

## Variantes do método do gradiente para otimização irrestrita

<b>Edital:</b>	<b>Edital Piic 2020/2021</b>
<b>Grande Área do Conhecimento (CNPq):</b>	<b>Ciências Exatas e da Terra</b>
<b>Área do Conhecimento (CNPq):</b>	<b>Matemática / Matemática Aplicada</b>
<b>Título do Projeto:</b>	<b>Métodos Computacionais em Otimização (9403/2019)</b>
<b>Título do Subprojeto:</b>	<b>Variantes do método do gradiente para otimização irrestrita</b>
<b>Professor Orientador:</b>	<b>Leonardo Delarmelina Secchin</b>
<b>Estudante:</b>	<b>Elivandro Oliveira Grippa</b>

### Resumo

O método do gradiente é fundamental em otimização irrestrita. Ele baseia-se no fato de a direção de maior decréscimo de uma função continuamente diferenciável ser a contrária ao de seu gradiente. Apesar de seu grande uso em implementações, é demasiadamente lento próximo à solução. Estratégias de aceleração incluem controle do tamanho de passo e ajuste na direção. Em particular, uma variante, chamada q-gradiente, foi proposta na literatura. Outra variante, o método do gradiente ponderado com atraso, aplica-se com admirável sucesso à funções quadráticas estritamente convexas, mostrando vantagens numéricas mesmo sobre o método dos gradientes conjugados. Também foram considerados os métodos do gradiente espectral e a generalização de gradientes conjugados para não quadráticas proposta por Hager e Zhang, CG-descent. Testes numéricos comprovaram que o método gradiente ponderado com atraso tem um desempenho superior ao do método de gradientes conjugados clássico, para problemas cuja hessiana tem um número grande de não zeros. Já para funções gerais, o melhor método é o do gradiente espectral, tendo eficácia e velocidade superiores aos métodos do gradiente e CG-descent. Nesse trabalho, comparamos, teórico e computacionalmente, os métodos citados. Ademais, analisamos uma adaptação do gradiente ponderado com atraso à problemas irrestritos gerais.

**Palavras chaves:** Programação Não-Linear, Método do gradiente e variantes, Gradientes conjugados, gradiente ponderado com atraso. Gradiente espectral.

### 1 Introdução

Nesta pesquisa, estamos interessados em resolver problemas de otimização da forma

$$\text{Min } \{f(x); x \in \mathbb{R}^n\}, \quad (1)$$

onde  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  é continuamente diferenciável, e cujo gradiente em  $x$  é denotado por  $\nabla f(x)$ . A resolução de tais problemas possui grande importância na área de otimização, devido a vários pacotes computacionais eficientes precisarem resolver subproblemas irrestritos.

Uma estratégia geral de otimização é considerar um ponto inicial  $x_0$  e aplicar um processo iterativo dado por

$$x_{k+1} = x_k + \alpha_k d_k, \quad (2)$$

onde  $k$  é o índice da iteração,  $x_k$  é o vetor de variáveis,  $\alpha_k > 0$  é o comprimento do passo ou distância percorrida ao longo da direção de busca  $d_k$ . O tamanho do passo é normalmente determinado por busca inexata, como a condição de Armijo, que nos garante um decréscimo suficiente da função. Para cada escolha de  $d_k$  temos um novo método. Por exemplo, para  $d_k = -\nabla f(x_k)$  chegamos ao clássico método do gradiente ou de máxima descida.

Dentre os métodos estudados temos o do q-gradiente, onde substituímos o gradiente usual por uma outra noção de gradientes. Outro método, recentemente apresentado, é o gradiente ponderado com atraso (DWGM) que, assim como o gradientes conjugados (CG), é aplicado à problemas quadráticos estritamente convexos e irrestritos. Sua iteração inicial é como a do gradiente, e nas iterações seguintes é feita uma correção utilizando o iterando anterior (atraso). Na seção 5 discutiremos com mais detalhes os métodos considerados.

Este trabalho objetivou comparar os métodos citados acima e outras variantes do gradiente, tanto do ponto de vista teórico quanto computacional, e estudar uma possível adaptação do DWGM para problemas gerais.

Por fim, vale lembrar que este subprojeto se enquadra no projeto de “**Métodos computacionais em otimização**”, sob registro PRPPG 9403/2019, no qual o orientador é coordenador. Um dos objetivos do projeto é o estudo de novos modelos e métodos de resolução eficientes para diferentes problemas de otimização.

## 2 Objetivos

---

O objetivo geral desta pesquisa é comparar, teórica e computacionalmente, métodos inspirados no esquema de descida por gradiente. Os métodos inicialmente propostos para tal comparação foram: gradiente, q-gradiente, DWGM e CG. Ainda, estudar possível adaptação/comportamento do DWGM em problemas gerais. Para isso, traçamos os seguintes objetivos:

1. Estudo/revisão dos conceitos fundamentais de otimização sem restrições e dos métodos do gradiente e gradientes conjugados de Hestenes e Stifel, dentre os textos de apoio estão o livro [Ribeiro & Karas, 2013].
2. Estudo dos principais artigos científicos sobre o tema proposto. Em especial [Andreani & Raydan, 2020], [Gouvêa et al., 2016], [Oviedo Leon, 2019] e [Soterroni et al., 2011].
3. Implementação e testes numéricos a fim de comparar os diferentes métodos considerados na pesquisa. Pretende-se realizar testes preliminares com um número reduzido de problemas-teste.
4. Analisar a adaptação/comportamento numérico do método do gradiente ponderado com atraso aplicado a minimização de funções não lineares gerais. A iniciativa não é tratada na literatura. No entanto, levando em conta que o método de gradientes conjugados é aplicado com relativo sucesso a problemas gerais, há uma expectativa de que o novo método tenha um comportamento semelhante.

## 3 Embasamento Teórico

---

Otimização compreende um conjunto de operações por meio das quais busca-se minimizar ou maximizar uma determinada função, que chamaremos *função objetivo*, sujeita ou não a uma série de *restrições*. Muitos esforços têm sido empregados no desenvolvimento de métodos numéricos para resolver tais problemas, que cresceram com o aumento da capacidade computacional. O problema de otimização pode ser definido como

$$\text{Minimizar } f(x) \text{ sujeito a } x \in \Omega \subset \mathbb{R}^n. \quad (3)$$

Neste caso,  $f$  é a função objetivo e  $\Omega$  o conjunto factível, que em muitos dos casos é determinado por um conjunto de equações e/ou inequações. Vale ressaltar que toda teoria é baseada em minimizar funções, pois maximizar uma função  $f$  equivale à minimizar  $-f$  [Martinez & Santos, 1995].

Nesse trabalho, estamos interessados em estudar métodos de otimização para problemas sem restrições, ou seja, consideramos  $\Omega = \mathbb{R}^n$ . Em tais problemas, nosso objetivo é determinar minimizadores locais de  $f(x)$ , visto que a busca por minimizadores globais é difícil.

Dizemos que  $x^* \in \mathbb{R}^n$  é um minimizador local de  $f$  se  $f(x^*) \leq f(x)$  para todo  $x$  em uma dada vizinhança de  $x^*$ . Para que  $x^* \in \mathbb{R}^n$  seja minimizador local, ele deve satisfazer algumas condições, chamadas *condições de otimalidade*. Daremos apenas um destaque à condição de otimalidade de primeira ordem, estabelecida no teorema a seguir [Ribeiro & Karas, 2013].

**Teorema 1.** *Seja  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciável no ponto  $x^* \in \mathbb{R}^n$ . Se  $x^*$  é minimizador local de  $f$ , então  $\nabla f(x^*) = 0$ .*

Aqui, o vetor  $\nabla f$  é o vetor gradiente de  $f$ , cujas componentes são as derivadas parciais de  $f$ , ou seja,

$$\nabla f(x) = \left[ \frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right]^t.$$

Com isso, a condição estabelecida no Teorema (1) garante que se um ponto  $x^*$  for minimizador local da função  $f$ , então  $x^*$  será solução do sistema  $\nabla f(x) = 0$ . Os pontos que satisfazem essa condição são chamados de *pontos críticos* ou *estacionários* de  $f$ , e vale observar que é apenas uma condição necessária, não suficiente. Assim, se  $\nabla f(x^*) = 0$  então  $x^*$  é minimizador, maximizador ou nenhum dos dois. No último caso, eles são chamados *pontos de sela*.

Embora a condição  $\nabla f(x) = 0$  não seja suficiente, a resolução desse sistema faz parte de diversos métodos de otimização irrestrita, pois, pela limitação computacional, não se pode utilizar a definição de minimizador por vizinhanças. Porém,  $\nabla f(x) = 0$  dificilmente é resolvido por métodos diretos. Usualmente são empregados processos iterativos, em que a partir de um ponto inicial  $x_0$ , geramos uma sequência  $\{x_k\} \subset \mathbb{R}^n$  onde a função  $f$  decresça.

Um algoritmo básico de otimização irrestrita, muito conhecido na literatura, consiste em, a partir de cada ponto obtido, determinar uma direção e dar um passo ao longo dela. Como o objetivo é minimizar a função  $f$ , é coerente que a função decresça localmente na direção escolhida. Chamamos uma direção com tal propriedade de *direção de descida*. Matematicamente, dizemos que  $d \in \mathbb{R}^n \setminus \{0\}$  é uma direção de descida para  $f$ , a partir do ponto  $\bar{x}$ , quando existe  $\delta > 0$  tal que  $f(\bar{x} + td) < f(\bar{x})$  para todo  $t \in (0, \delta)$ . Outra maneira mais intuitiva de se definir uma direção de descida, é através da seguinte condição suficiente [Ribeiro & Karas, 2013]: se  $\nabla f(\bar{x})^t d < 0$  então  $d$  é uma direção de descida para  $f$  a partir de  $\bar{x}$ .

Após a escolha de uma direção de descida, é necessário definir o quanto caminhar, ou seja, determinar o tamanho do passo que devemos dar. Métodos empregados para esse fim são chamados *métodos de busca linear*. O que diferencia os métodos de otimização irrestrita é a escolha da direção de descida  $d$  e a forma com que é calculado o tamanho do passo a ser dado.

O estudo de métodos de otimização irrestrita possui grande importância na área de otimização, visto que muitos pacotes computacionais para otimização restrita necessitam resolver subproblemas irrestritos, além da existência de importantes aplicações dos modelos irrestritos em si. Ao longo da pesquisa foram utilizados diversos artigos e

livros. Dentre eles, vale ressaltar os livros de Ribeiro & Karas, 2013 e Nocedal & Wright, 2006, que foram usados como base para revisão dos conceitos de otimização sem restrições, e estudo dos principais métodos de descida, como o do gradiente para funções gerais e do gradientes conjugados para funções quadráticas. Foram estudados os métodos do gradiente, q-gradiente [Soterroni et al., 2011, Gouvêa et al., 2016], gradiente espectral [Raydan, 1997] e gradientes conjugados de Hager e Zhang [Hager & Zhang, 2005] para funções gerais, e o método dos gradientes conjugados e gradiente ponderado com atraso [Oviedo Leon, 2019] para funções quadráticas estritamente convexas.

## 4 Metodologia

---

O desenvolvimento da pesquisa se deu por reuniões remotas semanais com o orientador, em que foi discutido a leitura dos artigos e possíveis dúvidas em implementações. A linguagem de programação escolhida foi Julia (<https://julialang.org/>) devido a ser uma linguagem de alto nível e desempenho, e possuir diversos pacotes voltados para análise numérica e otimização.

Para a fase de testes numéricos foi utilizado o servidor do grupo de pesquisa, e os problemas testes obtidos das bibliotecas [Gould et al., 2015] e [Davis & Hu, 2011]. Os resultados foram analisados e expostos em forma de gráficos de perfis de desempenho em relação ao tempo computacional e número de iterações, sendo possível fazer uma boa comparação dos métodos estudados.

## 5 Resultados e Discussão

---

Além dos métodos indicados na concepção do subprojeto, listados na seção de objetivos, consideramos durante o desenvolvimento do trabalho o método do gradiente espectral [Raydan, 1997] e uma generalização do gradientes conjugados chamada de CG-descent [Hager & Zhang, 2005]. A seguir descrevemos brevemente cada um dos métodos considerados.

### 5.1 Métodos para minimização de funções gerais

#### 5.1.1 Gradiente

Um dos métodos mais clássicos para minimizar uma função é o método do gradiente [Ribeiro & Karas, 2013]. Neste método, a direção de descida é dada pela direção oposta ao do vetor gradiente. Ou seja, basta tomar em (2)  $d_k = -\nabla f(x_k)$  e teríamos o processo iterativo abaixo

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad (4)$$

Tal escolha se deve ao fato de que, dentre todas as possíveis direções em que a função decresce, a direção oposta ao do gradiente é a de maior decréscimo, fazendo com que a escolha (4) seja razoável ao se minimizar uma função.

A convergência do método do gradiente é garantida tomando o tamanho de passo  $\alpha_k$  por busca inexata, como a condição de Armijo, que garanta um decréscimo suficiente da função ao longo da direção  $d_k$ . A seguir, apresentamos o algoritmo associado ao método do gradiente conforme apresentado em [Ribeiro & Karas, 2013] usando busca de Armijo com *Backtracking*.

---

**Algoritmo 1** Método do gradiente com busca de Armijo

---

- 1: Dados:  $x_0 \in \mathbb{R}^n$ ,  $\gamma \in (0, 1)$ ,  $\eta \in (0, 1)$ , tolerância  $tol$ .
  - 2: **para**  $\|\nabla f(x_k)\| > tol$  **faça**
  - 3:      $d_k = -\nabla f(x_k)$ ,
  - 4:     Tamanho do passo por Armijo:  $\alpha_k = 1$
  - 5:     Reduza  $\alpha_k$  até que  $f(x_k + \alpha_k d_k) \leq f(x_k) + \eta \alpha_k \nabla f(x_k)^T d_k$  utilizando *Backtracking* e interpolação quadrática
  - 6:      $x_{k+1} = x_k + \alpha_k d_k$ ,
  - 7:      $k = k + 1$ .
  - 8: **fim para**
- 

### 5.1.2 Q-gradiente

Nessa seção, apresentaremos o método do q-gradiente [Soterroni et al., 2011]. Para entendermos o método é preciso definir o vetor q-gradiente, que de certa forma é similar ao vetor gradiente clássico, mas em vez de usarmos as derivadas parciais de primeira ordem mostradas no Cálculo, usamos as q-derivadas parciais de primeira ordem obtidas das Derivadas de Jackson, comumente referida como q-derivada. Inicialmente, apresentaremos a definição da q-derivada para o caso de funções de uma variável e depois sua generalização.

Seja  $g : \mathbb{R} \rightarrow \mathbb{R}$  contínua e diferenciável em  $x = 0$ , a q-derivada de  $g(x)$  ( $q \neq 1$ ) é dada por

$$D_q g(x) = \begin{cases} \frac{g(x) - g(qx)}{(1-q)x}, & \text{se } x \neq 0, \\ g'(0), & \text{se } x = 0. \end{cases} \quad (5)$$

Onde parâmetro  $q$  é normalmente fixado como um número real  $0 < q < 1$ , mas pode ser qualquer número real diferente de 1. Note que no limite, quando  $q \rightarrow 1$ , a q-derivada tende a derivada clássica. Ou seja,

$$\lim_{q \rightarrow 1} D_q g(x) = \lim_{q \rightarrow 1} \frac{g(x) - g(qx)}{(1-q)x} = \lim_{y \rightarrow x} \frac{g(x) - g(y)}{x-y} = g'(x). \quad (6)$$

Agora considere  $g(x)$  uma função diferenciável de  $n$  variáveis. Soterroni et al., 2011 definiu a q-derivada parcial de primeira ordem de  $g(x)$  em relação a  $x_i$  por

$$D_{q,x_i} g(x) = \begin{cases} \frac{g(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) - g(x_1, \dots, x_{i-1}, qx_i, x_{i+1}, \dots, x_n)}{(1-q)x_i}, & \text{se } x_i \neq 0, q \neq 1, \\ \frac{\partial g}{\partial x_i}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n), & \text{caso contrário.} \end{cases} \quad (7)$$

Além disso, dado um vetor  $q = (q_1, q_2, \dots, q_n) \in \mathbb{R}^n$  o vetor q-gradiente de  $g(x)$  é definido para ser o vetor das  $n$   $q_i$ -derivadas parciais de primeira ordem:

$$\nabla_q g(x) = \left[ D_{q_1, x_1} g(x) \quad \cdots \quad D_{q_i, x_i} g(x) \quad \cdots \quad D_{q_n, x_n} g(x) \right]^T. \quad (8)$$

Com isso, podemos definir o Método de descida do q-gradiente para resolver (1). Basta tomar em (2)  $d_k = -\nabla_q f(x_k)$  e teríamos o processo iterativo a seguir.

$$x_{k+1} = x_k - \alpha_k \nabla_q f(x_k) \quad (9)$$

O tamanho de passo  $\alpha_k$  pode ser determinado por busca inexata, como a condição de Armijo. Outra estratégia usada é determinar os parâmetros  $q_i (i = 1, \dots, n)$ , a partir de números aleatórios, extraídos de uma distribuição log-normal com média  $\mu = 1$  e desvio padrão  $\sigma$ , e ao longo de cada iteração tomar  $\sigma_{k+1} = \beta \sigma_k$  sendo  $0 < \beta < 1$ . Isso faz com que os valores de  $q_i \rightarrow 1$  levando a  $\nabla_{qf}(x_k) \rightarrow \nabla f(x_k)$ . A convergência dos métodos variantes do q-gradiente pode ser encontrada em [Gouvêa et al., 2016], onde perturbações gaussianas são utilizadas. A seguir, apresentamos o algoritmo associado ao método do q-gradiente conforme proposto em [Soterroni et al., 2011] com busca pelo método de Armijo com *Backtracking*.

---

**Algoritmo 2** Método do q-gradiente com busca de Armijo

---

- 1: Dados:  $x_0 \in \mathbb{R}^n$ ,  $\mu = 1$ ,  $\sigma_0$  e  $\beta$ ,  $\gamma \in (0, 1)$ ,  $\eta \in (0, 1)$ .
  - 2: **para**  $\|\nabla f(x_k)\|_2 > tol$  **faça**
  - 3:   Gere as entradas do vetor  $q = (q_1, q_2, \dots, q_n)^T$  de uma distribuição log-normal, com média  $\mu$  e desvio padrão  $\sigma_k$ .
  - 4:    $d_k = -\nabla_{qf}(x_k)$ ,
  - 5:   Tamanho do passo por Armijo:  $\alpha_k = 1$
  - 6:   Reduza  $\alpha_k$  até que  $f(x_k + \alpha_k d_k) \leq f(x_k) + \eta \alpha_k \nabla f(x_k)^T d_k$  utilizando *Backtracking* e interpolação quadrática
  - 7:    $x_{k+1} = x_k + \alpha_k d_k$ ,
  - 8:    $\sigma_{k+1} = \beta \sigma_k$
  - 9:    $k = k + 1$ .
  - 10: **fim para**
- 

### 5.1.3 Gradiente espectral

Nessa seção apresentaremos o método do Gradiente Espectral (SG), descrito em [Raydan, 1997], para problemas de otimização irrestrita da forma (1). O método SG inspira-se nos métodos quase-Newton tipo secante. Nesses métodos, aplicamos o seguinte processo iterativo

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k), \quad (10)$$

onde a sequência de matrizes  $\{B_k\}$  satisfaz a equação secante

$$B_{k+1} s_k = y_k, \quad (11)$$

sendo  $s_k = x_k - x_{k+1}$  e  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ . No método SG assumiremos que a matriz  $B_{k+1}$  possua uma estrutura simples e satisfaça a Equação (11). De maneira mais específica, impomos que

$$B_{k+1} = \sigma_{k+1} I, \quad (12)$$

sendo  $\sigma_{k+1} \in \mathbb{R}$ , então da equação (11) temos

$$\sigma_{k+1} s_k = y_k. \quad (13)$$

Repare que não há garantia de que a Eq. (13) possua solução, logo estamos interessados em determinar o  $\alpha_{k+1}$  que melhor a aproxima. Portanto, uma boa aproximação para  $\alpha_{k+1}$  é obtida a partir da solução de um problema de

quadrados mínimos da forma  $\text{Min} \{ \|\sigma s_k - y_k\|_2^2; \sigma \in \mathbb{R} \}$ , cuja solução

$$\sigma_{k+1} = \frac{s_k^T y_k}{s_k^T s_k} \quad (14)$$

é facilmente calculada.

O método SG tem convergência garantida usando uma versão modificada da condição de Armijo, que realiza uma busca não monótona [Grippo et al., 1986] para o cálculo do tamanho do passo  $\alpha_k$ . Ela requer decréscimo suficiente de  $f(x)$  apenas em ciclos de iterações, não em toda iteração como na condição original.

### 5.1.4 Gradientes conjugados de Hager e Zhang

O método dos gradientes conjugados proposto por Hager & Zhang, 2005 (CG-descent), consiste em uma generalização, para funções não quadráticas, do clássico método de gradientes conjugados (CG). Tomando  $x_0 \in \mathbb{R}^n$  e  $d_0 = -\nabla f(x_0)$ , o método consiste em utilizar o processo iterativo da Equação (2) tomando

$$d_{k+1} = -\nabla f(x_{k+1}) + \beta_k^N d_k, \quad (15)$$

$$\beta_k^N = \frac{1}{d_k^T y_k} \left( y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T \nabla f(x_{k+1}), \quad (16)$$

onde  $\|\cdot\|$  é norma Euclidiana e  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ . O tamanho de passo  $\alpha_k$  é obtido por uma busca inexata que cumpra as condições de Wolfe, devido à, unicamente, Armijo não ser suficiente para garantir a convergência. Nas condições de Wolfe, além de exigir a condição de Armijo, acrescenta-se uma condição de curvatura, que descarta a possibilidade da obtenção de tamanhos de passo suficientemente pequenos.

Vale ressaltar que caso  $f(x)$  seja quadrática e o tamanho de passo  $\alpha_k$  seja obtido minimizando a função ao longo da direção  $d_k$ , então teríamos que  $\nabla f(x_{k+1})^T d_k = d_k^T \nabla f(x_{k+1}) = 0$ , e o método se resumiria ao método CG clássico. A demonstração da convergência do método, para funções não quadráticas gerais pode ser encontrada em [Hager & Zhang, 2005].

## 5.2 Métodos para minimização de funções quadráticas

Agora estamos interessados em resolver (1) considerando  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  quadrática. Ou seja,

$$f(x) = \frac{1}{2} x^T A x - b^T x, \quad (17)$$

onde  $b \in \mathbb{R}^n$  e  $A \in \mathbb{R}^{n \times n}$  são dados. A seguir apresentaremos métodos para resolver tais problemas.

### 5.2.1 Gradientes Conjugados

O método de gradientes conjugados é um caso particular do método de direções conjugadas [Ribeiro & Karas, 2013], que consiste em um método iterativo o qual estamos interessados em resolver o sistema

$$Cx = b, \quad (18)$$

onde  $C \in \mathbb{R}^{n \times n}$  é simétrica e definida positiva ( $C^T = C$  e  $x^T C x > 0 \ \forall x \neq 0 \in \mathbb{R}^n$ ). Se considerarmos a matriz  $A$  em (17) com as mesmas especificações de  $C$ , teríamos que  $f(x)$  é uma função estritamente convexa e  $\nabla f(x) = Ax - b = 0$  é suficiente para resolvermos (1). Levando em conta que este método é muito conhecido na literatura, faremos uma breve apresentação dos seus principais aspectos.

No método CG fazemos atualizações da direção de busca levando em conta a propriedade de  $A$ -conjugação. De maneira mais específica, dado  $x_0 \in \mathbb{R}^n$  e  $d_0 = -\nabla f(x_0)$  definimos

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (19)$$

sendo

$$\beta_k = \frac{d_k^T A \nabla f(x_{k+1})}{d_k^T A d_k}, \quad (20)$$

que é determinado de modo que  $d_k$  e  $d_{k+1}$  sejam  $A$ -conjugadas (i.e.,  $d_k^T A d_{k+1} = 0$ ). O cálculo do  $\beta_k$ , dada por (20), muitas vezes pode ser caro devido aos produtos matriz-vetor. Portanto, apresentamos outras maneiras de se calcular este coeficiente. Uma delas é a proposta por Polak e Ribiere, dada por

$$\beta_k^{PR} = \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla f(x_k)^T \nabla f(x_k)}. \quad (21)$$

Outra é a definida por Fletcher e Reeves

$$\beta_k^{FR} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)}. \quad (22)$$

Vale ressaltar, que no caso em que a função  $f$  é quadrática, temos  $\beta_k = \beta_k^{PR} = \beta_k^{FR}$  (Teorema (5.20) de [Ribeiro & Karas, 2013]). Devido a  $f(x)$  ser uma função estritamente convexa, o tamanho de passo  $\alpha_k = \min\{f(x_k + \alpha d_k); \alpha \in \mathbb{R}\}$  pode ser obtido de maneira exata anulando a derivada de  $f$ , obtendo assim

$$\alpha_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T A d_k}. \quad (23)$$

O método dos gradientes conjugados é muito conhecido por sua convergência ser mais rápida que o método do gradiente e ter um custo computacional inferior ao método de Newton. Outra propriedade relevante é a sua convergência finita, em no máximo  $n$  passos (na verdade, o número de autovalores distintos de  $A$ ), quando se minimiza uma quadrática definida em  $\mathbb{R}^n$ .

### 5.2.2 Gradiente Ponderado com atraso

O método do gradiente ponderado com atraso (DWGM) [Oviedo Leon, 2019] é um método de dois passos, em que é inicialmente feito uma iteração do método do gradiente, para algum tamanho de passo, e logo após uma correção no ponto obtido. Sendo assim, a ideia do método é gerar uma sequência auxiliar  $\{y_k\}$  da forma

$$y_k = x_k - \alpha_k \nabla f(x_k), \quad (24)$$



---

**Algoritmo 3** Método dos gradientes conjugados

---

- 1: Dados:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $x_0 \in \mathbb{R}^n$  e uma tolerância  $tol$ , faça  $g_0 = \nabla f(x_0)$ ,  $d_0 = -g_0$
  - 2: **enquanto**  $\|\nabla f(x_k)\|_2 > tol$  **faça**
  - 3:      $w_k = Ad_k$ ,
  - 4:      $z_k = d_k^T w_k$ ,
  - 5:      $\alpha_k = -\frac{g_k^T d_k}{z_k}$ ,
  - 6:      $x_{k+1} = x_k + \alpha_k d_k$ ,
  - 7:      $g_{k+1} = g_k + \alpha_k z_k$  ( $g_{k+1} = \nabla f(x_{k+1})$ )
  - 8:      $\beta_k = \frac{w_k^T g_{k+1}}{z_k}$ ,
  - 9:      $d_{k+1} = -g_{k+1} + \beta_k d_k$ ,
  - 10:     $k \leftarrow k + 1$ .
  - 11: **fim enquanto**
- 

em que

$$\alpha_k = \min \{ \|\nabla f(x_k - \alpha \nabla f(x_k))\|_2; \alpha > 0 \in \mathbb{R} \}. \quad (25)$$

e, a partir disso, é feito uma correção em  $y_k$  juntamente com o iterando  $x_{k-1}$  (atraso), tomando

$$x_{k+1} = x_{k-1} + \beta_k (y_k - x_{k-1}), \quad (26)$$

sendo  $\beta_k$  dado por

$$\beta_k = \min \{ \|\nabla f(x_{k-1} + \beta (y_k - x_{k-1}))\|_2; \beta \in \mathbb{R} \}. \quad (27)$$

É simples mostrar que os tamanhos de passo (25) e (27), para  $f(x)$  quadrática, são respectivamente

$$\alpha_k = \frac{\nabla f(x_k)^T A \nabla f(x_k)}{\nabla f(x_k)^T A^2 \nabla f(x_k)} \quad \text{e} \quad \beta_k = \frac{\nabla f(x_{k-1})^T (\nabla f(x_{k-1}) - \nabla f(y_k))}{\|\nabla f(x_{k-1}) - \nabla f(y_k)\|_2^2}.$$

O método do gradiente ponderado com atraso possui diversas propriedades, entre elas a convergência finita, assim como o CG, em no máximo  $p$  iterações sendo  $p$  o número de autovalores na matriz  $A$ . Uma análise teórica para o DWGM se encontra em [Andreani & Raydan, 2020], em que aborda diversas propriedades do DWGM, inclusive sua convergência.

---

**Algoritmo 4** Método do gradiente ponderado com atraso (DWGM)

---

- 1: Dados:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $x_{-1} = x_0$ ,  $g_0 = \nabla f(x_0)$ ,  $g_{-1} = g_0$ ,  $k = 0$ .
  - 2: **para**  $\|\nabla f(x_k)\|_2 > tol$  **faça**
  - 3:      $w_k = Ag_k$ ,
  - 4:      $\alpha_k = \frac{g_k^T w_k}{w_k^T w_k}$ ,
  - 5:      $y_k = x_k - \alpha_k g_k$ ,
  - 6:      $r_k = g_k - \alpha_k w_k$ ,
  - 7:      $\beta_k = \frac{g_{k-1}^T (g_{k-1} - r_k)}{\|g_{k-1} - r_k\|_2^2}$ ,
  - 8:      $x_{k+1} = x_{k-1} + \beta_k (y_k - x_{k-1})$ ,
  - 9:      $g_{k+1} = g_{k-1} + \beta_k (r_k - g_{k-1})$ ,
  - 10:     $k = k + 1$
  - 11: **fim para**
-

### 5.3 Uma adaptação do método do gradiente ponderado com atraso à funções gerais

Ao longo do estudo do método do gradiente ponderado com atraso, notou-se sua grande eficiência para problemas quadráticos estritamente convexos, principalmente aqueles em que a hessiana é mal-condicionada. Logo, como houve sucessos na adaptação do CG para não quadráticas com o desenvolvimento do método CG-descent [Hager & Zhang, 2005], tornou-se interessante uma possível adaptação do DWGM para funções gerais. Tal iniciativa não é tratada na literatura, e se trata de uma proposta em andamento no âmbito do projeto de pesquisa do orientador com colaboradores externos, que ainda não foi submetida a periódico.

### 5.4 Testes computacionais

Na presente seção, faremos uma comparação do comportamento numérico dos métodos, para funções quadráticas e não quadráticas. Sendo todos implementados na linguagem Julia versão 1.5.3 (2020-11-09) e testados no servidor do grupo de pesquisa do projeto, apoiado pela FAPES 116/2019, com as seguintes especificações: sistema GNU/Linux Ubuntu Server 20.04.2 LTS, 1 processador Intel(R) Xeon(R) Silver 4114 CPU 2.20 GHz 10 núcleos (20 threads), 160 Gb RAM.

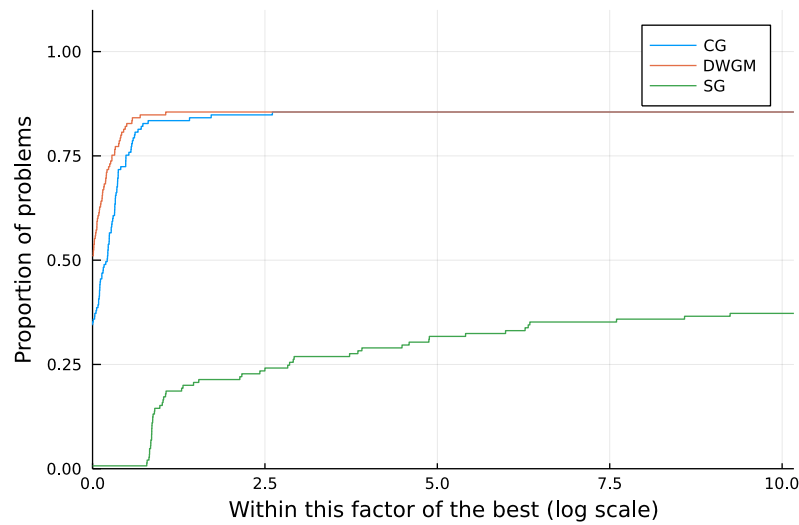
Para a comparação, foram considerados os problemas das bibliotecas CUTEst [Gould et al., 2015], para problemas gerais de otimização irrestrita, e as matrizes da coletânea da Universidade da Flórida [Davis & Hu, 2011], para problemas quadráticos estritamente convexos. Como critério de parada usamos uma tolerância de  $10^{-7}$  para a norma do máximo do gradiente e o número máximo de iterações  $5 \times 10^4$ . Foram utilizados os perfis de desempenho propostos por Dolan & Moré, 2002, gerados a partir do pacote BenchmarkProfiles (<https://github.com/JuliaSmoothOptimizers/BenchmarkProfiles.jl>) da linguagem Julia. Em tais perfis é feito a montagem de um gráfico, para cada método, da fração de problemas resolvidos pelo método dentro de um fator relativo ao melhor algoritmo, podendo ser o tempo ou número de iterações.

#### 5.4.1 Funções quadráticas

Iniciamos os métodos no ponto inicial  $x_0 = (0, \dots, 0)^t$ . As matrizes  $A \in \mathbb{R}^{n \times n}$ , simétricas e definidas positivas, foram obtidas em formato esparsa da coletânea da Universidade da Flórida [Davis & Hu, 2011], através do pacote SuiteSparseMatrixCollection (<https://github.com/JuliaSmoothOptimizers/SuiteSparseMatrixCollection.jl>). O vetor  $b$  de termos independentes, foi gerado a partir de uma randomização da forma  $A \cdot rand(n, 1)$ , em que  $rand(n, 1)$  é um vetor com  $n$  em  $[0, 1]$  ( $i = 1, \dots, n$ ). Foram selecionadas as matrizes com número de não zeros inferior a  $10^6$  e com número de linhas superior 100.

O algoritmo implementado para o método CG é o Algoritmo 3, como em [Ribeiro & Karas, 2013], que utiliza o  $\beta_k$  dado na Equação (20). Foram feitas algumas modificações algébricas para o código, para tornar o método menos custoso, reduzindo os produtos matriz-vetor a um por iteração. Para o DWGM foi utilizado o Algoritmo (4) [Oviedo Leon, 2019], que se compara à modificação do CG com relação a custo por iteração. Além desses dois métodos, consideramos uma adaptação do método do gradiente espectral para quadráticas, baseado na implementação feita em Fortran do projeto TANGO (<https://www.ime.usp.br/~egbirgin/tango>). O principal motivo para se considerar o último, é devido a sua grande eficiência em problemas não quadráticos superando em muito o clássico método do gradiente, logo, é interessante analisar seu comportamento para problemas quadráticos.

Figura 1: Perfil de desempenho dos algoritmos propostos para funções quadráticas com relação ao tempo computacional.



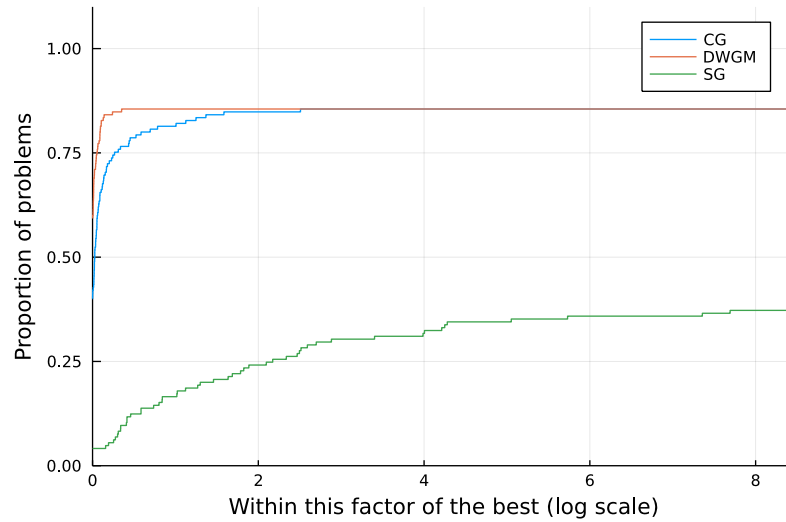
Nas Figuras 1 e 2 apresentamos os perfis de desempenho com relação ao tempo computacional e número de iterações, respectivamente. Note que o método SG tem um desempenho muito inferior ao CG e DWGM, resolvendo cerca de 37% dos problemas, e sendo mais rápido em quase nenhum deles. Tal fato era de se esperar devido ao SG não ser um método específico para quadráticas. Durante os testes numéricos, reparamos que em muitos casos o SG tende a se aproximar da solução, mas para isso requer mais iterações e, conseqüentemente, um maior tempo computacional. Dentre os métodos apresentados para quadráticas, DWGM teve o melhor desempenho, resolvendo pouco mais de 87% dos problemas, sendo o mais rápido em  $\approx 50\%$  e tendo feito o menor número de iterações em  $\approx 60\%$  deles. Note que para o CG ter a mesma eficácia, leva-se em média em torno 5 vezes mais o tempo do algoritmo mais rápido ( $\log_2 5 \approx 2,32$ , próximo à marca no eixo horizontal dos gráficos, em escala  $\log_2$  seguindo [Dolan & Moré, 2002]) e 5 vezes mais o número de iterações. Devido aos problemas selecionados serem relativamente grandes, com um número considerável de não zeros, nota-se que o DWGM tem um diferencial importante em comparação o CG. Este comportamento está em concordância com o que foi explicado e observado no artigo referente ao método [Oviedo Leon, 2019].

#### 5.4.2 Funções não quadráticas

Nos testes numéricos para funções não quadráticas, foram inicialmente considerados problemas da ordem de 50 a 1000 variáveis da CUTEst [Gould et al., 2015]. Tais problemas foram carregados para o Julia a partir do pacote CUTEst (<https://github.com/JuliaSmoothOptimizers/CUTEst.jl>), em uma estrutura onde o cálculo de derivadas e hessianas são automatizados. Para todos os métodos, o ponto inicial foi obtido do próprio arquivo do problema, e caso não o tenha, tomamos como a origem.

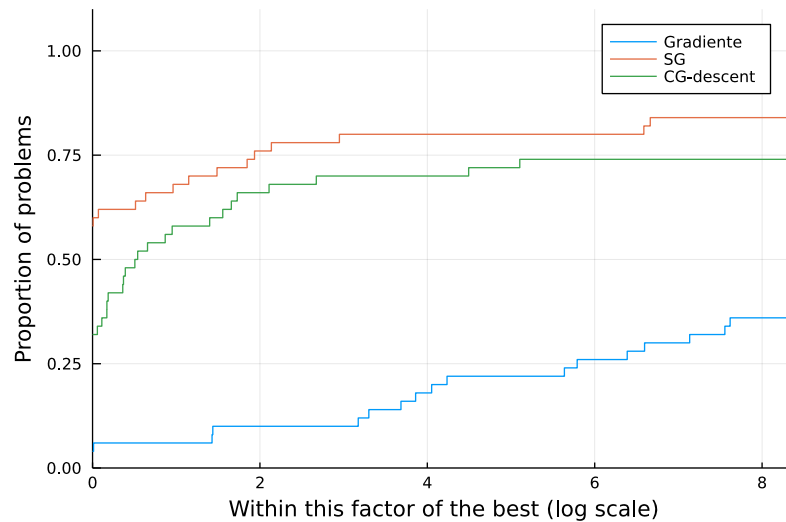
Na implementação do método do gradiente, usou-se o Algoritmo 1 que foi obtido de [Ribeiro & Karas, 2013], considerando o tamanho de passo  $\alpha_k$  obtido por busca inexata (condição de Armijo com  $\eta = 10^{-4}$ ) com interpolação quadrática. Para o método do Gradiente Espectral (SG) implementamos uma tradução do método do Gradiente Espectral Projetado (SPG) fornecido pelo Projeto TANGO (<https://www.ime.usp.br/~egbirgin/tango/>), retirando a projeção sobre restrições de caixa. Essa implementação traz a busca não monótona de [Grippio

Figura 2: Perfil de desempenho dos algoritmos propostos para funções quadráticas com relação ao número de iterações.



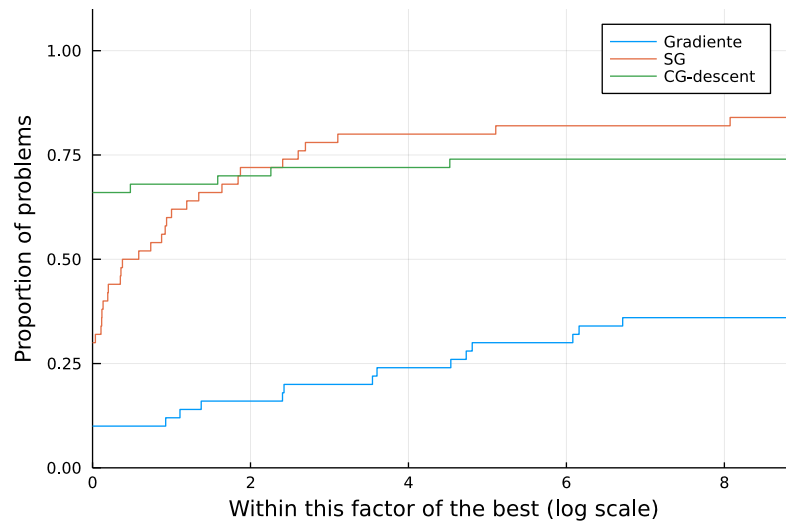
et al., 1986] para SG, tomando  $M = 100$  e  $\gamma = 10^{-4}$ . A implementação do método do q-gradiente se deu pelo Algoritmo 2, que foi proposto em [Soterroni et al., 2011], tomando  $\mu = 1$ ,  $\sigma_0 = 0.5$ ,  $\beta = 0.9$  e o tamanho de passo obtido pela mesma busca linear do método do gradiente. Considerou-se também uma adaptação do gradientes conjugados para não quadráticas, proposto por [Hager & Zhang, 2005], chamado de CG-descent. Para o último, a busca linear usada foi a busca de Hager/Zhang, proposta no artigo referência, a partir do uso do pacote LineSearches (<https://github.com/JuliaNLSolvers/LineSearches.jl>) e o método obtido do pacote Optim (<https://github.com/JuliaNLSolvers/Optim.jl>).

Figura 3: Perfil de desempenho dos métodos do Gradiente, SG e CG-descent com relação ao tempo computacional.



Nas Figuras 3 e 4, apresentamos os perfis de desempenho com relação ao número de iterações e tempo computacional dos métodos do Gradiente, SG e CG-descent. Podemos observar na Figura 3 que o SG foi o método mais eficaz e veloz, resolvendo mais de 80% dos problemas, e sendo o mais rápido em  $\approx 60\%$  deles. O CG-descent ficou como o segundo melhor, sendo pior que o SG em eficácia e velocidade, mas superando em muito o clássico método do gradiente, que teve o pior desempenho no conjunto de problemas selecionados. Na Figura 4, podemos extrair

Figura 4: Perfil de desempenho dos métodos Gradiente, SG e CG-descent com relação ao número de iterações.

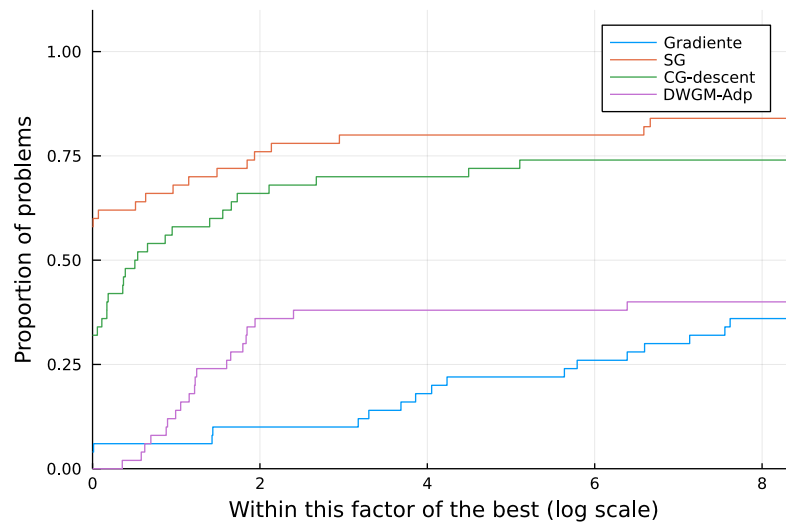


algumas informações importantes: note que o método CG-descent resolve a maior proporção de problemas com poucas iterações, mas perde para o método SG em relação ao tempo. Logo, concluímos que o CG-descent tem um custo maior por iteração, assim levanto um maior tempo de CPU para encontrar a solução; outro ponto importante são as iterações do SG, que são muito baratas computacionalmente. Ou seja, mesmo levando mais iterações para solucionar o problema, acaba por ser um método muito eficiente. Adiante falaremos um pouco sobre o método do q-gradiente e a adaptação, ainda em andamento, do DWGM para não quadráticas.

Testes preliminares mostraram que o método do q-gradiente é pouco efetivo. Para ilustrar, comparamos com o método menos eficiente dos testes anteriores, o do gradiente, em problemas da CUTEst [Gould et al., 2015] com número de variáveis entre 10 e 50. Diante dos testes feitos, chegou-se a conclusão que o método do q-gradiente é ineficaz pois não resolveu nenhum dos problemas propostos, e teve um desempenho muito abaixo do método do gradiente. Vale ressaltar que a implementação do método do q-gradiente se deu pelas especificações da referência [Soterroni et al., 2011], onde os autores apresentam muito poucos testes numéricos (apenas 3 problemas-teste). Nossa ideia, foi efetuar testes mais rigorosos, para verificar as qualidades do método para os problemas irrestritos. A partir disso, chegou-se a conclusão de que o método do q-gradiente, com o algoritmo proposto em [Soterroni et al., 2011], possui propriedades interessantes, mas não apresenta um bom desempenho computacional. É claro que para comprovar sua real ineficácia é preciso fazer uma análise mais rigorosa na implementação e buscar maneiras eficientes de determinar o tamanho do passo, devido à direção de busca nem sempre ser de descida. Esse problema é contornado em [Gouvêa et al., 2016], onde abordam outras propriedades e melhoram o método do q-gradiente. Tal estratégia não foi implementada durante a vigência do projeto.

Finalmente, tratamos da adaptação, ainda em andamento, do método DWGM para funções não quadráticas. Nos testes numéricos foram considerados os mesmos problemas, inicialmente propostos para o Gradiente, SG e CG-descent. Na Figura 5, apresentamos o perfil de desempenho com relação ao tempo computacional, comparando a adaptação do DWGM (DWGM-Adp) com os demais métodos apresentados. Note que sua eficácia foi superior ao do gradiente, mas em velocidade foi o pior de todos os métodos. Vale lembrar que o método ainda está em fase de desenvolvimento, e tem muito espaço para melhorias.

Figura 5: Perfil de desempenho dos métodos Gradiente, SG, CG-descent e DWGM-Adp com relação ao tempo computacional.



## 6 Conclusões

No desenvolver da pesquisa, fizemos o estudo teórico e computacional de diversas variantes do método do gradiente. Inicialmente, executamos uma revisão dos conceitos fundamentais dos métodos de descida e o estudo do clássico método do gradiente tanto do ponto de vista teórico, com sua convergência, quanto do computacional. Logo após, foi feito o estudo de um método pouco conhecido, chamado de q-gradiente, com algumas propriedades teóricas interessantes, mas notamos, a partir de testes numéricos e análises de desempenho, que não é um método eficaz para problemas irrestritos. Para problemas irrestritos gerais, também foram considerados os métodos do gradiente espectral (SG) e gradientes conjugados de Hager e Zhang (CG-descent) [Hager & Zhang, 2005], que tiveram um ótimo desempenho computacional nos problemas-testes. Para o caso de funções quadráticas estritamente convexas, fizemos o estudo do método de gradientes conjugados (CG) e gradiente ponderado com atraso (DWGM) com uma adaptação do SG para quadráticas. Nesse caso, o DWGM foi considerado o melhor método, tendo eficácia e velocidade superiores ao do CG na maioria dos problemas. O SG no caso quadrático teve o comportamento esperado, não se equiparando aos métodos específicos para quadráticas.

Um dos principais problemas enfrentados foi a escolha da linguagem de programação, que inicialmente foi o Octave (<https://www.gnu.org/software/octave/>), devido a facilidade em trabalhar com matrizes e ser uma linguagem muito intuitiva. Com o decorrer do projeto, se tornou vantajoso a mudança para o Python 3, que foi útil por uma grande parte do desenvolver dos testes numéricos para o método CG e DWGM. Mas novamente, mesmo com todas as ferramentas da linguagem se tornou difícil a leitura de problemas com formato *nl* (que nos fornece o cálculo de derivadas e hessianas automáticas). Por último, a linguagem Julia (<https://julialang.org/>), sendo uma linguagem de alto nível assim como Octave, se tornou a padrão, devido a ser dinâmica, fácil, de alto desempenho e possuir diversos pacotes voltados para análise numérica e otimização.

Este projeto teve grande contribuição na minha formação, pois incentivou a constante busca por conhecimentos na área de matemática aplicada, com o estudo dos mais variados métodos voltados para otimização irrestrita. Também foi possível aprender sobre a dinâmica de diversas linguagens de programação. Em particular, a linguagem Julia com suas incríveis ferramentas para área de otimização. Com o estudo dos artigos, melhorou-se a

compreensão do inglês e também a estrutura organizacional e argumentativa.

## Agradecimentos

---

Este trabalho teve o apoio da FAPES (processo 116/2019) e da UFES.

## Referências Bibliográficas

---

- [Andreani & Raydan, 2020] Andreani, R. & Raydan, M. (2020). Properties of the delayed weighted gradient method. *Computational Optimization and Applications*, v.78(n.1):167–180.
- [Davis & Hu, 2011] Davis, T. A. & Hu, Y. (2011). The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1–25.
- [Dolan & Moré, 2002] Dolan, E. D. & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213.
- [Gould et al., 2015] Gould, N. I., Orban, D., & Toint, P. L. (2015). CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational optimization and applications*, 60(3):545–557.
- [Gouvêa et al., 2016] Gouvêa, É. J., Regis, R. G., Soterroni, A. C., Scarabello, M. C., & Ramos, F. M. (2016). Global optimization using q-gradients. *European Journal of Operational Research*, v. 251(n. 3):727–738.
- [Grippo et al., 1986] Grippo, L., Lampariello, F., & Lucidi, S. (1986). A nonmonotone line search technique for newton’s method. *SIAM Journal on Numerical Analysis*, 23(4):707–716.
- [Hager & Zhang, 2005] Hager, W. W. & Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1):170–192.
- [Martinez & Santos, 1995] Martinez, J. M. & Santos, S. A. (1995). Métodos computacionais de otimização. *Colóquio Brasileiro de Matemática, Apostilas*, 20.
- [Nocedal & Wright, 2006] Nocedal, J. & Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- [Oviedo Leon, 2019] Oviedo Leon, H. F. (2019). A delayed weighted gradient method for strictly convex quadratic minimization. *Computational Optimization and Applications*, v.74(n.3):729–746.
- [Raydan, 1997] Raydan, M. (1997). The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, v. 7(n. 1):26–33, <https://doi.org/10.1137/S1052623494266365>.
- [Ribeiro & Karas, 2013] Ribeiro, A. & Karas, E. (2013). *Otimização contínua : aspectos teóricos e computacionais*. Cengage Learning.
- [Soterroni et al., 2011] Soterroni, A. C., Galski, R. L., & Ramos, F. M. (2011). The q-gradient vector for unconstrained continuous optimization problems. In *Operations Research Proceedings 2010*, pages 365–370. Springer.