

Métodos de otimização livres de derivadas

Vitaliano S. Amaral – UFPI
vitalianoamaral.github.io

II Workshop de Matemática da UFPI



Janeiro de 2026

Zero de uma Função

Dado uma função $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, considere o seguinte problema:

$$\text{encontrar } x^* \text{ tal que } F(x^*) = 0. \quad (1)$$

► Como resolver o Problema (1)?

1. Isolando a variável x .
2. Método de Newton

$$JF(x^k)(x^{k+1} - x^k) + F(x^k) = 0, \quad k = 0, 1, 2, \dots$$

► Qual a melhor maneira de resolver esse Problema (8)?

- Depende das propriedades da função F :

Mínimo de uma Função

Dado uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$, considere o seguinte problema:

$$\text{encontrar } x^* \text{ tal que } f(x^*) \leq f(x) \text{ para todo } x \in \mathbb{R}^n. \quad (2)$$

► Como resolver o Problema (2)?

1. Em geral, não é simples obter x^* .
2. Determinar x^* de modo que $\nabla f(x^*) = 0$ e aplicar o teste das derivadas.
3. Método de Cauchy: $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$,
4. Método de Newton: $\nabla^2 f(x^k)(x^{k+1} - x^k) + \nabla f(x^k) = 0$.

► Qual a melhor maneira de resolver esse Problema (2)?

- Depende das propriedades da função f , disponibilidade e custo computacional.

Otimização Livre de Derivadas

Em muitos problemas práticos de otimização, as derivadas das funções envolvidas:

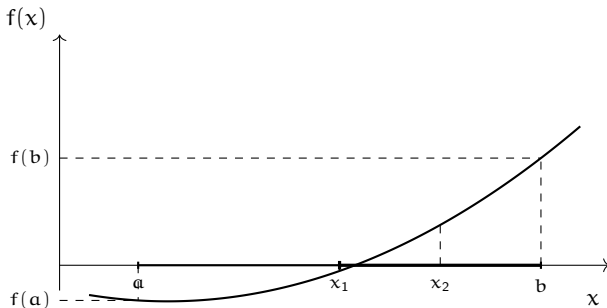
- ▶ não estão disponíveis,
- ▶ são difíceis de calcular,
- ▶ ou são muito custosas do ponto de vista computacional.

Nesse contexto, os métodos livres de derivadas tem chamado atenção, pois busca uma solução do problema utilizando apenas avaliações da função, sem exigir o cálculo explícito de gradientes ou Hessianas.

Provavelmente, os primeiros métodos livres de derivadas surgiram para encontrar zeros de uma função $f : \mathbb{R} \rightarrow \mathbb{R}$, onde podemos destacar:

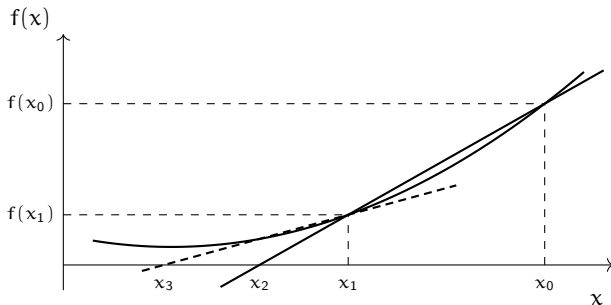
O método da bissecção

- ▶ Começamos com um intervalo $[a_0, b_0]$ tal que $f(a_0) f(b_0) < 0$.
- ▶ Definimos o ponto médio $x_1 = \frac{a_0 + b_0}{2}$.
- ▶ Se $f(a_0)f(x_1) < 0$, então $a_1 = a_0$, $b_1 = x_1$ e $[a_0, b_0] \leftarrow [a_1, b_1]$, senão $a_1 = x_1$, $b_1 = b_0$ e $[a_0, b_0] \leftarrow [a_1, b_1]$.



O método da secante

- ▶ Começamos com x_0 e x_1 com $f(x_0) \neq f(x_1)$.
- ▶ A secante por $(x_0, f(x_0))$ e $(x_1, f(x_1))$ cruza o eixo x em x_2 .
- ▶ Repetimos: a secante por $(x_1, f(x_1))$ e $(x_2, f(x_2))$ define x_3 .



O método gera uma sequência $\{x_k\}$ satisfazendo a seguinte relação:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k).$$

Métodos livres de derivadas

Nos últimos anos, o interesse por **métodos livres de derivadas** tem aumentado, motivado principalmente por:

- ▶ aplicações práticas em que derivadas não estão disponíveis ou são difíceis de calcular;
- ▶ a busca por métodos com garantias teóricas de convergência e baixo custo computacional;
- ▶ a extensão de resultados clássicos da otimização suave para cenários não suaves e outros contextos.

Métodos livres de derivadas

Recentemente, diversos trabalhos têm sido propostos na literatura sobre **métodos livres de derivadas**. +A seguir, destacamos alguns deles:



Amaral, V. (2025). - *A partially derivative-free cyclic block coordinate descent method for nonseparable composite optimization*. Mathematical Modelling and Analysis, 30(3), 535-552.
<https://doi.org/10.3846/mma.2025.23064>



V. S. Amaral, P. B. Assunção, D. R. Souza. - *A Partially Derivative-Free Proximal Method for Composite Multiobjective Optimization in the Hölder Setting*.
<https://doi.org/10.48550/arXiv.2508.20071>



G.N. Grapiglia. - *Quadratic regularization methods with finite-difference gradient approximations*. Comput. Optim. Appl., 85:683–703, 2022.



G.N. Grapiglia. - *Worst-case evaluation complexity of a derivativefree quadratic regularization method*. Optim Lett, 18:195–213, 2024.

No trabalho¹ foi considerado o seguinte problema:

$$\text{Minimizar } F(x) := f(x) + h(x) \text{ sujeito a } x \in \mathbb{R}^n, \quad (3)$$

onde f diferenciável e $h : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa.

Neste trabalho foi proposto uma versão dos métodos BCD parcialmente livre de derivadas.

Para o desenvolvimento do método proposto, foi considerado a função

$$\varphi_f : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n \text{ onde } \lim_{\lambda \rightarrow 0} \varphi_f(x, \lambda) = \nabla f(x). \quad (4)$$

¹V. S. Amaral. - *A partially derivative-free cyclic block coordinate descent method for nonseparable composite optimization*. Mathematical Modelling And Analysis, v. 30, p. 535-552, 2025.

Métodos livres de derivadas

Seja $x^0 \in \mathbb{R}^n$; para cada $i = 1, \dots, q$ tome $0 \leq B_{(i)}(x^0) \in \mathbb{R}^{n_i \times n_i}$ simétrica; tome $\alpha \in (0, 1)$, $\epsilon \in (0, 1)$, $\sigma_0 \geq 1$ e $F_{\text{target}} \in \mathbb{R}$. Inicialize $k \leftarrow 0$.

Passo 1: Escolha $\lambda_k \in \left[0, \frac{\epsilon}{\sigma_k \sqrt{n}}\right]$, e considere $\varphi_f(x^k, \lambda_k)$.

Passo 2: Ponha $x^{k,0} = x^k$ e, para cada $i = 1, \dots, q$, calcule $x^{k,i} = x^{k,i-1} + U_i s_{(i)}^k$, onde $s_{(i)}^k \in \mathbb{R}^{n_i}$ é solução de

$$\min_{s \in \mathbb{R}^{n_i}} \langle U_i^T \varphi_f(x^{k,i-1}, \lambda_k), s \rangle + \frac{1}{2} \langle B_{(i)}(x^{k,i-1}) s, s \rangle + h(x^{k,i-1} + U_i s) - h(x^{k,i-1}) + \frac{\sigma_k}{2} \|s\|^2.$$

Passo 3: Se $\left\| \sum_{i=1}^q U_i s_{(i)}^k \right\|_{\infty} < \frac{\epsilon}{\sigma_k}$ ou $F(x^{k,q}) \leq F_{\text{target}}$, **pare e declare** $x^{k,q}$ como uma solução aceitável. Caso contrário, ir ao Passo 4.

Passo 4: Se

$$F(x^{k,q}) \leq F(x^k) - \frac{\alpha}{\sigma_k} \epsilon^2,$$

defina $x^{k+1} = x^{k,q}$ e $\sigma_{k+1} = \sigma_k$, escolha $0 \leq B_{(i)}(x^{k+1}) \in \mathbb{R}^{n_i \times n_i}$ simétrica, $k \leftarrow k + 1$ e ir ao Passo 1. Caso contrário, defina $\sigma_k \leftarrow 2\sigma_k$ e ir ao Passo 1.

Boa definição

Para garantir a boa definição do método foi considerado que:

Assumption 1.

Existem $L, M \in (0, +\infty)$ e $\beta \in (0, 1]$ tais que para todo $x, y \in \mathbb{R}^n$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 + \frac{M}{\beta + 1} \|y - x\|^{\beta+1} \quad (5)$$

e

$$\|\nabla f(x) - \varphi_f(x, \lambda)\| \leq \frac{\sqrt{n}L}{2} \lambda + \frac{\sqrt{n}M}{\beta + 1} \lambda^\beta \quad (6)$$

A Suposição 1 é não restritiva.

Boa definição

Pois se $f := g_1 + g_2$, onde ∇g_1 L -Lipschitz e ∇g_2 M -Hölder com expoente β e além disso φ_f for uma das fierenças finitas:

- $\varphi_f(x, \lambda) = \left[\frac{f(x + \lambda e_1) - f(x)}{\lambda}, \dots, \frac{f(x + \lambda e_n) - f(x)}{\lambda} \right].$

- $\varphi_f(x, \lambda) = \left[\frac{f(x) - f(x - \lambda e_1)}{\lambda}, \dots, \frac{f(x) - f(x - \lambda e_n)}{\lambda} \right].$

- $\varphi_f(x, \lambda) = \left[\frac{f(x + \lambda e_1) - f(x - \lambda e_1)}{2\lambda}, \dots, \frac{f(x + \lambda e_n) - f(x - \lambda e_n)}{2\lambda} \right],$

então (5) e (6) são válidas.

Boa definição

De fato:

- ▶ $g_1(y) \leq g_1(x) + \langle \nabla g_1(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 \quad \forall x, y \in \mathbb{R}^n$
- ▶ $g_2(y) \leq g_2(x) + \langle \nabla g_2(x), y - x \rangle + \frac{M}{\beta + 1} \|y - x\|^{\beta+1} \quad \forall x, y \in \mathbb{R}^n.$

Somando as duas expressões obtemos (5).

- ▶ $\|\nabla g_1(x) - \varphi_{g_1}(x, \lambda)\| \leq \frac{\sqrt{n}L}{2} \lambda.$
- ▶ $\|\nabla g_2(x) - \varphi_{g_2}(x, \lambda)\| \leq \frac{\sqrt{n}M}{\beta + 1} \lambda^\beta.$

Somando as duas expressões e usando desigualdade trianfular obtemos (6).

Boa definição

Além das suposições anterior foi considerado que: existe $\bar{B} \geq 1$ tal que $\|B_{(i)}(x^{k,i-1})\| \leq \bar{B}$ para todos os $i \in \{1, \dots, q\}$ e k .

Assim, provamos que se

$$\|s^k\|_\infty \geq \frac{\epsilon}{\sigma_k} \text{ e } \sigma_k \geq \left[\frac{2L + q\bar{B}}{(1-\alpha)} + \frac{2q(Mn^{\frac{1-\beta}{2}} + M)}{(\beta+1)(1-\alpha)} \epsilon^{\beta-1} \right]^{\frac{1}{\beta}}$$

então

$$F(x^{k,q}) \leq F(x^k) - \alpha \frac{\sigma_k}{2} \|s^k\|_\infty^2$$

e

$$F(x^{k,q}) \leq F(x^k) - \frac{\alpha}{\sigma_k} \epsilon^2.$$

Garantindo assim a boa definição do método.

Complexidade

- ▶ Números de Iterações é limitado superiormente por:

$$\frac{c(F(x^0) - F_{\text{target}})}{\alpha} e^{-\frac{\beta+1}{\beta}}.$$

- ▶ Número de avaliações de f e seus subdiferenciais é limitado superiormente por:

$$\frac{c(F(x^0) - F_{\text{target}})}{\alpha} e^{-\frac{\beta+1}{\beta}} + \log_2 \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right).$$

com

$$\sigma_{\max} = 2\epsilon^{\frac{\beta-1}{\beta}} \max \left\{ \sigma_{\min}, \left[\frac{2L + q\bar{B}}{(1-\alpha)} + \frac{2q(Mn^{\frac{1-\beta}{2}} + M)}{(\beta+1)(1-\alpha)} \right]^{\frac{1}{\beta}} \right\}.$$

Testes Numéricos

Nos testes foi considerado o problema de mínimos quadrados penalizado por uma norma L_p :

$$\min F(x) = \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda}{p} \|\Phi(x)\|_p^p, \quad (7)$$

com $1 < p < 2$, A diferenciável e Φ linear.

Consideramos $B_{(i)} \equiv 0$, $n = 10$, $\alpha = 0,5$, $\epsilon = 5 \times 10^{-5}$, $F_{\text{target}} = 0,5$ e a função $\varphi_f : \mathbb{R}^{10} \times \mathbb{R} \rightarrow \mathbb{R}^{10}$ definida por

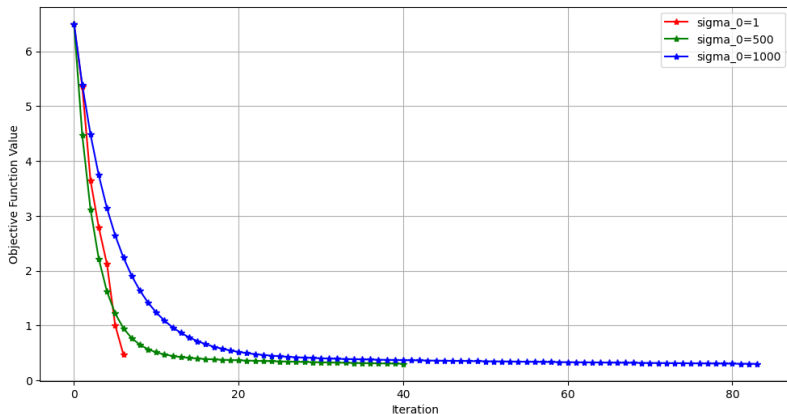
$$\varphi_f(x, \lambda) = \left[\frac{f(x + \lambda e_1) - f(x)}{\lambda}, \dots, \frac{f(x + \lambda e_{10}) - f(x)}{\lambda} \right].$$

Teste 1: $A \in \mathbb{R}^{10 \times 10}$ e $b \in \mathbb{R}^{10}$ são fixados. Utilizamos $p = 3/2$,

$f(x) = 0.5 \|Ax - b\|^2 + 5 \times 10^{-5} \sum_{i=1}^{10} |x_i|^{3/2}$, $h(x) = 0$ e $x^0 = 0 \in \mathbb{R}^{10}$.

Testes Numéricos

λ_k	N. Blocos	σ_0	N. Iterações	Approx. F^*
	10	1	10	0.4731
$\frac{\epsilon}{\sigma_k \sqrt{10}}$	10	500	40	0.3021
	10	1000	83	0.3003



Testes numéricos

Teste 2: $A \in \mathbb{R}^{10 \times 10}$ e $b \in \mathbb{R}^{10}$ são gerados aleatoriamente, Φ é a função identidade e h é nulo. Foram avaliadas 100 instâncias distintas, todas com x^0 aleatório.

Blocos	σ_0	Iterações Médias	Média-F
10	1	11.55	0.4329
1	1	17.56	0.4682

Tabela: BCDC-Dfree com A e b aleatórios.

Em seguida, σ_0 foi sorteado no intervalo $[1, 1000]$, mantendo os demais dados inalterados.

Blocos	σ_0	Iterações Médias	Média-F
10	rand()	805.45	0.4988
1	rand()	809.02	0.4988

Tabela: BCDC-Dfree com A , b e σ_0 aleatórios.

Testes numéricos

Teste 3: Consideramos o caso em que A em (7) é não linear, definido por

$$A(x) = (A_1(x), \dots, A_n(x)), \quad A_i(x) = \frac{x_i}{1 + x_{i+1}^2}, \quad i = 1, \dots, n-1,$$

com $x_{n+1} = x_1$. Tomamos $\Phi(x) = Bx$, com $B \in \mathbb{R}^{m \times n}$ aleatório e b também aleatório. Foram avaliadas 100 instâncias, todas com x^0 aleatório e σ_0 sorteado em $[1, 100]$.

Blocos	σ_0	Iterações Médias	Média-F
10	rand()	648.96	0.5008
1	rand()	664.61	0.5008

Tabela: BCDC-Dfree para o caso com A não linear.

Métodos livres de derivadas em otimização multiobjetivo

Em nosso trabalho², estudamos o seguinte problema de otimização:

$$\min_{x \in \mathbb{R}^n} G(x), \quad (8)$$

onde $G : \mathbb{R}^n \rightarrow \mathbb{R}^m$ é uma função vetorial cujos componentes são dados por

$$G_j = f_j + h_j, \quad j \in \mathcal{N} = \{1, \dots, m\}. \quad (9)$$

Assumimos que

- ▶ $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função diferenciável;
- ▶ $h_j : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ é uma função convexa.

²V. S. Amaral, P. B. Assunção, D. R. Souza. *A Partially Derivative-Free Proximal Method for Composite Multiobjective Optimization in the Hölder Setting* (submetido para publicação e disponível no arXiv)

Métodos livres de derivadas em otimização multiobjetivo

- ▶ x^* é um ponto Pareto ótimo para o Problema (1) se não existe x tal que

$$G(x) \leq G(x^*) \quad \text{e} \quad G(x) \neq G(x^*),$$

onde a desigualdade \leq entre vetores deve ser entendida no sentido componente a componente.

Isto é, não é possível melhorar um objetivo sem piorar o outro.

- ▶ x^* é um ponto Pareto crítico se existe $\gamma \in \mathbb{R}^m$ com

$\gamma \geq 0$ e $\sum_{j=1}^m \gamma_j = 1$, tal que

$$0 \in \sum_{j=1}^m \gamma_j (\nabla f_j(x^*) + \partial h_j(x^*)). \quad (10)$$

Métodos livres de derivadas em otimização multiobjetivo

Propomos um método parcialmente livre de derivadas para resolver o Problema (8).

Dado x^k , o método encontra um novo ponto x^{k+1} tal que

$$G(x^{k+1}) < G(x^k).$$

Para isso, minimizamos o seguinte modelo:

$$\Phi_{x^k}(x) + \frac{\sigma_k}{2} \|x - x^k\|^2, \quad (11)$$

onde Φ_{x^k} é definida pela seguinte expressão

$$\Phi_{x^k}(x) := \max_{j \in \mathcal{J}} \left[\langle g_{f_j}(x^k, \lambda_k) + \frac{1}{2} B_j^k (x - x^k), x - x^k \rangle + h_j(x) - h_j(x^k) \right],$$

em que B_j^k é uma matriz simétrica semidefinida positiva e $g_{f_j} : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n$ é uma aproximação de ∇f_j .

Partially Derivative-Free Proximal Method - PDFPM

Algoritmo: Seja $x^0 \in \mathbb{R}^n$, $\alpha, \epsilon \in (0, 1)$, $\sigma_0 \geq 1$, e, para todo $j \in \mathcal{N}$, $B_j^0 \in \mathbb{R}^{n \times n}$ simétrica semidefinida positiva. Inicialize $k \leftarrow 0$.

Passo 1. Para

$$0 < \lambda_k \leq \frac{\epsilon}{\sigma_k \sqrt{n}}$$

e para cada $j \in \mathcal{N}$, calcule $g_{f_j}(x^k, \lambda_k)$.

Passo 2. Encontre $\bar{x}^k \in \mathbb{R}^n$ solução do seguinte subproblema:

$$\min_{x \in \mathbb{R}^n} \max_{j \in \mathcal{J}} \left[\langle g_{f_j}(x^k, \lambda_k) + \frac{1}{2} B_j^k (x - x^k), x - x^k \rangle + h_j(x) - h_j(x^k) \right] + \frac{\sigma_k}{2} \|x - x^k\|^2.$$

Passo 3. Se $\sigma_k \|\bar{x}^k - x^k\| < \epsilon$, pare. Caso contrário, vá para o **Passo 4**.

Passo 4. Se

$$G_j(\bar{x}^k) \leq G_j(x^k) - \frac{\alpha \epsilon^2}{2 \sigma_k}, \quad \forall j \in \mathcal{N}, \quad (12)$$

for satisfeita, defina $x^{k+1} := \bar{x}^k$, $\sigma_{k+1} := \sigma_k$ e, para cada j , escolha $B_j^{k+1} \in \mathbb{R}^{n \times n}$; faça $k \leftarrow k + 1$ e retorne ao **Passo 1**. Caso contrário, defina $\sigma_k \leftarrow 2\sigma_k$ e retorne ao **Passo 1**.

Boa definição

Consideramos as seguintes suposições.

Suposição 1. Existe um número real não negativo \bar{B} tal que

$$\|B_j^k\| \leq \bar{B}, \quad \forall k \text{ e } j \in \mathcal{N}.$$

Suposição 2. Para cada $j \in \mathcal{N}$, existem números reais positivos L_j, M_j e $\beta_j \in (0, 1]$ tais que as seguintes condições são satisfeitas.

- ▶ $f_j(y) \leq f_j(x) + \langle \nabla f_j(x), y - x \rangle + \frac{L_j}{2} \|y - x\|^2 + \frac{M_j}{\beta_j + 1} \|y - x\|^{\beta_j + 1}.$
- ▶ $\|\nabla f_j(x) - g_{f_j}(x, \lambda)\| \leq \lambda \frac{\sqrt{n} L_j}{2} + \sqrt{n} \frac{M_j}{\beta_j + 1} \lambda^{\beta_j}.$

A Suposição 2 não é restritiva. Ela é válida quando algumas funções f_j possuem gradientes Lipschitz contínuos, outras possuem gradientes de Hölder, e g_{f_j} é obtida por diferenças finitas (diferenças progressivas, regressivas e centrais).

Boa definição

Mais precisamente, temos o seguinte:

- ▶ Se ∇f_j é Lipschitz contínuo, então as seguintes condições são satisfeitas:

$$f_j(y) \leq f_j(x) + \langle \nabla f_j(x), y - x \rangle + \frac{L_j}{2} \|y - x\|^2,$$

$$\|\nabla f_j(x) - g_{f_j}(x, \lambda)\| \leq \lambda \frac{\sqrt{n} L_j}{2}.$$

- ▶ Se ∇f_j é Hölder contínuo, então as seguintes condições são satisfeitas:

$$f_j(y) \leq f_j(x) + \langle \nabla f_j(x), y - x \rangle + \frac{M_j}{\beta_j + 1} \|y - x\|^{\beta_j + 1},$$

$$\|\nabla f_j(x) - g_{f_j}(x, \lambda)\| \leq \sqrt{n} \frac{M_j}{\beta_j + 1} \lambda^{\beta_j}.$$

Isso implica que, neste caso, a Suposição 2 é satisfeita.

Boa definição

O resultado a seguir mostra que o Passo 4 do algoritmo está bem definido.

Teorema: Seja \bar{x}^k o ponto obtido no Passo 2 do algoritmo PDFPM. Suponha que as Suposições 1 e 2 sejam satisfeitas. Se

$$\sigma_k \|\bar{x}^k - x^k\| \geq \epsilon \quad (13)$$

e

$$\sigma_k \geq \max_{j \in \mathcal{J}} \left[\frac{5L_j + \bar{B}}{1 - \alpha} + \frac{2n^{\frac{1-\beta_j}{2}} M_j + 2M_j}{(\beta_j + 1)(1 - \alpha)} \epsilon^{\beta_j - 1} \right]^{\frac{1}{\beta_j}}, \quad (14)$$

então

$$G_j(\bar{x}^k) \leq G_j(x^k) - \frac{\alpha \epsilon^2}{2\sigma_k}, \quad \forall j \in \mathcal{N}. \quad (15)$$

Em outras palavras, a redução em G exigida no Passo 4 é garantida para todo σ_k suficientemente grande.

Complexidade

Dado um alvo $\bar{G} < G(x^0)$, queremos estimar o número máximo de iterações (ou de avaliações da função) para as quais as seguintes condições são simultaneamente satisfeitas:

$$G(x^{k+1}) > \bar{G} \quad \text{e} \quad \sigma_k \|x^{k+1} - x^k\| \geq \epsilon. \quad (16)$$

O número de avaliações de G depende de quantas vezes σ_k é aumentado no Passo 4.

Assumindo que as Suposições 1 e 2 sejam satisfeitas, provamos que:

- ▶ O número máximo de iterações que satisfazem (16) é

$$\max_{j \in \mathcal{J}} \left\{ G_j(x^0) - \bar{G}_j \right\} \frac{2c}{\alpha} \epsilon^{-\frac{\beta+1}{\beta}}, \quad (17)$$

$$\text{onde } c = 2 \max_{j \in \mathcal{J}} \left[\frac{2L_j + \bar{B}}{1 - \alpha} + \frac{2n^{\frac{1-\beta_j}{2}} M_j + 2M_j}{(\beta_j + 1)(1 - \alpha)} \right]^{\frac{1}{\beta_j}}.$$

- ▶ O número máximo de avaliações que satisfazem (16) é

$$\max_{j \in \mathcal{J}} \{G_j(x^0) - \overline{G}_j\} \frac{2c}{\alpha} \epsilon^{-\frac{\beta+1}{\beta}} + \log_2 \left(\frac{\sigma_{\max}}{\sigma_0} \right), \quad (18)$$

onde $\sigma_{\max} = \epsilon^{\frac{\beta-1}{\beta}} c$.

- ▶ Coincidindo com a complexidade do método proposto por Calderón et al.³ para $p = 1$.
- ▶ Para $m = 1$, recupera-se a complexidade clássica dos modelos com regularização quadrática sob condições de Hölder^{4,5}.

³Calderón, L., Diniz-Ehrhardt, M.A., Martínez, J.M.: On high-order model regularization for multiobjective optimization. Optimization Methods and Software 37(1), 175–191 (2022)

⁴Martínez, J.M.: On high-order model regularization for constrained optimization. SIAM Journal on Optimization 27(4), 2447–2458 (2017)

⁵Yashtini, M.: On the global convergence rate of the gradient descent method for functions with gradients Hölder contínuos. Optimization Letters 10(6), 1361–1370 (2016)

Aplicação à Otimização Multiobjetivo Robusta: Considere

$F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ onde

$$F_j(x) := f_j(x) + \langle x, z \rangle,$$

$f_j(x)$ diferenciável e $z \in \mathcal{Z}_j \subset \mathbb{R}^n$ é o parâmetro de incerteza.

Minimizar $F(x)$ no pior cenário possível significa resolver o seguinte problema

$$\min_{x \in \mathbb{R}^n} G(x) \tag{19}$$

onde cada componente de $G : \mathbb{R}^n \rightarrow \mathbb{R}^m$ é dada por $f_j(x) + \max_{z \in \mathcal{Z}_j} \langle x, z \rangle$.

Consideramos $h_j(x) := \max_{z \in \mathcal{Z}_j} \langle x, z \rangle$ e o conjunto de incerteza \mathcal{Z}_j dado por

$$\mathcal{Z}_j = \{z \in \mathbb{R}^n \mid A_j z \leq \delta e\}$$

onde $e = (1, \dots, 1)^T \in \mathbb{R}^n$, $\delta \geq 0$, $A_j := [\tilde{A}_j; -\tilde{A}_j] \in \mathbb{R}^{2n \times n}$, com $\tilde{A}_j \in \mathbb{R}^{n \times n}$ uma matriz não singular.

Testes numéricos

Para ilustrar a aplicabilidade do algoritmo proposto, consideramos os seguintes exemplos.

- **AAS1:** $f_1(x) = \frac{1}{2} \|Ax - b\|_2^2$ e $f_2(x) = \frac{\mu}{p} \|Dx\|_p^p$ onde $\mu = 0.9$,

$$A = \begin{bmatrix} 2.0 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}, b = \begin{bmatrix} 1.0 \\ -0.5 \end{bmatrix}, D = \begin{bmatrix} 1.0 & 0.8 \\ 0.3 & 1.2 \end{bmatrix}, p = 1.003.$$

Neste caso, f_1 tem gradiente Lipschitz e f_2 gradiente Hölder.

- **AAS2:** $f_1(x) = \frac{\mu_1}{p_1} \|D_1(x - c_1)\|_{p_1}^{p_1}$ e $f_2(x) = \frac{\mu_2}{p_2} \|D_2(x - c_2)\|_{p_2}^{p_2}$,

$$\text{com } \mu_1 = 1.2, \mu_2 = 0.8, \quad p_1 = 1.003, \quad p_2 = 1.002, \quad c_2 = \begin{bmatrix} -1.2 \\ 0.8 \end{bmatrix}.$$

$$D_1 = \begin{bmatrix} 1.2 & -0.3 \\ 0.4 & 1.5 \end{bmatrix}, D_2 = \begin{bmatrix} 1.8 & 0.5 \\ -0.2 & 1.1 \end{bmatrix} \text{ e } c_1 = \begin{bmatrix} 1.5 \\ -1.0 \end{bmatrix}.$$

Neste caso, f_1 e f_2 possuem gradientes Hölder contínuos.

Testes numéricos

Os experimentos foram realizados para diferentes níveis de incerteza

$$\delta \in \{0.0, 0.02, 0.05, 0.1\}.$$

Para cada valor fixo de δ , foram geradas m matrizes $A_j \in \mathbb{R}^{2n \times n}$, com $j \in \mathcal{N}$.

Cada instância do problema foi resolvida 200 vezes, com pontos iniciais aleatórios:

$$x^0 \in [-2, 2] \times [-2, 2] \text{ (AAS1)}, \quad x^0 \in [-5, 5] \times [-5, 5] \text{ (AAS2)}.$$

Frentes de Pareto: AAS1

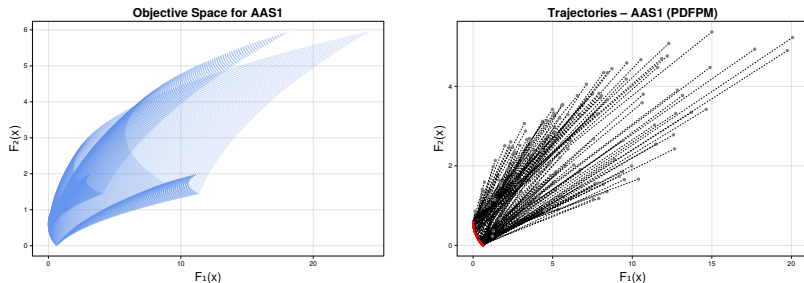


Figura: Espaço objetivo e fronteira de Pareto aproximada para AAS1 ($\delta = 0$).

- ▶ **Direita:** trajetória do algoritmo e aproximação da fronteira de Pareto (ponto inicial em cinza e final em vermelho).
- ▶ **Esquerda:** visão global do espaço objetivo.

Frentes de Pareto: AAS2

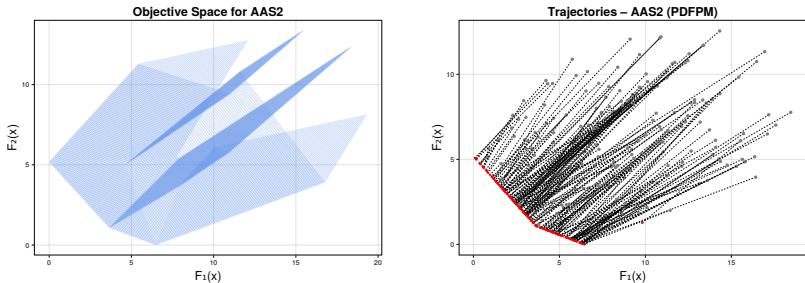
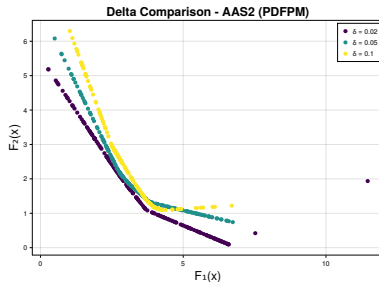
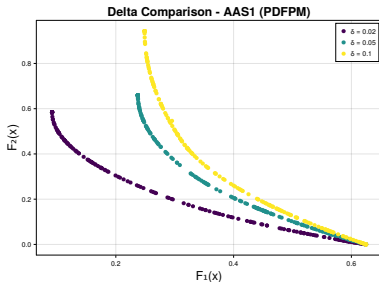


Figura: Espaço objetivo e fronteira de Pareto aproximada para AAS2 ($\delta = 0$).

- ▶ **Direita:** trajetória do algoritmo e aproximação da fronteira de Pareto (ponto inicial em cinza e final em vermelho).
- ▶ **Esquerda:** visão global do espaço objetivo.

Efeito da Incerteza

A figura a seguir compara as reconstruções para diferentes valores de δ .



- ▶ **Esquerda:** fronteiras de Pareto aproximadas para o problema AAS1.
- ▶ **Direita:** fronteiras de Pareto aproximadas para o problema AAS2.
- ▶ À medida que o nível de incerteza aumenta, a fronteira de Pareto obtida pelo PDFPM se afasta da fronteira esperada.

Testes numéricos

Para avaliar a confiabilidade do PDFPM no contexto de Hölder, realizamos 200 testes independentes para cada problema e para cada valor de δ .

Contabilizamos quantas execuções convergiram para um ponto que satisfaz o critério de parada.

Problema	$\delta = 0.0$	$\delta = 0.02$	$\delta = 0.05$	$\delta = 0.1$
AAS1	196 (98%)	199 (99.5%)	199 (99.5%)	195 (97.5%)
AAS2	199 (99.5%)	200 (100%)	200 (100%)	200 (100%)

- ▶ **AAS1:** entre 195 e 199 execuções bem-sucedidas, com taxas de sucesso variando de 97.5% a 99.5%.
- ▶ **AAS2:** entre 199 e 200 execuções bem-sucedidas para todos os valores de δ , com taxas de sucesso entre 99.5% e 100%.

Esses resultados mostram que o PDFPM é estável e confiável sob diferentes níveis de incerteza, mesmo com condições de Hölder.

Comparação com ProxGrad e CondG

Comparamos PDFPM com ProxGrad⁶ e CondG⁷.

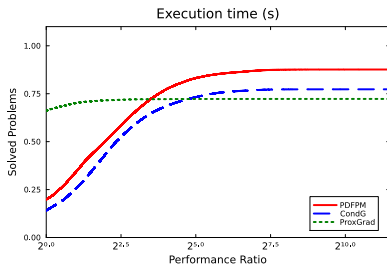
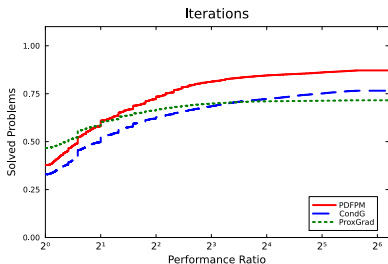
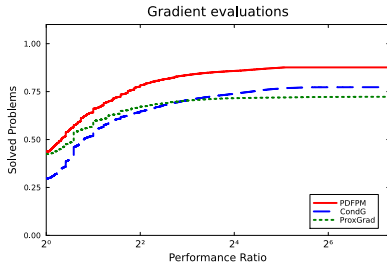
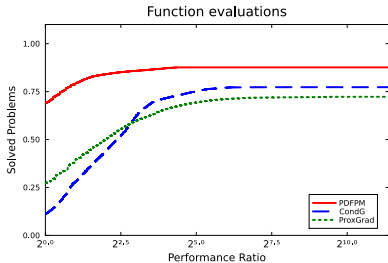
- ▶ Foram analisados diferentes níveis de incerteza:

$$\delta \in \{0.0, 0.02, 0.05, 0.1\}.$$

- ▶ Para cada instância, cada algoritmo foi executado 200 vezes, usando o mesmo conjunto de pontos iniciais, garantindo uma comparação justa.
- ▶ Consideramos 56,000 instâncias (70 problemas, 4 níveis de incerteza e 200 pontos iniciais).

⁶Tanabe, H., Fukuda, E.H., Yamashita, N.: Proximal gradient methods for multi-objective optimization and their applications. Computational Optimization and Applications 72(2), 339–361 (2019)

⁷Assunção, P.B., Ferreira, O.P., Prudente, L.F.: A generalized conditional gradient method for multiobjective composite optimization problems. Optimization 74(2), 473–503 (2025)



O PDFPM apresenta maior eficiência e robustez em termos de avaliações de função e gradiente; o ProxGrad é geralmente mais rápido em iterações e tempo, mas o PDFPM resolve a maior fração total de instâncias.

Organização e Comunidade

- ▶ Organização **VectorOptimizerGroup** no GitHub.
- ▶ Inspiração: JuliaSmoothOptimizers e JuMP.
- ▶ Objetivo: consolidar pacotes open source em Julia para **otimização multiobjetivo direta**.
- ▶ Espaço para colaboração, melhorias, novas ferramentas e integração.



Contribuições e integrações são bem-vindas.

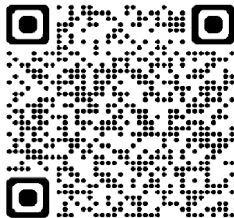
OBRIGADO!

`vitalianoamaral@ufpi.edu.br`

`https://vitalianoamaral.github.io`



`vitalianoamaral.github.io`



Organização (GitHub)

