratadas na seção 4.3.

Figura 5 – Curvas de nível da função teste $f = (x-2)^4 + (x-2y)^2$, que possui o mínimo no ponto (2, 1), representado por meio do asterisco azul (*). Todos os métodos de gradiente conjugado foram iniciados no mesmo ponto (3,3), representado por meio do asterisco vermelho (*).



Fonte: figura do autor.

Na figura 6a são ilustrados os passos (caminhos) gerados pelos métodos do gradiente conjugado até o ponto de mínimo (2, 1), acentuando que o número de iterações até este ponto é de:

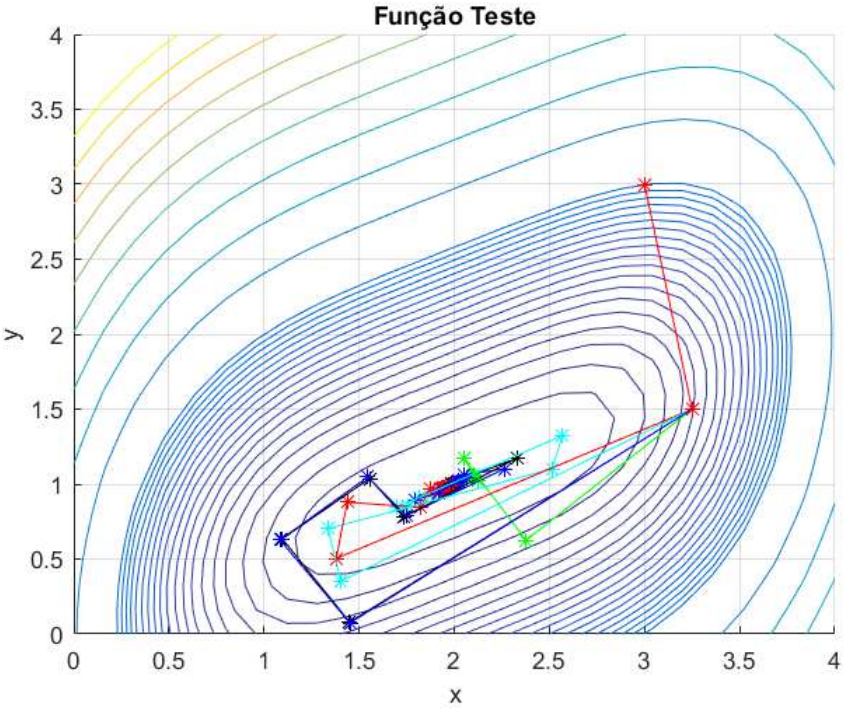
- 23 iterações, para o método GC-HS;
- 27 iterações, para o método GC-PR;
- 19 iterações, para o método GC-DL;
- 18 iterações, para o método GC-GY;
- 18 iterações, para o método GC-HS*, proposto através do algoritmo 6.

Já a figura 6b apresenta as iterações dos métodos citados na vizinhança do ponto de mínimo local (2, 1) da função f, descrita conforme equação (3.12).

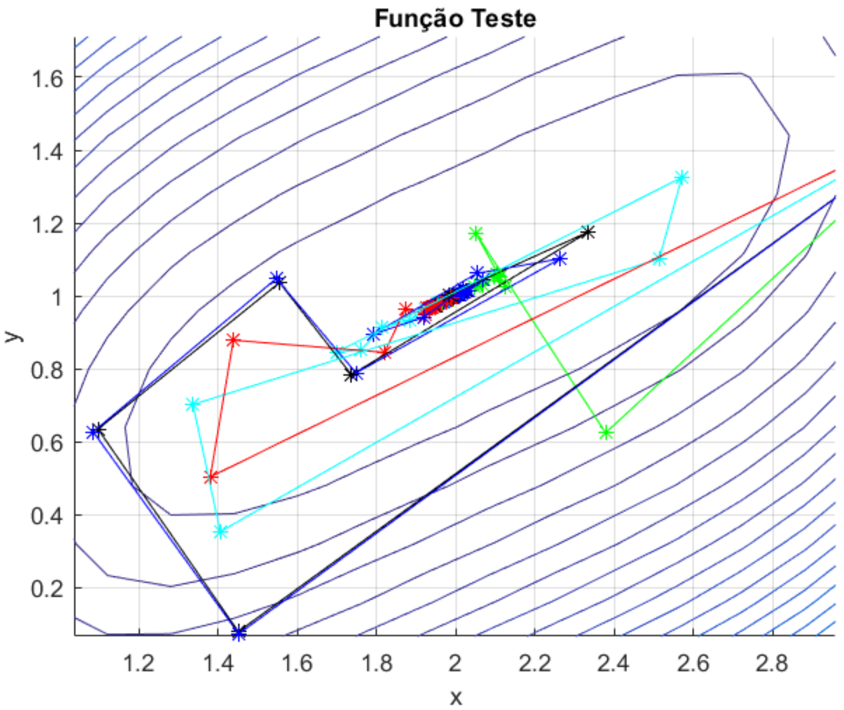
Assim sendo, os caminhos gerados pelos métodos de gradiente conjugado, figura 6, apresentam comportamentos distintos. Uma análise mais detalhada é realizada no próximo capítulo para problemas de otimização irrestrita e suave em alta dimensão.

Figura 6 – Interpretação geométrica dos passos (caminhos) gerados pelos métodos do gradiente conjugado HS Modificado, GY, DL, HS e PR, utilizados neste trabalho, tendo como função teste $f = (x-2)^4 + (x-2y)^2$, representada por meio das curvas de nível da figura 5.

(a) Passos gerados pelos métodos de gradiente conjugado até o ponto de mínimo local (2,1) da função teste f, representado na figura 5.



(b) Passos dos métodos de gradiente conjugado próximos ao ponto de mínimo (2, 1) da função teste f, representado na figura 5.



Fonte: figura do autor.

4 Experimentos Numéricos

São apresentados nesse capítulo os resultados dos experimentos numéricos a fim de validar o desempenho computacional do método proposto, o gradiente conjugado HS modificado (GC-HS*), em relação a outros métodos para problemas de otimização em alta dimensão. Ressalta-se que, neste texto, o desempenho computacional é medido pelos seguintes indicadores:

- a capacidade de encontrar o mínimo da função objetivo f, ou seja, solucionar o problema (3.1), quantificado através da variável robustez;
- o número de iterações até a solução do problema (3.1);
- o tempo de processamento computacional até a solução do problema (3.1).

Com a finalidade de quantificar essas informações para fins comparativos, utilizou-se a técnica de análise do *performance profile* (MORÉ; GARBOW; HILLSTROM, 1981).

Então, este capítulo está organizado em quatro seções, sendo a primeira, seção 4.1, dedicada a construção do conjunto de problemas $\mathscr{P}$, que possui o objetivo de representar a classe de problemas de otimização suave irrestrita em alta dimensão. A seção 4.2 apresenta a técnica de análise do *performance profile*, que permitirá tecer comparações entre os métodos de gradiente conjugado escolhidos para comparação. Esta análise é realizada na seção 4.3. Na última seção, seção 4.4, é efetuado uma comparação do métodos de gradiente conjugado, incluindo o método proposto, com o método de região de confiança, presente na biblioteca de otimização do software $Matlab$, utilizando a técnica de análise do *Performance Profile*.

4.1 Problemas do Conjunto $\mathscr{P}$

Os problemas de otimização irrestritos selecionados foram retirados do banco de problemas da *CUTest* (GOULD; ORBAN; TOINT, 2015), sendo as características de alguns dos seus problemas citadas ao final dessa seção. Assim, formou-se o conjunto desses problemas $\mathscr{P}$, listados na tabela 3, em que se altera a classificação do problema, tabela 2, com a sua dimensão. Destaca-se que o objetivo central é avaliar o desempenho computacional do método proposto (GC-HS*) em problemas de otimização em alta dimensão.

A seguir, são apresentados os problemas de otimização suave irrestrita selecionados para construção do conjunto $\mathscr{P}$. Destaca-se que todos os problemas escolhidos possuem a função objetivo f, definida como:

$$f(x) = \sum_{i=0}^{n} f_i^2(x), \tag{4.1}$$

Tabela 2 – Classificação dos Problemas do conjunto $\mathscr{P}$

Nomes dos problemas do conjunto $\mathscr{P}$	
ID do Problema	Nome do Problema
A	argtrig-pág. 54
B	bdarwhd
C	brownal
D	broyden3d
E	dqrtic
F	eg2
G	integreq
H	nlminsurf
I	powellsg-pág. 55
J	powr
K	rosenbr-pág. 54
L	tquartic
M	tridia-pág. 54
N	vardim
O	arwhead
P	broydenbd
Q	crglvy
R	dixon
S	genhumps
T	indef
U	helix

Tabela 3 – Enumeração dos problemas do conjunto $\mathscr{P}$, tendo como referência de classificação do problema a tabela 2.

Conjunto de Problemas $\mathscr{P}$		
Número do Problema	ID do Problema	Dimensão
1 a 14	Problemas A a N	400
15	Problema M	500
16 a 29	Problemas A a N	900
30	Problema M	1000
31 a 44	Problemas A a N	1600
45	Problema M	2000
46 a 66	Problemas A a U	2500
67 a 69	Problemas K, M, P	5000
70 a 72	Problemas I, K, M	15000
73	Problema M	30000
74 e 75	Problemas I e K	150000
76 a 77	Problemas I e K	300000
78	Problema I	1000000

em que n é a dimensão do problema e $f_i : \mathbb{R}^n \to \mathbb{R}$ é estabelecida de acordo com o problema de otimização que está sendo utilizado.

Função Trigonométrica

A função trigonométrica, problema 26 de (MORÉ; GARBOW; HILLSTROM, 1981), é definida por:

$$f_i(x) = n - \sum_{j=0}^{n} cos(x_j) + i(1 - cos(x_i)) - sen(x_i); \tag{4.2}$$

em que $x = (x_1, ..., x_n)$.

O ponto inicial é definido como $x_0 = (1/n, ..., 1/n) \in \mathbb{R}^n$, sendo que f, conforme a expressão (4.1), atinge o valor mínimo em 0.

Função de Broyden

Também conhecida como função tridiagonal de Broyden, problema 30 de (MORÉ; GARBOW; HILLSTROM, 1981) é definida por:

$$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1; \tag{4.3}$$

em que $x_0 = x_{n+1} = 0$.

Destaca-se que:

- o ponto inicial é $x_0 = (-1, ..., -1) \in \mathbb{R}^n$;
- o valor mínimo de f, definido conforme a expressão (4.1), é 0.

Função de Rosenbrock

Utiliza-se a versão de Rosenbrock em alta dimensão, denominada função de Rosenbrock Estendida, problema 21 de (MORÉ; GARBOW; HILLSTROM, 1981), definida por:

$$f_{2i-1}(x) = 10(x_{2i} - x_{i-1}^2),$$
$$f_{2i} = 1 - x_{2i-1}.$$

Pela própria definição acima, a dimensão do problema n deve ser um número par.

Ressalta-se:

- o ponto inicial é $x_0 = (x_1^0, ..., x_{n-1}^0, x_n^0) \in \mathbb{R}^n$, com $x_{2i-1}^0 = 1,2$ e $x_{2i}^0 = 1$;
- o valor mínimo de f, definido conforme a expressão (4.1), ocorre na origem, sendo igual a 0.

A função de Rosenbrock possui uma peculiaridade interessante, para $n = 2$ a função é unimodal e para valores superiores a 2, multimodal.

Função de Powell

Denominada em (MORÉ; GARBOW; HILLSTROM, 1981), problema 22, por função Estendida de Powell, é definida por:

$$\begin{aligned} f_{4i-3}(x) &= x_{4i-3} + 10x_{4i-2}, \\ f_{4i-2}(x) &= \sqrt{5}\,(x_{4i-1} - x_{4i}), \\ f_{4i-1}(x) &= (x_{4i-2} - 2x_{4i-1})^2, \\ f_{4i} &= \sqrt{10}\,(x_{4i-3} - x_{4i})^2. \end{aligned}$$

para dimensão n múltipla de 4.

Destaca-se:

- o ponto inicial é $x_0 = (x_1^0, ..., x_{n-1}^0, x_n^0) \in \mathbb{R}^n$, com $x_{4i-3}^0 = 3$, $x_{4i-2}^0 = -1$, $x_{4i-1}^0 = 0$ e $x_{4i}^0 = 1$;
- o valor mínimo de f, definido conforme a expressão (4.1), é igual a 0 na origem.

4.2 Perfil de Performance (*Performance Profile*)

O *Performance Profile* é uma técnica de análise, proposta por Dolan e Moré (DOLAN; MORÉ, 2002), que permite comparar o desempenho computacional de um conjunto de métodos de otimização (algoritmos), $\mathscr{A}$, sobre um conjunto de problemas, $\mathscr{P}$, a fim de determinar o melhor método (algoritmo) para aquela classe de problemas. Dessa forma, procura-se formar o conjunto $\mathscr{P}$ com problemas de otimização que possuam certas características em comum, a fim de evidenciar o desempenho computacional dos métodos de otimização.

Assim sendo, ao surgir um novo problema de otimização que não pertença ao conjunto $\mathscr{P}$, mas que possua as mesmas características dos problemas desse conjunto, é esperado que o método (algoritmo) definido com respaldo do *Performance Profile* seja capaz de solucioná-lo com bom desempenho computacional. Como figura de mérito para avaliar esse desempenho computacional, pode-se escolher o número de iterações, número de avaliações da função f ou o tempo de processamento computacional para solucionar o problema de otimização.

Por exemplo, deseja-se comparar m métodos de otimização, i.e. $\mathscr{A} = \{A_1, ..., A_m\}$, sobre um conjunto $\mathscr{P}$ com n problemas de otimização, i.e. $\mathscr{P} = \{P_1, ..., P_n\}$, definindo como critério de desempenho o número de iterações. Primeiramente, obtém-se $I_{P_j}^{A_i}$, que é o número de iterações necessárias para solucionar o problema P_j utilizando o método A_i. Se o método A_{i^*} não soluciona P_{j^*}, então $I_{P_{j^*}}^{A_{i^*}} = I_M$, que é número máximo de iterações permitido, utilizado como critério de parada do método de otimização. No exemplo, $i \in \{1, ..., m\}$ e $j \in \{1, ..., n\}$.

Em seguida, para cada problema P_j, define-se $I_{P_j}^m$ como o menor número de iterações dentre todos os métodos de $\mathscr{A}$ e dividem-se os elementos $I_{P_j}^{A_i}$ com $i \in \{1, ..., m\}$ por este valor, obtendo r_j^i. Logo, ao melhor método sempre é atribuído o valor 1 e aos demais, valores proporcionais. Assim, procede-se para todos os problemas do conjunto $\mathscr{P}$, sendo possível, ao final do processo, determinar:

- a eficiência dos métodos;
- a robustez dos métodos.

A eficiência do método A_i é o número de problemas do conjunto $\mathscr{P}$ no qual foi atribuído valor 1, i.e. $r_j^i = 1$ para $j \in \{1, ..., n\}$, dividido pela cardinalidade desse conjunto $\mathscr{P}$. Por sua vez, a robustez do método é o número máximo problemas de $\mathscr{P}$ solucionados por esse, sendo definida ao se tomar o extremo direito dos gráficos de *performance profile*.

Tais gráficos são funções cumulativas, definidas para cada método A_i como:

$$\rho_{A_i}(\tau) = \frac{1}{|\mathscr{P}|} |\{p \in \mathscr{P}; r_j^i \leq \tau\}|; \tag{4.4}$$

em que $|\cdot|$ denota a cardinalidade de seu argumento. Desse modo, a função $\rho_{A_i} : \mathbb{R} \to [0, 1]$ está bem definida.

4.3 Resultados e Análises

Os métodos de gradiente conjugado não linear utilizados para comparação foram:

- GC-HS, seção 2.2.1;
- GC-PR, seção 2.2.2;
- GC-DL, seção 2.2.3;
- GC-GY, seção 3.1;
- GC-HS*, seção 3.2;

sendo os seus parâmetros configurados conforme tabela 4, em que:

- $c_1 \in (0, 1)$, é a constante de Armijo, presente na condição de Armijo da desigualdade (2.5);
- $c_2 \in (c_1, 1)$, é a constante de curvatura, presente na condição de curvatura da desigualdade (2.7);

- $t \geq 0$, é a constante presente nos algoritmos GC-DL e GC-HS*, que considera a condição de conjugação (2.15) não identicamente nula, estando inserida na equação (2.24) e no cômputo do coeficiente de conjugação $\beta_k^{HS^*}$, passo **A.1.4** do algoritmo 6;

- $\mu \in \left(\frac{1}{2}, 1\right)$, é a constante para a condição suficiente de descida, presente no cálculo dos coeficientes de conjugação $\beta_k^{HS^*}$ e β_k^{GY}, algoritmos 5 (passo **Y.1.4**) e 6 (passo **A.1.4**), respectivamente;

- $\delta \geq 0$, é a constante utilizada para reiniciar o cômputo das direções conjugadas do método GC-HS*, presente no passo **A.1.4** do algoritmo 6;

- $\varepsilon > 0$, é a precisão de $\|g_k\|$, utilizado como critério de parada. Se $\|g_k\| < \varepsilon$, considera-se que o algoritmo convergiu para um ponto suficientemente próximo do ponto estacionário da função objetivo f;

- Max_iter_gc> 0, é o número máximo de iterações, utilizado como critério de parada, para os métodos de gradiente conjugado;

- Max_iter_ls> 0, é o número máximo de iterações, utilizado como critério de parada, para a busca linear satisfazendo a condição fraca de Wolfe-Powell, sendo implementada computacionalmente conforme algoritmo 2.

Tabela 4 – Parâmetros dos Algoritmos GC

Parâmetros	Valor
ε	$1{,}00E^{-03}$
c_1	$1{,}00E^{-04}$
c_2	$1{,}00E^{-01}$
t	$4{,}00E^{+00}$
μ	$5{,}60E^{-01}$
δ	$5{,}00E^{-03}$
Max_iter_gc	20000
Max_iter_ls	$5{,}00E^{+02}$

Ressalta-se novamente que o método de busca linear utilizado procura atender a condição fraca de Wolfe-Powell, sendo implementado conforme o algoritmo 2. Porém, destaca-se que a condição forte de Wolfe-Powell, subseção 2.1.2.2, é fundamental para a análise de convergência global demonstrada na propriedade **(P8)**. O que justificou essa troca de método de busca linear foi o elevado tempo computacional requerido pelos métodos gradiente conjugado quando utilizados em conjunto com a busca linear satisfazendo a condição forte de Wolfe-Powell, implementado conforme algoritmo 3. Esse acréscimo de tempo substancial reduziria o número de problemas do conjunto $\mathscr{P}$, influenciando na qualidade da análise de desempenho computacional, por intermédio da técnica do *performance profile*.

Ademais, menciona-se que todos os algoritmos abordados nesse trabalho foram implementados no Matlab R2012b-64bits, sendo os experimentos numéricos executados em um Notebook Dell Precision, com processador Intel(R) Core(TM) i7 de 2,60 GHz, 32,0 GB de memória e sistema operacional Windows 10 Pro.

Logo, dispondo do conjunto $\mathscr{P}$, definido na seção 4.1, os métodos de otimização do conjunto $\mathscr{A}$ e a técnica de comparação, o *performance profile* (seção 4.2), definiu-se como critério de desempenho computacional o número de iterações do método de otimização e o tempo de processamento computacional, visto que estas informações são complementares. Não é desejado um método com baixo número de iterações e com alto tempo de processamento computacional por iteração, pois, o agregado pode ser um tempo de processamento muito elevado se comparado com os demais métodos. Portanto, tem-se todas as informações necessárias para iniciar as análises qualitativas e quantitativas.

Tabela 5 – *Performance Profile* do conjunto de problemas $\mathscr{P}$, listados na tabela 2. "iter" é o acrónimo para número de iterações.

Performance Profile			
Método	Eficiência (Iter) %	Eficiência (tempo) %	Robustez %
CG_HS	12,82	6,41	55,13
CG_PR	5,13	1,28	58,97
CG_DL	17,95	14,10	75,64
CG_GY	32,05	28,21	82,05
CG_HS*	52,56	44,87	88,46

A tabela 5 e as figuras 7 e 8 apresentam os resultados para o conjunto $\mathscr{P}$ utilizando a técnica de análise do *performance profile*, seção 4.2. Os dados para esta análise podem ser consultados nas tabelas 7 e 8 do Apêndice B, em que é apresentado o número do problema de otimização, seguindo a mesma referência da tabela 3, juntamente à informação da dimensão do problema de otimização. Em seguida, para cada método de gradiente conjugado, é apresentado o número de iterações, na tabela 7, e tempo de processamento computacional, na tabela 8, para o caso de convergência do método e, caso contrário, a sigla NC, que significa a não convergência ao ponto minimizador do problema.

Destacam-se a eficiência, em relação ao tempo de $44,87\%$, do método do gradiente conjugado HS* e a sua robustez, de $88,46\%$. É notória, com a análise das figuras 7a e 8a, a sua superioridade em solucionar problemas de otimização irrestrita em alta dimensão, pois não existe um intervalo de τ no qual o gráfico de $\rho_{\text{ CG-HS}^*}(\tau)$, definido na expressão (4.4), está abaixo do gráfico de $\rho_{A_i}(\tau)$ dos demais métodos, com A_i representando os métodos: GC-HS, GC-PR, GC-DL e GC-GY. Essa observação é valida para ambos os critérios de desempenho computacional: tempo de processamento e número de iterações.

As figuras 7b e 8b possuem o objetivo de realçar o desempenho dos métodos para $\tau \in [1, 2]$ e, desta forma, destacar os valores de eficiência encontrados para ambos os critérios de desempenho. Além disso, ao analisar as figuras 7a e 8a é atestado que as alterações introduzidas por (YUAN, 2009) ao método clássico GC-PR, descrito na seção 2.2.2, originando o método GC-GY, seção 3.1, melhoram o desempenho computacional deste para a classe de problemas do conjunto $\mathscr{P}$.

Outra observação é a existência de problemas do conjunto $\mathscr{P}$ em que nenhum dos métodos de otimização utilizados para comparação foi capaz de solucioná-los, podendo tal fato ser aferido nas tabelas 7 e 8 do Apêndice B. Mais ainda, pela análise dos valores de robustez, presentes na tabela 5 e diferentes entre si, é possível concluir que há problemas de otimização, no conjunto $\mathscr{P}$, que foram solucionados apenas por alguns dos métodos de otimização.

Enfatiza-se que, entre as possíveis causas de não convergência de métodos de otimização, a sequência gerada:

- entra em estado de "estagnação", com comprimentos de passo, $\|s_k\| = \|x_{k+1} - x_k\|$, suficientemente próximos a zero sem alcançar o ponto de mínimo;
- é prematuramente finalizada, pois o número de iterações excedeu o máximo permitido, indicando que havia progresso, com comprimento de passos, $\|s_k\| = \|x_{k+1} - x_k\|$, não suficientemente próximos a zero, não tendo convergido devido ao critério de parada;
- tende a ser ilimitada, sendo truncada pelo critério de parada do número máximo de iterações;
- converge para mínimo local diferente do esperado.

No caso dos experimentos numéricos reportados neste trabalho, não foi feita a distinção entre possíveis causas de um algoritmo não convergir, ficando registrada apenas a ocorrência de não convergência.

Por fim, durante a execução dos testes computacionais, em especial ao executar os testes com a função trigonométrica (4.2), alterou-se a busca em linha a fim de atender as condições de Armijo (2.5), sendo a implementação computacional realizada conforme algoritmo 1. O resultado foi que os métodos GC-HS e GC-DL, respectivamente seções 2.2.1 e 2.2.3, ficam mal definidos numericamente, não ocorrendo esse problema com o método gradiente conjugado HS modificado, seção 3.2. Logo, surge a seguinte pergunta: Por quê o método proposto não fica mal definido numericamente, uma vez que, possuem o coeficiente de conjugação β_k com o mesmo denominador: $p_k^\intercal y_k$?

Primeiramente, entende-se por: "o método do gradiente conjugado está numericamente mal definido"; que em alguma iteração $k > 1$ o valor β_k se torna NaN (*not a number*) ou infinito, indicando uma possível divisão por número muito próximo a zero, mas, que na precisão do computador, torna-se uma divisão por zero.

Retomando a resposta para a pergunta, tem-se que para computar β_k nesses métodos, GC-HS, GC-DL e GC-HS*, é necessária, como já mencionado, uma divisão por $p_k^\intercal y_k$, com $y_k = g_{k+1} - g_k$. Como retira-se a condição de curvatura (2.7), há valores para k, em que $\alpha_k \ll 1$, implicando que, para problemas suaves, $\|y_k\| \ll 1$ e, na precisão numérica do computador, tem-se $g_{k+1} = g_k$. Portanto, uma divisão de zero por zero, pois y_k também está presente no numerador, resultando em um valor indefinido (*NaN*). No método HS modificado há uma proteção contra esse tipo de problema, pois, β_k^{HS*} só será calculado se $s_k^\intercal y_k > \delta$, caso contrário $\beta_k^{HS*} = 0$. Dessa forma, garante-se que o método do gradiente conjugado HS modificado esteja bem definido numericamente quando se utilizar a condição de Armijo (2.5). Nota-se tal característica demonstrada na propriedade **(P1)**.

4.4 Comparação com o Método de Região de Confiança do Software *Matlab*

Uma pergunta inerente ao processo de proposta de um novo algoritmo, é como este comporta-se em relação a outros algoritmos de otimização presentes em softwares comerciais, e.g. *Matlab* com a biblioteca *Optimization Toolbox*.

Para responder essa pergunta, utiliza-se a análise de *performance profile*, seção 4.2, para os problemas de otimização definidos de 1 à 14 na tabela 2, com dimensão igual à 400, empregando, para fins de comparação, os métodos de gradiente conjugado: GC-HS; GC-PR; GC-DL; GC-GY; CG-HS*; e o método de região de confiança (COLEMAN; LI, 1996). Este método está disponível por meio do comando *fminunc*, presente na biblioteca *Optmization* do software *Matlab*. Como critérios de desempenho computacional, mantém-se o número de iterações e o tempo de processamento computacional.

Esse número de dimensão para os problemas de otimização é motivado, principalmente, pelo baixo desempenho computacional, em relação ao tempo de processamento, do método de região de confiança, que utiliza a informação da curvatura da função objetivo, matriz Hessiana, para encontrar a nova solução-tentativa. Entretanto, ao empregar esse método para otimização em alta dimensão, essa matriz torna-se extremamente grande (n^2 elementos), necessitando de métodos computacionais específicos para lidar com a esparsidade ou a densidade dessa matriz a fim de tornar o método de região de confiança eficaz, em relação ao tempo de processamento, para essa classe de problemas de otimização.

Logo, tem-se todas as informações necessárias para o início das análises qualitativas e quantitativas. A tabela 6 apresenta os resultados: eficiência e robustez; comparando o método comercial, região de confiança, aos métodos de gradiente conjugado, entre eles o método gradiente conjugado HS modificado (GC-HS*), proposto neste trabalho. De acordo com essa tabela 6, destacam-se a robustez de $57,14\%$ e a baixa eficiência, em relação ao tempo de processamento computacional, de $0,00\%$ do método de região de confiança, o que o torna pouco

Tabela 6 – *Performance Profile* comparando os métodos de Otimização do MatLab com os métodos de Gradiente Conjugado para os problemas de 1 à 14 da tabela 2, com dimensão igual a 400. "iter" é o acrónimo para número de iterações.

Performance Profile			
Método	Eficiência (tempo) %	Eficiência (Iter) %	Robustez %
CG_HS	0,00	7,14	50,00
CG_PR	7,14	7,14	57,14
CG_DL	21,43	7,14	85,71
CG_GY	28,57	7,14	78,57
CG_HS*	42,86	50,00	92,86
Trust-Region	0,00	50,00	57,14

recomendável para problemas de otimização em alta dimensão, como já esperado para este critério de desempenho. Pois, há uma comparação, quanto ao critério de tempo de processamento computacional, entre um método de segunda ordem (região de confiança), que como já dito, utiliza informações de segunda ordem da função objetivo f a cada iterando, e métodos de primeira ordem (gradiente conjugado). O tempo de processamento computacional por iteração tende a ser menor nestes métodos em comparação com aqueles. Por isso, salienta-se a importância de analisar o tempo e o número de iterações simultaneamente como critério de desempenho computacional.

A figura 9 mostra os gráficos de $\rho_{A_i}(\tau)$, expressão (4.4), para os métodos testados, tendo o número de iterações como critério de desempenho computacional. Para este critério, o valor $\rho_{A_i}(1)$ define a eficiência do método de otimização. Contudo, ao se tomar o extremo direito dos gráficos de $\rho_{A_i}(\tau)$, tem-se a robustez do método. Destaca-se que a robustez é a mesma, independente do critério de desempenho que está sendo adotado. Tal afirmação pode ser visualizada no extremo direito do gráfico da figura 10a, que possui como critério de desempenho o tempo de processamento computacional.

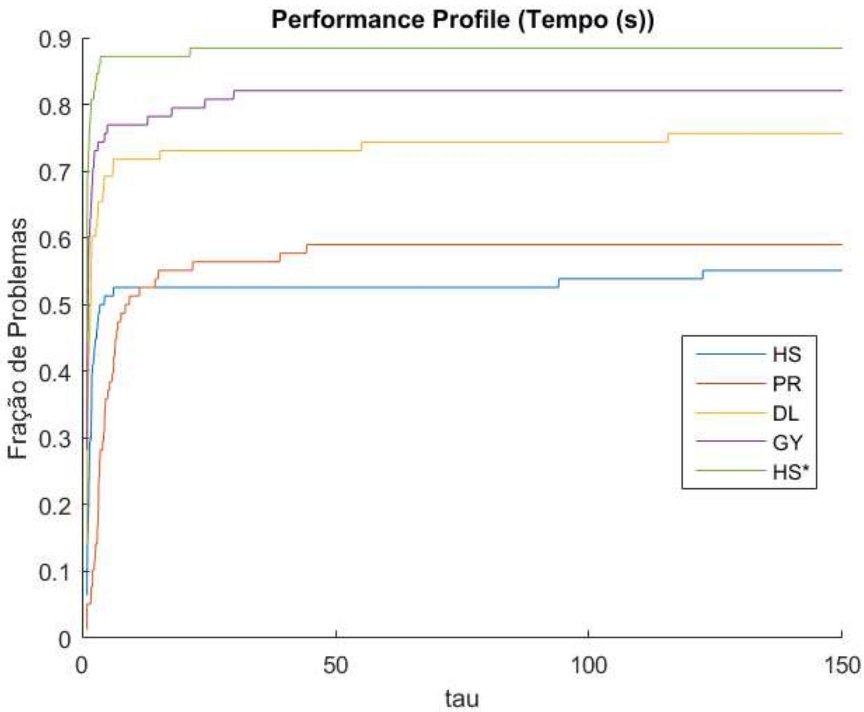
Em seguida, ao inspecionar a figura 10, tem-se os resultados da análise do *performance profile* para o critério de desempenho: tempo de processamento computacional. Na figura 10b, são apresentados os gráficos de $\rho_{A_i}(\tau)$ da figura 10a, reduzindo o intervalo de análise para $\tau \in [1, 1.7]$. O objetivo da redução desse intervalo é possibilitar ao leitor a conferência dos valores de eficiência, em relação a esse critério, presentes na tabela 6.

As tabelas 9 e 10 com os dados utilizados para a análise do *performance profile*, e também na construção da tabela 6 e figuras 9 e 10, podem ser encontradas no Apêndice B. Nessas tabelas, são apresentados o número do problema de otimização, seguindo a mesma referência da tabela 3 e, para cada método de otimização, o número de iterações, na tabela 9, e tempo de processamento computacional, na tabela 10, para o caso de convergência do método. Caso contrário, é escrito a sigla NC, que significa a não convergência ao ponto minimizador do problema.

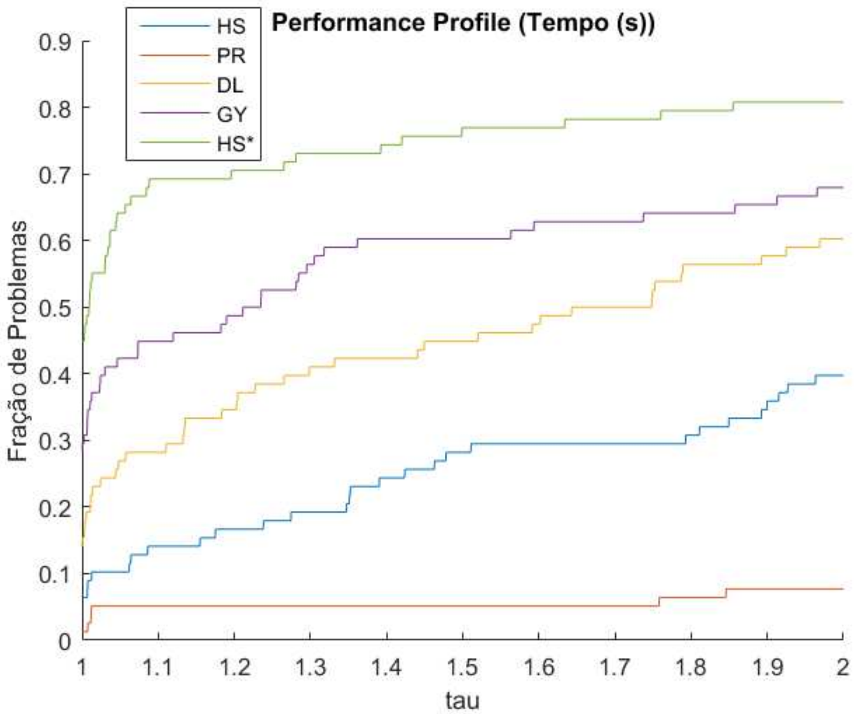
Por fim, destaca-se o bom desempenho computacional do método proposto para os testes preliminares. O método do gradiente conjugado HS modificado (GC-HS*) apresentou maior robustez, capacidade de encontrar a solução do problema de otimização em alta-dimensão, alinhada a eficiência computacional. Ou seja, esse novo método, dentre os analisados, é o que apresenta maior capacidade de solucionar um problema de otimização no menor tempo de processamento computacional.

Figura 7 – *Performance Profile* do Tempo de Processamento do Conjunto $\mathscr{P}$, apresentando os gráficos de $\rho_{A_i}(\tau)$, definida na expressão (4.4), em que A_i representa os métodos de otimização: GC-HS; GC-PR; GC-DL; GC-GY; GC-HS*. O eixo das ordenas representa a fração de problemas solucionados pelo método A_i até um determinado valor de $\tau \geq 1$.

(a) Ênfase na robustez, que é definida como o extremo direito dos gráficos de $\rho_{A_i}(\tau)$.



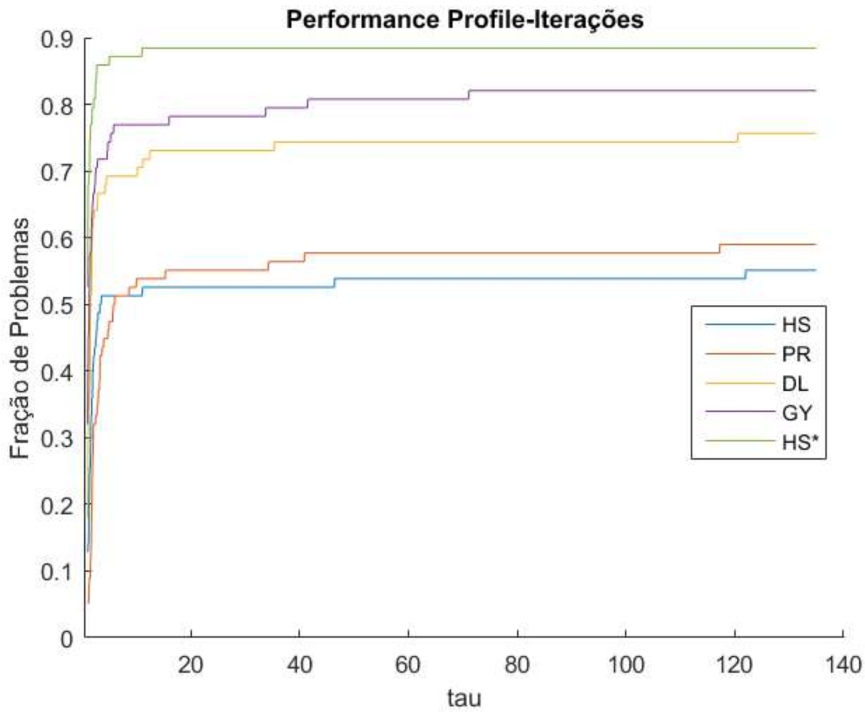
(b) Ênfase na eficiência, que é definida como o valor de $\rho_{A_i}(1)$. Este gráfico é o mesmo da figura acima, destacando $\tau \in [1, 2]$.



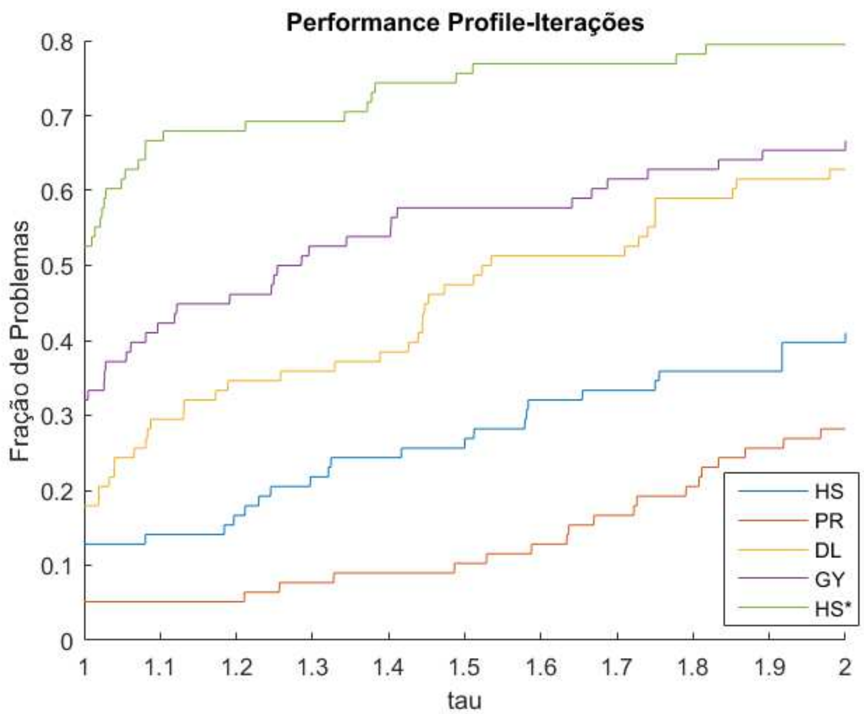
Fonte: figura do autor.

Figura 8 – *Performance Profile* do Número de Iterações do Conjunto $\mathscr{P}$, apresentando os gráficos de $\rho_{A_i}(\tau)$, definida na expressão (4.4), em que A_i representa os métodos de otimização: GC-HS; GC-PR; GC-DL; GC-GY; GC-HS*. O eixo das ordenas representa a fração de problemas solucionados pelo método A_i até um determinado valor de $\tau \geq 1$.

(a) Ênfase na robustez, que é definida como o extremo direito dos gráficos de $\rho_{A_i}(\tau)$.

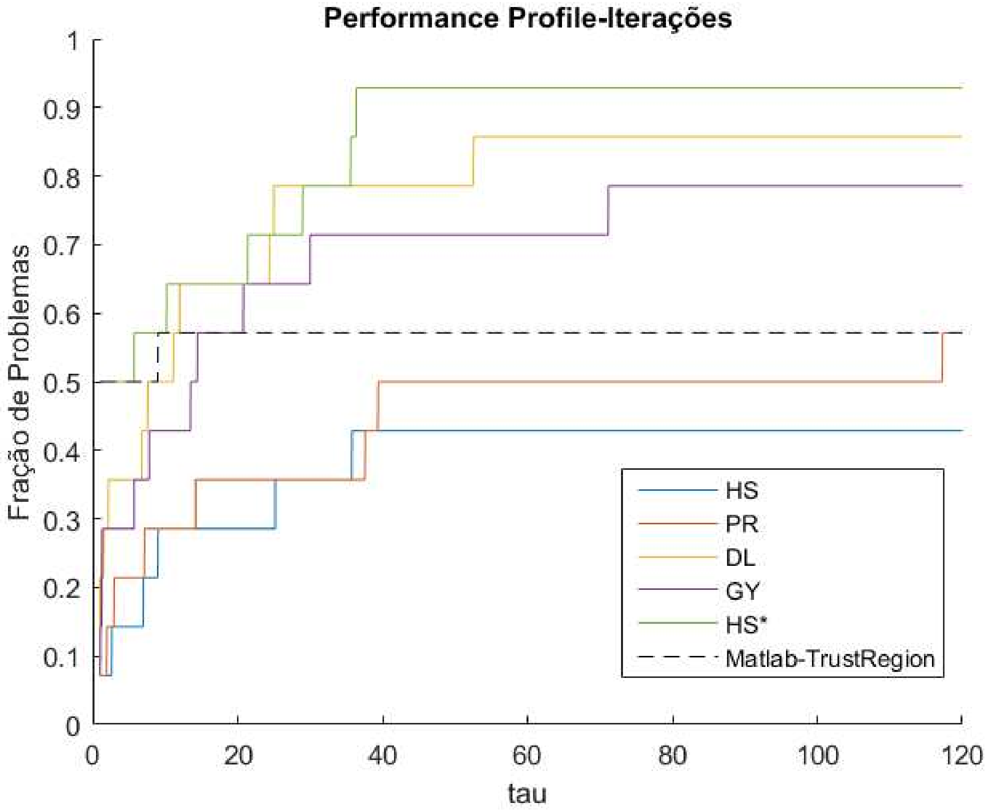


(b) Ênfase na eficiência, que é definida como o valor de $\rho_{A_i}(1)$. Este gráfico é o mesmo da figura acima, destacando $\tau \in [1, 2]$.



Fonte: figura do autor.

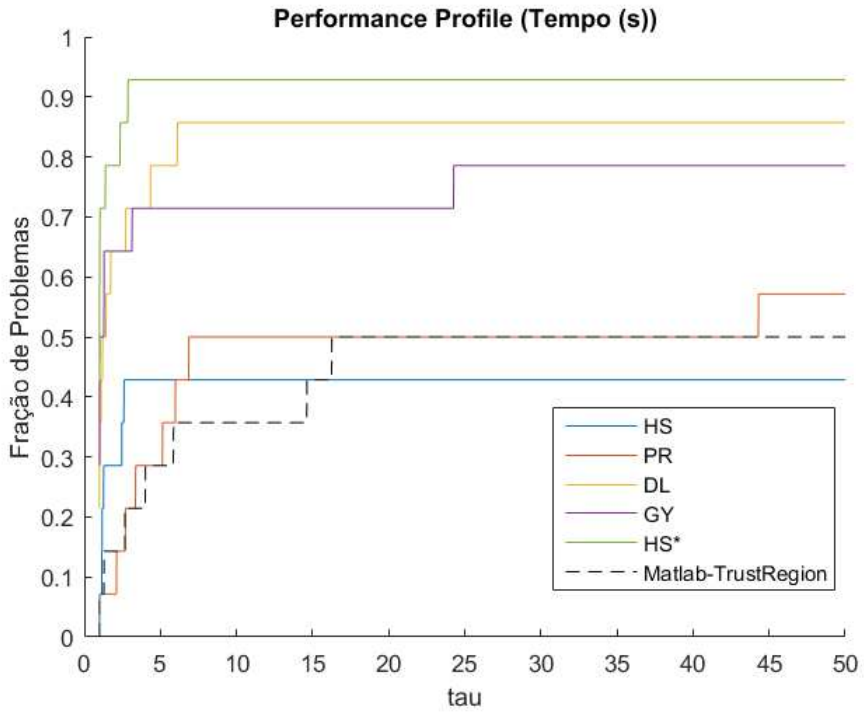
Figura 9 – *Performance Profile* do Número de Iterações utilizando os métodos de otimização Matlab e GC. O valor $\rho_{A_i}(1)$, com $\rho_{A_i}(\tau)$ sendo definida na expressão (4.4), representa a eficiência do método de otimização. Já, ao se tomar o extremo direito dos gráficos de $\rho_{A_i}(\tau)$, tem-se a robustez do método.A_i representa os métodos de otimização: GC-HS; GC-PR; GC-DL; GC-GY; GC-HS*.



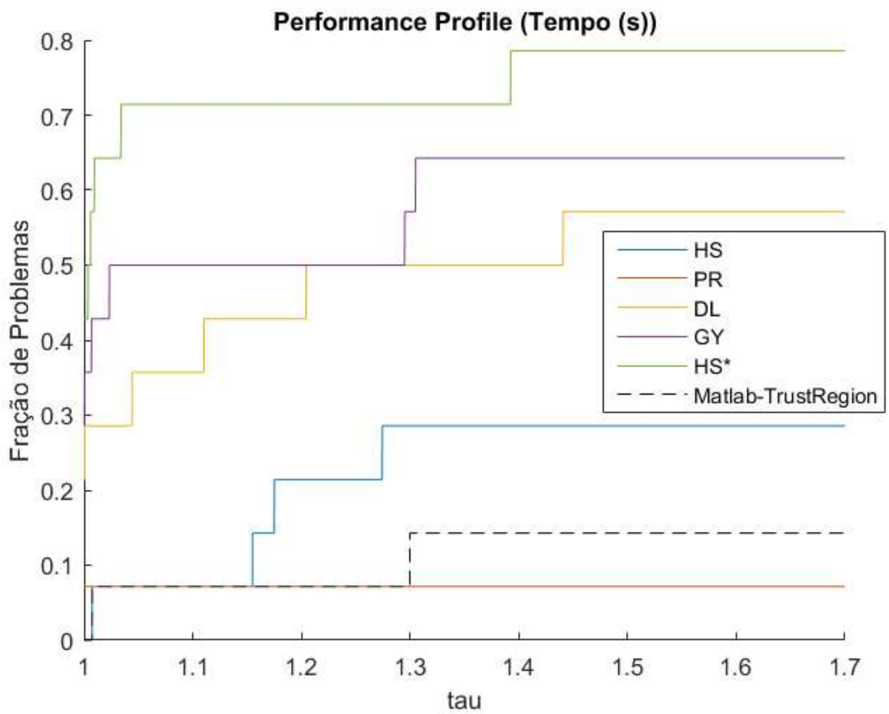
Fonte: figura do autor.

Figura 10 – *Performance Profile* do Tempo de Processamento utilizando os métodos de otimização Matlab e GC, apresentando os gráficos de $\rho_{A_i}(\tau)$, definida na expressão (4.4), em que A_i representa os métodos de otimização: GC-HS; GC-PR; GC-DL; GC-GY; GC-HS*. O eixo das ordenas representa a fração de problemas solucionados pelo método A_i até um determinado valor de $\tau \geq 1$. Os problemas deste conjunto são os definidos de 1 à 14 na tabela 2, com dimensão igual a 400.

(a) Ênfase na robustez, que é definida como o extremo direito dos gráficos de $\rho_{A_i}(\tau)$.



(b) Ênfase na eficiência, que é definida como o valor de $\rho_{A_i}(1)$. Este gráfico é o mesmo da figura acima, destacando $\tau \in [1, 1.7]$.



Fonte: figura do autor.

5 Conclusões e Considerações Finais

O presente trabalho é iniciado, no capítulo 1, relatando os possíveis problemas ocasionados pelo aumento da dimensão dos problemas de otimização, sendo possíveis destacar: memória física do computador insuficiente; tempo de processamento computacional elevado; métodos numéricos ineficientes quando se eleva a dimensão do problema. Em seguida, restringiu-se o universo de problemas de otimização à classe dos problemas de otimização suave irrestrita em alta dimensão, selecionando, entre os métodos de primeira ordem, o gradiente conjugado para propor uma nova abordagem para esta família de métodos, definida nos algoritmos 6 e 7 e denominada Gradiente Conjugado HS Modificado (CG HS*-seção 3.2). Ressalta-se que este método proposto tem como pilares os métodos de gradiente conjugado HS (seção 2.2.1), DL (seção 2.2.3) e GY (seção 3.1), possuindo como objetivo apresentar melhor desempenho computacional que os métodos base (CG-HS, CG-DL e CG-GY).

Sabida a necessidade de utilização da busca em linha nessa classe de métodos de gradiente conjugado para otimização não-linear, utiliza-se aquela satisfazendo a:

- condição forte de Wolfe-Powell 2.1.2.3 para a prova de convergência global;
- condição fraca de Wolfe-Powell 2.1.2.2 para os experimentos numéricos.

Entretanto, observou-se através da propriedade **(P1)** que o coeficiente de conjugação β_k do método HS modificado ficaria bem definido para qualquer das condições de parada da busca em linha (Armijo, Wolfe fraca, Wolfe forte). Tal observação sugeriu que o método HS modificado poderia ser executado satisfatoriamente utilizando apenas a condição mais simples, que é a condição de Armijo 2.1.2.1, implementada pelo algoritmo 1. Porém, esta conjugação não pode ser realizada com os métodos HS e DL. Esses elementos conduzem à seguinte questão, que deverá ser abordada em trabalhos futuros:

- há como provar a convergência global, sob as hipóteses tratadas na seção 3.2, utilizando a busca em linha satisfazendo as condições fraca de Wolfe-Powell ou apenas a condição de Armijo?

Na etapa de experimentos numéricos, capítulo 4, utilizando a técnica do *performance profile*, seção 4.2, para comparar os métodos, teve-se o indicativo que o método proposto é promissor, apresentando, em geral, menor tempo e número de iterações para a convergência, além de maior robustez, apontando superioridade na capacidade de solucionar problemas de otimização suave irrestrita em alta dimensão em comparação aos demais métodos (CG-HS, CG-PR, CG-DL e CG-GY), como pode ser aferido na tabela 5.

Contudo, é notória a necessidade de se aumentar o número de problemas teste e de métodos de otimização desenvolvidos para a otimização em alta dimensão para afirmações e conclusões mais sólidas e robustas. A princípio, foram encontrados bons indicativos de que o método proposto possa se tornar uma boa escolha para essa classe de otimização.

Por fim, como proposta para trabalhos futuros, elenca-se:

- responder a pergunta teórica realizada acima, destacando, como ponto de partida, a leitura do artigo (DANIEL, 1967);
- aumentar o conjunto de métodos numéricos e problemas teste (problemas *benchmark*) para otimização em alta dimensão com o intuito de analisar o *performance profile* e, consequentemente, conferir maior confiabilidade na análise de eficiência e robustez do método proposto. Como sugestão, poderia-se formar o conjunto de problemas como sendo todos os problemas presentes na biblioteca CUTEst (GOULD; ORBAN; TOINT, 2015), utilizando *Julia* como linguagem de programação. Pois, há a interface do banco de problemas de otimização da CUTest com o *Julia*, fato que não ocorre com o software *Matlab*. Destaca-se também a necessidade de aumentarmos o conjunto de métodos de otimização, podendo ser acrescidos com os métodos presentes na biblioteca *BIGDOT*.
- realizar a análise de sensibilidade da eficiência e robustez quanto ao aumento da dimensão dos problemas de otimização nos diferentes métodos. Sugere-se construir um conjunto de problemas de otimização e aumentar gradativamente a dimensão dos problemas desse conjunto, registrando os valores médios de: tempo de processamento computacional e número de iterações até a solução dos problemas. Posteriormente, deve-se registrar esses valores em um gráfico, eixo das abcissas é a dimensão do problemas e o eixo das ordenadas é a média da variável de interesse, e observar o seu comportamento, se é monotônico, se há cruzamento entre métodos, etc. Destaca-se que não é aconselhado utilizar a técnica do *performance profile*, pois este faz uma ordenação dos métodos por problema. Outra sugestão, seria a análise por *box-plot* a fim de capturar a variância das informações;
- desenvolver um método para aperfeiçoar a escolha dos parâmetros t, μ e δ, presentes nos algoritmos 6 e 7 propostos, e.g. utilizar a análise do *performance profile* nesses parâmetros;
- ou, até mesmo, formular um método do gradiente conjugado HS modificado adaptativo, no qual os parâmetros t, μ e δ são atualizados a cada iteração ou ciclos de iteração. O intuito dessa proposta seria alterar dinamicamente os valores desses parâmetros com base em informações locais da função objetivo, por exemplo: condição de curvatura, número de condicionamento da matriz que aproxima localmente a Hessiana da função objetivo, etc. O objetivo final seria um novo método com desempenho computacional superior a todos os métodos que mantém os parâmetros estáticos.

Referências

AL-BAALI, M.; FLETCHER, R. An efficient line search for nonlinear least squares. *Journal of Optimization Theory and Applications*, Springer, v. 48, n. 3, p. 359–377, 1986. Citado na página 26.

ALENCAR, F. O que é a lei de Moore. *Entenda a teoria que "prevê"o futuro da Informática. TechTudo. Disponível em: http://www. techtudo. com. br/noticias/noticia/2015/06/o-que-e-lei-de-moore-entenda-teoria-que-preve-futuro-da-informatica. html. Acesso em: 1º/2022*, 2019. Citado na página 15.

ANDREI, N. Accelerated conjugate gradient algorithm with finite difference Hessian/vector product approximation for unconstrained optimization. *Journal of Computational and Applied Mathematics*, Elsevier, v. 230, n. 2, p. 570–582, 2009. Citado na página 20.

BAUSCHKE, H. H.; COMBETTES, P. L. et al. *Convex analysis and monotone operator theory in Hilbert spaces*. [S.l.]: Springer, 2011. v. 408. Citado na página 31.

BROYDEN, C. G. Quasi-Newton methods and their application to function minimisation. *Mathematics of Computation*, v. 21, n. 99, p. 368–381, 1967. Citado 3 vezes nas páginas 15, 20 e 31.

BURKE, J. V. *Nonlinear Optimization*. [S.l.]: https://sites.math.washington.edu/ burke, 2020. Citado na página 24.

COLEMAN, T. F.; LI, Y. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, SIAM, v. 6, n. 2, p. 418–445, 1996. Citado 3 vezes nas páginas 15, 16 e 60.

DAI, Y. H.; LIAO, L. Z. New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and Optimization*, Springer, v. 43, n. 1, p. 87–101, 2001. Citado 3 vezes nas páginas 16, 30 e 31.

DAI, Y. H.; YUAN, Y. A nonlinear conjugate gradient with a strong global convergence property. *SIAM Journal of Optimization*, v. 10, 1999. Citado na página 34.

DAI, Y. H.; YUAN, Y. Nonlinear conjugate gradient methods. *Shanghai Science and Technology Publisher, Shanghai*, 2000. Citado na página 29.

DANIEL, J. W. The conjugate gradient method for linear and nonlinear operator equations. *SIAM Journal on Numerical Analysis*, SIAM, v. 4, n. 1, p. 10–26, 1967. Citado na página 68.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming*, Springer, v. 91, n. 2, p. 201–213, 2002. Citado na página 55.

FLETCHER, R. *Practical methods of optimization. Volume 1: Unconstrained Optimization*. [S.l.]: John Wiley & Sons, 1987. Citado na página 29.

FLETCHER, R.; REEVES, C. M. Function minimization by conjugate gradients. *The Computer Journal*, Oxford University Press, v. 7, n. 2, p. 149–154, 1964. Citado 2 vezes nas páginas 29 e 34.

GILBERT, J. C.; NOCEDAL, J. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, SIAM, v. 2, n. 1, p. 21–42, 1992. Citado 2 vezes nas páginas 30 e 31.

GOLUB, G. H.; O'LEARY, D. P. Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. *SIAM Review*, SIAM, v. 31, n. 1, p. 50–102, 1989. Citado 3 vezes nas páginas 16, 19 e 29.

GOULD, N. I.; ORBAN, D.; TOINT, P. L. Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, Springer, v. 60, n. 3, p. 545–557, 2015. Citado 2 vezes nas páginas 52 e 68.

HESTENES, M. R.; STIEFEL, E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, US Government Printing Office, v. 49, n. 6, p. 409, 1952. Citado 2 vezes nas páginas 16 e 27.

ISMAILOV, A.; SOLODOV, M. *Otimização: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. [S.l.]: Rio de Janeiro: IMPA, 2005. v. 1. Citado 4 vezes nas páginas 28, 39, 73 e 75.

KRESSNER, D. *Advanced Numerical Analysis*. [S.l.]: Springer, 2015. Citado na página 22.

LIU, Y.; STOREY, C. Efficient generalized conjugate gradient algorithms, Part 1: Theory. *Journal of Optimization Theory and Applications*, Springer, v. 69, n. 1, p. 129–137, 1991. Citado 2 vezes nas páginas 29 e 34.

MORÉ, J. J.; GARBOW, B. S.; HILLSTROM, K. E. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software (TOMS)*, ACM New York, NY, USA, v. 7, n. 1, p. 17–41, 1981. Citado 4 vezes nas páginas 52, 54, 55 e 76.

NOCEDAL, J.; WRIGHT, S. J. Conjugate gradient methods. *Numerical Optimization*, Springer, p. 101–134, 2006. Citado 4 vezes nas páginas 15, 16, 17 e 30.

NOCEDAL, J.; WRIGHT, S. J. Line search methods. *Numerical Optimization*, Springer, p. 30–65, 2006. Citado 7 vezes nas páginas 16, 17, 20, 21, 22, 24 e 25.

POLAK, E.; RIBIÈRE, G. Note sur la convergence de méthodes de directions conjuguées. *Revue Française D'Informatique et de Recherche Opérationnelle*, v. 16, p. 35–43, 1969. Citado na página 29.

POLYAK, B. T. The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, Elsevier, v. 9, n. 4, p. 94–112, 1969. Citado na página 16.

POWELL, M. J. Nonconvex minimization calculations and the conjugate gradient method. In: *Numerical analysis*. [S.l.]: Springer, 1984. p. 122–141. Citado na página 29.

RAO, S. S. *Engineering optimization: theory and practice*. [S.l.]: John Wiley & Sons, 2019. Citado 3 vezes nas páginas 15, 21 e 22.

SAFF, E. B.; SNIDER, A. D. *Fundamentals of Matrix Analysis with Applications*. [S.l.]: John Wiley & Sons, 2015. Citado na página 27.

SHEWCHUK, J. R. *An introduction to the conjugate gradient method without the agonizing pain*. Pittsburgh: Carnegie-Mellon University. Department of Computer Science, 1994. Citado 3 vezes nas páginas 16, 19 e 28.

VIANA, M. Avanço da matemática explica o poder da computação. *Folha de São Paulo - https://www1.folha.uol.com.br/colunas/marceloviana/2022/04/avanco-da-matematica-explica-o-poder-da-computacao.shtml*, 2022. Citado na página 15.

YUAN, G. Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. *Optimization Letters*, Springer, v. 3, n. 1, p. 11–21, 2009. Citado 12 vezes nas páginas 13, 16, 17, 20, 29, 30, 33, 34, 37, 41, 48 e 59.

YUAN, G.; SHENG, Z.; LIU, W. The modified HZ conjugate gradient algorithm for large-scale nonsmooth optimization. *PLoS One*, Public Library of Science, San Francisco, CA USA, v. 11, n. 10, p. e0164289, 2016. Citado na página 15.

ZHANG, L. New versions of the Hestenes-Stiefel nonlinear conjugate gradient method based on the secant condition for optimization. *Computational & Applied Mathematics*, Scielo Brasil, v. 28, n. 1, p. 111–133, 2009. Citado na página 29.

ZOUTENDIJK, G. Nonlinear programming, computational methods. *Integer and nonlinear programming*, North-Holland, p. 37–86, 1970. Citado 4 vezes nas páginas 17, 29, 34 e 40.

Apêndices

APÊNDICE A – Demonstrações

É apresentado a demonstração para **(C2)**.

(C2) Sejam $f : \mathbb{R}^n \to \mathbb{R}$ uma função continuamente diferenciável, $x_0 \in \mathbb{R}^n$ tal que $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ é limitado. Se f' é localmente Lipschitz, então existe um conjunto Ω_0 aberto e convexo, tal que $\Omega \subset \Omega_0$ e em Ω_0:

i) f é inferiormente limitada;

ii) f é diferenciável;

iii) f' é Lipschtz contínua

Demonstração:
Como Ω é limitado, então o envelope convexo de Ω, denotada por $CO\ \Omega$ é um conjunto:

I$_1$) convexo, pela própria definição de envelope convexo;

I$_2$) limitado, pois o envelope convexo de um conjunto limitado é limitado (Proposição 3.2.14 na página 79 de (ISMAILOV; SOLODOV, 2005)).

De I_1 e I_2, temos $\overline{CO}\ \Omega$ é um conjunto compacto.
Visto que f' é localmente Lipschtiz, para cada $x \in \overline{CO}\ \Omega$ existem ε_x e $L_x > 0$ tal que

$$\|f'(y) - f'(z)\| \leq L_x \|y - z\| \quad \forall\, y, z \in B(x, \varepsilon_x).$$

Logo,

$$CO\ \Omega \subset \overline{CO}\ \Omega \subset \bigcup_{x \in \overline{CO}\ \Omega} B(x, \varepsilon_x)$$

sendo $\overline{CO}\ \Omega$ um conjunto compacto, existe uma subcobertura finita de $\overline{CO}\ \Omega$ tal que

$$\Omega \subset CO\ \Omega \subset \overline{CO}\ \Omega \subset \bigcup_{k=1}^{r} B(x_k, \varepsilon_{x_k}).$$

Definindo $\mathcal{C} = CO\ \bigcup_{k=1}^{r} B(x_k, \varepsilon_{x_k})$, temos

I$_3$) $\mathcal{C}$ é um conjunto aberto, pois o envelope convexo de um conjunto aberto é aberto **Proposição A.1)**;

I$_4$) convexo, por definição;

I_5) $\overline{\mathcal{C}}$ é um conjunto compacto e f é continuamente diferenciável, portanto, contínua. Pelo Teorema de Weierstrass, existe $a \in \mathbb{R}$ tal que

$$a \leq f(x) \quad \forall\, x \in \overline{\mathcal{C}} \quad (*)$$

consequentemente, (*) é válida para todo $x \in \mathcal{C}$;

I_6) f' é Lipschtz contínua em $\mathcal{C}$.

Como $\overline{\mathcal{C}}$ é um conjunto compacto, existe uma subcobertura finita de $\overline{\mathcal{C}}$, $\{U_i\}_{i=1}^{p}$ com $U_i \subset \mathbb{R}^n$, $f|_{U_i}$ lipschitz contínua, L_i a constante de Lipschitz e

$$\mathcal{C} \subset \overline{\mathcal{C}} \subset \bigcup_{i=1}^{p} U_i.$$

Dados $x, y \in \mathcal{C}$, então $[x, y] \subset \mathcal{C}$, pois $\mathcal{C}$ é convexo. Tome uma partição de $[x, y]$, formada por caminhos retilíneos $[a_i, b_i]$, disjuntos entre si, tais que:

$$[a_i, b_i] \subset U_j \cap [x, y] \quad \text{com } j \in \{1, \cdots, p\},$$

quando $U_j \cap [x, y] \neq \emptyset$. Logo, existe $m \in \mathbb{N}$ tal que $[x, y] = \bigcup_{i=1}^{m} [a_i, b_i]$.

Defina $L = \max\limits_{L \in \{1, \cdots, p\}} \{L_i\}$ e

$$\begin{aligned}
\|f'(x) - f'(y)\| \leq & \left\| \sum_{i-1}^{m} f'(a_i) - f'(b_i) \right\| \\
\leq & \sum_{i=1}^{m} \|f'(a_i) - f'(b_i)\| \\
\leq & \sum_{i=1}^{m} L_i \|a_i - b_i\| \\
\leq & L \sum_{i=1}^{m} \|a_i - b_i\| \\
\leq & L\|x - y\|
\end{aligned}$$

portanto, basta tomar $\Omega_o = \mathcal{C}$.

□

O exercício 3.2.17 de (ISMAILOV; SOLODOV, 2005) é tratado como proposição e sua demonstração apresentada. Utilizou-se essa proposição para provar **(C2)**.

Proposição A.1) Seja $D \subset \mathbb{R}^n$ um conjunto aberto, então $CO\ D$ é aberto.

Prova:

Primeiramente, enunciamos a seguinte proposição:

Proposição 3.2.13 (pagina 78, (ISMAILOV; SOLODOV, 2005)) : Seja $D \subset \mathbb{R}^n$ um conjunto qualquer. o envelope convexo de D é o conjunto de todas as combinações convexas de pontos de D.

Dado $x \in CO\ D$, pela Proposição 3.2.1 existe $p \in \mathbb{N}$ tal que $x = \sum_{i=1}^{p} \alpha_i x_i$, $\sum_{x=1}^{p} \alpha_i = 1$, $x_i \in D$, $\alpha_i \in \mathbb{R}_+$ e $i = 1, \cdots, p$.

Como $x_i \in D$ e sendo D um conjunto aberto, existe $\delta_i > 0$ tal que $B(x_i, \delta_i) \subset D \subset CO\ D$ para $i = 1, \cdots, p$.

Defina $\delta = \min\limits_{i \in \{1, \cdots, p\}} \{\delta_i\}$, daí $B(x_i, \delta) \subset D$ para $i = 1, \cdots, p$. Afirmamos que $B(x, \delta) \subset CO\ D$.

Com efeito, dado $y \in B(x, \delta)$, definindo $w = y - x$ e $\|w\| < \delta$ implica

$$\begin{aligned} y &= \sum_{i=1}^{p} \alpha_i x_i + w \\ &= \sum_{i=1}^{p} \alpha_i x_i + \sum_{i=1}^{p} \alpha_i w \\ &= \sum_{i=1}^{p} \alpha_i (x_i + w). \end{aligned}$$

Visto que $z_i = x_i + w \in B(x_i, \delta)$ para $i = 1, \cdots, p$ então $y = \sum_{i=1}^{p} \alpha_i z_i$, com $\sum_{i=1}^{p} \alpha_i = 1$, $z_i \in D$, $\alpha_i \in \mathbb{R}$ e $i = 1, \cdots, p$.

Utilizando a Proposição 3.2.13, temos $y \in CO\ D$.

□

APÊNDICE B – Dados Utilizados na Análise do *Performance Profile*

Nesta parte do apêndice é apresentado as tabelas para a construção dos gráficos do perfil de performance *performance profile* (MORÉ; GARBOW; HILLSTROM, 1981). O significado de NC, presente nas tabelas a seguir, é não convergência.

Tabela 7 – Dados *Performance Profile* conjunto $\mathscr{P}$-Iterações. A dimensão n do problema de otimização está entre parêntese após o número do problema. NC é a sigla para **Não Convergência**.

Performance Profile (iteração) - Problemas Teste					
Problema \ Algoritmo	CG_HS	CG_PR	CG_DL	CG_GY	CG_HS*
1 ($n = 400$)	62	67	59	46	41
2 ($n = 400$)	42	103	61	54	45
3 ($n = 400$)	76	270	48	81	53
4 ($n = 400$)	73	289	87	61	64
5 ($n = 400$)	33805	NC	33407	391	277
6 ($n = 400$)	NC	NC	NC	416	333
7 ($n = 400$)	NC	NC	NC	414	295
8 ($n = 400$)	NC	NC	NC	859	307
9 ($n = 400$)	23	39	21	12	12
10 ($n = 400$)	23	39	21	12	12
11 ($n = 400$)	23	39	21	12	12
12 ($n = 400$)	209	664	134	186	77
13 ($n = 400$)	246	255	121	140	70
14 ($n = 400$)	184	362	262	174	92
15 ($n = 500$)	296	411	147	162	130
16 ($n = 900$)	116	131	67	57	44
17 ($n = 900$)	NC	821	78	498	7
18 ($n = 900$)	557	NC	12	30	29
19 ($n = 900$)	126	188	122	104	107
20 ($n = 900$)	1	1	1	1	1
21 ($n = 900$)	NC	NC	100	54	NC
22 ($n = 900$)	91	93	88	74	74

23 ($n = 900$)	NC	NC	1678	NC	1646
24 ($n = 900$)	465	512	NC	188	463
25 ($n = 900$)	NC	NC	281	NC	272
26 ($n = 900$)	NC	NC	1115	642	512
27 ($n = 900$)	NC	NC	1471	NC	1018
28 ($n = 900$)	45	71	38	39	51
29 ($n = 900$)	NC	61	NC	34	31
30 ($n = 1000$)	68	87	65	43	59
31 ($n = 1600$)	NC	327	81	332	8
32 ($n = 1600$)	NC	84	14	NC	NC
33 ($n = 1600$)	NC	NC	437	98	99
34 ($n = 1600$)	1	1	1	1	1
35 ($n = 1600$)	NC	NC	448	60	36
36 ($n = 1600$)	132	192	113	109	106
37 ($n = 1600$)	NC	NC	NC	NC	NC
38 ($n = 1600$)	551	545	419	333	605
39 ($n = 1600$)	NC	NC	435	NC	400
40 ($n = 1600$)	415	NC	423	681	503
41 ($n = 1600$)	NC	NC	955	NC	339
42 ($n = 1600$)	48	71	37	38	39
43 ($n = 1600$)	NC	229	NC	23	115
44 ($n = 1600$)	NC	77	66	43	112
45 ($n = 2000$)	NC	274	NC	270	8
46 ($n = 2500$)	NC	NC	26	NC	24
47 ($n = 2500$)	NC	182	462	571	109
48 ($n = 2500$)	1	1	1	1	1
49 ($n = 2500$)	NC	NC	1732	49	NC
50 ($n = 2500$)	329	181	129	121	114
51 ($n = 2500$)	NC	NC	NC	NC	NC
52 ($n = 2500$)	NC	NC	NC	235	635
53 ($n = 2500$)	NC	NC	442	1412	300
54 ($n = 2500$)	NC	NC	NC	332	459
55($n = 2500$)	NC	NC	445	NC	613
56 ($n = 2500$)	49	55	40	40	37
57 ($n = 2500$)	NC	NC	NC	44	NC
58 ($n = 2500$)	60	46	55	38	39
59 ($n = 2500$)	18	52	26	41	32

60 ($n = 2500$)	NC	11	NC	11	6
61 ($n = 2500$)	NC	46	3	48	33
62 ($n = 2500$)	70	347	NC	321	71
63 ($n = 2500$)	1024	NC	NC	317	NC
64 ($n = 2500$)	154	NC	117	88	131
65 ($n = 2500$)	679	NC	879	721	514
66 ($n = 2500$)	1	1	1	1	1
67 ($n = 5000$)	44	NC	39	21	45
68 ($n = 5000$)	1808	NC	1880	NC	1954
69 ($n = 5000$)	NC	NC	NC	NC	NC
70 ($n = 15000$)	NC	NC	NC	NC	NC
71 ($n = 15000$)	119	145	84	94	86
72 ($n = 15000$)	1808	NC	1880	NC	1954
73 ($n = 30000$)	165	193	94	112	142
74 ($n = 150000$)	201	247	186	187	190
75 ($n = 150000$)	NC	318	NC	362	208
76 ($n = 300000$)	543	281	97	284	49
77 ($n = 300000$)	27	31	25	19	18
78 ($n = 1000000$)	NC	59	NC	39	29

Tabela 8 – Dados *Performance Profile* conjunto $\mathscr{P}$-Tempo(s). A dimensão n do problema de otimização está entre parêntese após o número do problema. NC é a sigla para **Não Convergência**.

Performance Profile (Tempo (s)) - Problemas Teste					
Algoritmo / Problema	CG_HS	CG_PR	CG_DL	CG_GY	CG_HS*
1 ($n = 400$)	7,02	9,10	6,05	4,93	5,21
2 ($n = 400$)	14,92	57,32	17,74	15,05	14,02
3 ($n = 400$)	76,91	359,88	58,48	72,90	55,31
4 ($n = 400$)	684,13	2910,01	749,26	467,66	473,85
5 ($n = 400$)	39,19	NC	36,99	0,41	0,32
6 ($n = 400$)	NC	NC	NC	1,75	1,48
7 ($n = 400$)	NC	NC	NC	16,49	12,11
8 ($n = 400$)	NC	NC	NC	26,24	27,04
9 ($n = 400$)	39,80	65,86	35,77	18,90	19,77
10 ($n = 400$)	306,60	521,49	278,81	160,17	159,10
11 ($n = 400$)	1188,10	2009,23	1084,48	620,34	622,12

12 ($n = 400$)	31,04	161,19	20,54	25,17	10,67
13 ($n = 400$)	3,80	7,13	1,93	2,07	1,08
14 ($n = 400$)	49,40	178,65	80,54	46,74	25,16
15 ($n = 500$)	269,10	634,77	131,40	143,30	116,00
16 ($n = 900$)	13,40	27,96	9,50	7,04	5,43
17 ($n = 900$)	NC	33,52	4,63	18,36	0,76
18 ($n = 900$)	282,50	NC	3,00	9,47	8,64
19 ($n = 900$)	12,37	36,21	11,18	10,71	10,77
20 ($n = 900$)	0,04	0,04	0,04	0,04	0,04
21 ($n = 900$)	NC	NC	11,52	4,21	NC
22 ($n = 900$)	12,18	22,04	12,49	10,37	10,37
23 ($n = 900$)	NC	NC	264,93	NC	267,49
24 ($n = 900$)	201,75	462,16	NC	77,12	181,97
25 ($n = 900$)	NC	NC	0,40	NC	0,36
26 ($n = 900$)	NC	NC	306,21	91,81	70,35
27 ($n = 900$)	NC	NC	270,16	NC	187,52
28 ($n = 900$)	13,35	71,98	10,47	10,72	14,58
29 ($n = 900$)	NC	134,59	NC	49,69	49,87
30 ($n = 1000$)	36,91	88,83	36,83	20,59	29,23
31 ($n = 1600$)	NC	36,73	10,42	29,78	1,68
32 ($n = 1600$)	NC	103,83	9,15	NC	NC
33 ($n = 1600$)	NC	NC	91,81	21,70	21,19
34 ($n = 1600$)	0,09	0,09	0,09	0,09	0,09
35 ($n = 1600$)	NC	NC	122,75	9,83	7,96
36 ($n = 1600$)	68,40	178,01	53,04	50,62	52,14
37 ($n = 1600$)	NC	NC	NC	NC	NC
38 ($n = 1600$)	533,81	1062,59	382,69	294,75	546,80
39 ($n = 1600$)	NC	NC	0,58	NC	0,58
40 ($n = 1600$)	134,53	NC	133,61	208,86	159,75
41 ($n = 1600$)	NC	NC	468,95	NC	156,02
42 ($n = 1600$)	29,22	166,61	21,61	21,88	23,51
43 ($n = 1600$)	NC	943,24	NC	100,73	367,69
44 ($n = 1600$)	NC	204,59	89,96	62,07	109,25
45 ($n = 2000$)	NC	45,92	NC	41,14	3,18
46 ($n = 2500$)	NC	NC	40,64	NC	42,45
47 ($n = 2500$)	NC	126,21	170,84	208,71	41,58
48 ($n = 2500$)	0,17	0,16	0,16	0,16	0,16

49 ($n = 2500$)	NC	NC	751,49	13,63	NC
50 ($n = 2500$)	392,03	316,06	144,93	133,55	127,65
51 ($n = 2500$)	NC	NC	NC	NC	NC
52 ($n = 2500$)	NC	NC	NC	355,87	945,86
53 ($n = 2500$)	NC	NC	0,77	1,40	0,64
54 ($n = 2500$)	NC	NC	NC	458,58	233,31
55 ($n = 2500$)	NC	NC	320,39	NC	410,41
56 ($n = 2500$)	52,93	177,83	46,49	42,17	39,28
57 ($n = 2500$)	NC	NC	NC	305,28	NC
58 ($n = 2500$)	556,30	514,76	466,03	292,84	317,60
59 ($n = 2500$)	51,19	228,64	77,81	106,79	83,65
60 ($n = 2500$)	NC	64,80	NC	22,17	13,91
61 ($n = 2500$)	NC	598,18	15,31	458,77	326,33
62 ($n = 2500$)	112,24	962,86	NC	503,67	113,39
63 ($n = 2500$)	2617,68	NC	NC	809,35	NC
64 ($n = 2500$)	2831,86	NC	1736,63	1531,20	1936,93
65 ($n = 2500$)	580,31	NC	770,07	601,88	468,65
66 ($n = 2500$)	0,70	0,70	0,70	0,70	0,70
67 ($n = 5000$)	145,44	NC	73,48	23,37	77,98
68 ($n = 5000$)	1343,39	NC	1358,53	NC	1392,85
69 ($n = 5000$)	NC	NC	NC	NC	NC
70 ($n = 15000$)	NC	NC	NC	NC	NC
71 ($n = 15000$)	862,76	1917,20	591,93	653,65	583,76
72 ($n = 15000$)	1343,39	NC	1358,53	NC	1392,85
73 ($n = 30000$)	1047,88	2452,59	553,77	658,85	830,13
74 ($n = 150000$)	1,63	3,27	1,57	1,54	1,64
75 ($n = 150000$)	NC	1501,17	NC	822,95	473,71
76 ($n = 300000$)	1417,18	976,75	626,53	786,21	318,18
77 ($n = 300000$)	110,42	458,35	97,31	75,28	73,08
78 ($n = 1000000$)	NC	2498,94	NC	1157,33	955,45

Tabela 9 – Dados *Performance Profile* CG x MatLab-Iteração. A dimensão n de todos os problemas é igual a 400 ($n = 400$). NC é a sigla para **Não Convergência**.

Performance Profile (iteração)														
Problema / Algoritmo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
CG_HS	116	NC	557	126	1	NC	91	NC	465	NC	NC	NC	45	NC
CG_PR	131	821	NC	188	1	NC	93	NC	512	NC	NC	NC	71	61
CG_DL	67	78	12	122	1	100	88	1678	NC	281	1115	1471	38	NC
CG_GY	57	498	30	104	1	54	74	NC	188	NC	642	NC	39	34
CG_HS*	44	7	29	107	1	NC	74	1646	463	272	512	1018	51	31
Trust-Region	NC	NC	1	5	9	4	13	NC	13	NC	NC	28	5	NC

Tabela 10 – Dados *Performance Profile* CG x MatLab-Tempo(s). A dimensão n de todos os problemas é igual a 400 ($n = 400$). NC é a sigla para **Não Convergência**.

Performance Profile (Tempo(s))														
Problema / Algoritmo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
CG_HS	13,405	NC	282,504	12,369	0,041	NC	12,185	NC	201,747	NC	NC	NC	13,346	NC
CG_PR	27,965	33,518	NC	36,208	0,040	NC	22,037	NC	462,158	NC	NC	NC	71,977	134,593
CG_DL	9,504	4,630	3,003	11,181	0,040	11,520	12,489	264,930	NC	0,405	306,208	270,164	10,473	NC
CG_GY	7,037	18,364	9,471	10,708	0,041	4,213	10,374	NC	77,117	NC	91,806	NC	10,718	49,685
CG_HS*	5,433	0,756	8,640	10,774	0,042	NC	10,368	267,490	181,970	0,365	70,348	187,520	14,583	49,871
Trust-Region	NC	NC	43,836	29,122	43,664	24,784	166,939	NC	99,771	NC	NC	188,842	41,719	NC