UNIVERSIDADE FEDERAL DE JUIZ DE FORA
DEPARTAMENTO DE MATEMÁTICA
MESTRADO EM MATEMÁTICA

Gabriel de Oliveira Machado

Algoritmos para geração da frente de Pareto da regressão Lasso

Juiz de Fora
2023

Gabriel de Oliveira Machado

Algoritmos para geração da frente de Pareto da regressão Lasso

Dissertação apresentada ao Departamento de Matemática da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de mestre em matemática.

Orientador: Prof. Dr. Wilhelm Passarella Freire

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Machado, Gabriel de Oliveira.
Algoritmos para geração da frente de Pareto da regressão Lasso / Gabriel de Oliveira Machado. – 2023.
65 f. : il.

Orientador: Wilhelm Passarella Freire
Dissertação – Universidade Federal de Juiz de Fora, Departamento de Matemática. Mestrado em Matemática, 2023.

1. Otimização Multiobjetivo. 2. Regressão Lasso . 3. Otimização. I. Freire, Wilhelm Passarella, orient. II. Título.

Gabriel de Oliveira Machado

Algoritmos para geração da frente de Pareto da regressão Lasso

Dissertação apresentada ao Programa de Pós-graduação em Matemática da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Matemática. Área de concentração: Matemática Aplicada

Aprovada em 16 de março de 2023.

BANCA EXAMINADORA

Prof. Dr. Wilhelm Passarella Freire - Orientador

Universidade Federal de Juiz de Fora

Prof. Dr. Sandro Rodrigues Mazorche

Universidade Federal de Juiz de Fora

Prof. Dr. Hernando José Rocha Franco

IF Sudeste/ MG

Juiz de Fora, 17/03/2023.

Documento assinado eletronicamente por **Wilhelm Passarella Freire**, **Professor(a)**, em 20/03/2023, às 14:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Sandro Rodrigues Mazorche**, **Professor(a)**, em 24/03/2023, às 08:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **HERNANDO JOSE ROCHA FRANCO**, **Usuário Externo**, em 28/03/2023, às 13:24, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1191322** e o código CRC **D21739EC**.

Aos meus pais e à minha namorada,
sem os quais não teria a determinação para prosseguir.

AGRADECIMENTOS

Aos meus pais, Eduardo e Beatriz, que por meio de seus incansáveis esforços, permitiram que eu tivesse acesso à educação de qualidade. À minha namorada, Ana Carolina, que esteve presente desde o início dessa fase, por prover todo o suporte necessário.

Ao meu orientador, professor Wilhelm Passarella Freire, pela dedicação e esforços empregados a essa orientação e no ensino da matemática aplicada de forma didática e eficiente. A todos os demais professores que fizeram parte da minha trajetória, e que certamente contribuíram para minha formação.

À CAPES por fomentar o desenvolvimento da pesquisa e pelo apoio financeiro. À Universidade Federal de Juiz de Fora, por fornecer ensino de qualidade e de forma gratuita, ao qual eu dificilmente teria acesso de outra forma.

"Don't be afraid
What your mind conceives
You should make a stand
Stand up for what you believe." (Muse - Invincible)

RESUMO

Problemas de modelagem podem envolver um número muito elevado de variáveis de entrada, principalmente quando estamos interessados em estudar dados experimentais e obter um modelo explicativo para um certo fenômeno ou evento a partir destes. Em geral, deseja-se que o modelo seja interpretável e que seja possível obter uma conclusão clara sobre a relação de cada variável explicativa com a resposta, onde um número muito grande de variáveis pode dificultar tal interpretação. A utilização da regressão Lasso é uma opção viável para obter modelos com um menor número de variáveis de entrada, enquanto mantendo a precisão obtida pelos mesmos. No entanto, a geração de modelos a partir do Lasso exige maior esforço computacional quando comparado a outros métodos, e por esse motivo é importante que o processo de geração destes modelos seja eficiente. Nesse estudo, realizamos a análise de diferentes algoritmos para a geração de modelos a partir do Lasso, bem como formas de reduzir o esforço computacional quando desejamos obter diversos modelos, para diferentes valores do parâmetro de regularização, para um dado problema, por meio da aproximação da frente de Pareto do Lasso.

Palavras-chave: Otimização Multiobjetivo. Regressão Lasso. Otimização.

ABSTRACT

Modeling problems can involve a very large number of input variables, especially when we are interested in studying experimental data and obtaining an explanatory model for a certain phenomenon or event from them. In general, it is desired that the model be interpretable and that a clear conclusion can be obtained about the relationship of each input variable to the response, where a large number of variables can make such interpretation difficult. The use of Lasso regression is a viable option for obtaining models with a smaller number of input variables, while maintaining the accuracy obtained by them. However, generating models from Lasso requires greater computational effort compared to other methods, and for this reason, it is important that the process of generating these models is efficient. In this study, we conducted an analysis of different algorithms for generating models from Lasso, as well as ways to reduce computational effort when we want to obtain multiple models for different regularization parameter values for a given problem, by approximating the Pareto front of Lasso.

Keywords: Multiobjective Optimization. Lasso Regression. Optimization.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

COBYLA	Constrained Optimization by Linear Approximation
FFT	Fast Fourier Transform
MQO	Mínimos Quadráticos Ordinários
OM	Otimização Multiobjetivo
POM	Problema de Otimização Multiobjetivo
PQS	Programação Quadrática Sequencial
RC	Região de Confiança
SLSQP	Sequential Least Squares Programming

SUMÁRIO

1 INTRODUÇÃO

Modelos construídos a partir de regressão são comuns em diferentes áreas da ciência e da engenharia, principalmente quando é conhecida a existência de uma relação linear entre as variáveis de entrada, ou alguma função destas, e a variável de resposta que se deseja estudar. Sendo uma abordagem comum para problemas, existem vários métodos para criar modelos de regressão, cada um com propriedades estatísticas diferentes, e que podem se mostrar adequados em diferentes cenários.

Em algumas aplicações, o número de variáveis de entrada (ou *preditores*) é alto, e o processo de geração de um modelo de regressão pode se tornar custoso computacionalmente, isto é, necessitando de um alto número de operações computacionais para a obtenção do modelo. A complexidade computacional de gerar um modelo de regressão depende, além do método utilizado, da quantidade de preditores e de amostras disponíveis no conjunto de dados utilizado para ajustar o modelo.

A regressão Lasso (*Least Absolute Shrinkage and Selection Operator*) é um dos métodos disponíveis e é conhecido por ser um método de regularização, ou seja, impõe uma restrição sobre a norma do vetor de parâmetros da regressão, capaz de reduzir o número de variáveis de entrada no modelo, pois elimina variáveis cujos parâmetros tenham valor próximo a 0. Esse comportamento faz com que por vezes os modelos gerados pelo Lasso possuam um número significativamente menor de variáveis de entrada e, dessa forma, sejam mais fáceis de interpretar e extrair informações.

Assim como outros métodos de regularização, o Lasso depende de um parâmetro de regularização para gerar um modelo para um conjunto de dados específico, e diferentes modelos podem ser gerados para o mesmo conjunto de dados ao se variar tal parâmetro. A escolha do modelo, dentre todos os possíveis por meio da variação do parâmetro de regularização, deve ser feita pelo tomador de decisão, que poderá levar em conta características específicas do problema ao qual se deseja solucionar ou do cenário estudado. Existem também métodos estatísticos como a validação cruzada, que permite determinar objetivamente o modelo que possui melhor performance no caso de classificadores, por exemplo.

Embora o Lasso possua propriedades interessantes, a geração de diferentes modelos conforme o parâmetro de regularização escolhido é mais custosa em relação a outros métodos por utilizar a norma da soma ($\|\cdot\|_1$) e, portanto, tornar a função de minimização não-diferenciável. Por esta razão, não podemos obter uma expressão para o vetor de parâmetros com base nos dados de entrada, e devem ser utilizados métodos de otimização não diferenciável para obter os modelos, que em geral são mais custosos computacionalmente que os métodos para otimização de funções diferenciáveis, como é o caso da função de minimização do Ridge, que utiliza a norma euclidiana ($\|\cdot\|_2$).

Por fim, geralmente desejamos gerar diversos modelos para o mesmo conjunto de dados, sendo cada um destes para um valor diferente do parâmetro de regularização, que denominaremos t neste trabalho, de forma que o tomador de decisão possa escolher quais são os valores de t que melhor se adéquam ao problema. Nesse sentido, o cenário ideal é aquele onde disponibilizamos todos os modelos possíveis (para todos os valores possíveis de t). Como discutiremos mais adiante, isso é possível se descrevermos uma fórmula paramétrica $\mathbf{x}^*(t)$ para o vetor de parâmetros dependendo apenas do valor de t.

Como veremos, o conjunto de todas as soluções $\mathbf{x}^*(t)$ para $t > 0$ representa a frente de Pareto de um problema de otimização bi-objetivo associado à regressão Lasso para um conjunto de dados específico. Diversos métodos para otimização multiobjetivo podem ser aplicados na solução deste problema, como apresentado em [15]. Realizaremos uma análise mais aprofundada de tais métodos no Capítulo 3.

Normalmente, o conjunto de soluções para a regressão Lasso é apresentado em um gráfico 2D onde os eixos representam as quantidades que estamos interessados em minimizar, ou seja, o *quadrado do erro* e a *norma da soma do vetor de parâmetros*. Para simplificar, a partir de agora nos referimos ao vetor de parâmetros como $\mathbf{x}$. A Figura 1 exibe um representação do conjunto de soluções para um conjunto de dados específico.

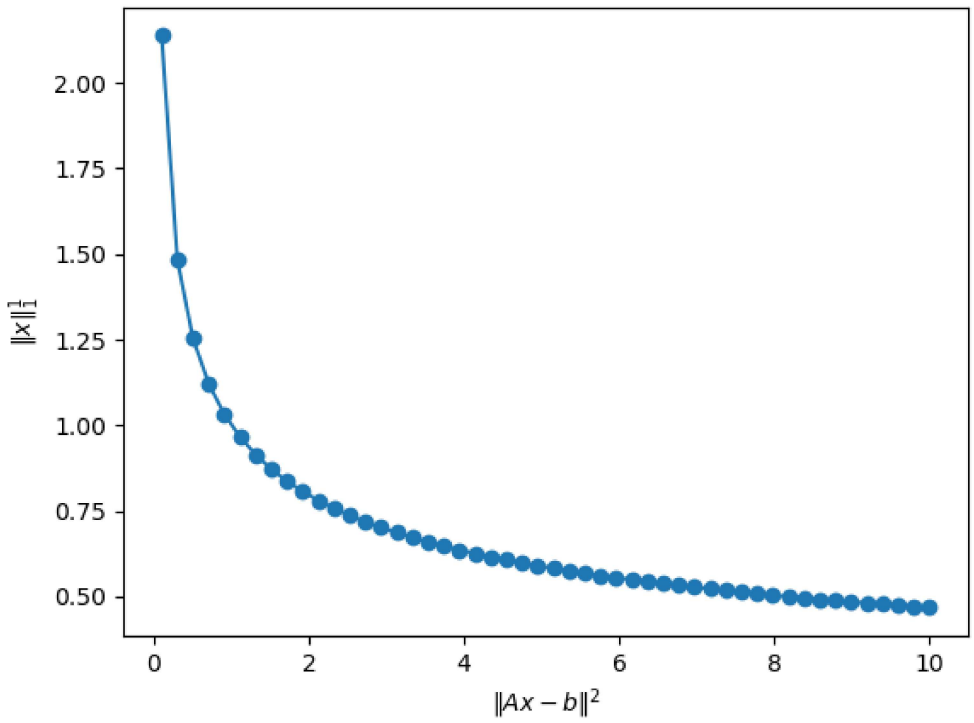

Figura 1 – Exemplo de frente de Pareto.

Esta representação é muito útil, visto que as soluções do problema bi-objetivo que representa o Lasso com n preditores está no espaço $\mathbb{R}^n$ e, para $n > 3$, a visualização das

soluções se torna impossível. Além disso, esse tipo de gráfico nos permite visualizar os atributos mais importantes de nossas soluções e pode auxiliar o *tomador de decisão* no processo de seleção de um modelo.

O *tomador de decisão* refere-se ao indivíduo ou algoritmo responsável por escolher o melhor modelo entre os modelos disponíveis, dado pela regressão Lasso. Existe uma série de métodos estatísticos para selecionar o melhor modelo e até mesmo métricas para medir a qualidade de um determinado modelo. Esses assuntos não serão aprofundados nesse texto, mas serão apresentados nas seções seguintes.

No decorrer deste texto, apresentaremos métodos computacionais para gerar modelos com regressão Lasso e suas propriedades estatísticas. Além disso, propomos um algoritmo para obter uma aproximação da frente de Pareto do problema Lasso com um número reduzido de minimizações para diferentes valores do parâmetro de regularização t. Também investigamos as perdas que ocorrem quando usamos a interpolação para aproximar a fórmula paramétrica de $\mathbf{x}^*(t)$ por meio do algoritmo descrito, ao utilizarmos diferentes métodos de interpolação.

O trabalho está organizado como segue: O Capítulo 2 apresenta os principais conceitos associados à regressão Lasso e suas características. O Capítulo 3 apresenta os conceitos relacionados à otimização multiobjetivo, incluindo a definição da frente de Pareto e a caracterização do problema da regressão Lasso. O Capítulo 4 apresenta os algoritmos utilizados para a obtenção dos modelos com valor do parâmetro de regularização específico, bem como o algoritmo proposto para gerar a frente de Pareto do problema. O Capítulo 5 apresenta os resultados dos experimentos computacionais realizados. Finalmente, o Capítulo 6 apresenta as conclusões obtidas a partir das análises dos resultados.

2 CONCEITOS GERAIS

A regressão Lasso é uma variação da regressão linear com abordagem de Mínimos Quadrados Ordinários (MQO) apresentada pela primeira vez em [17]. Sua principal diferença do MQO é a restrição adicionada ao problema sobre a norma da soma da variável no problema de otimização.

Conforme apresentado em [17], o Lasso é um método de regularização, os quais são métodos que tentam alcançar um melhor resultado na regressão trocando variância por viés, ou seja, diminuindo a variância do modelo mas introduzindo algum nível de viés. Em alguns casos, a regularização pode levar a uma melhor precisão geral, como Tibshirani descreveu em [8]. Deve-se notar que o método de regressão linear com menor variância entre todas as regressões sem viés é o MQO (Teorema de Gauss-Markov, ver [8] Seção 3.2.2), mas a ausência de viés nem sempre é a melhor escolha em um determinado problema.

Existem outros exemplos de regressão, semelhantes ao Lasso, que utilizam regularização, como a Ridge, cuja restrição se impõe ao quadrado da norma euclidiana dos coeficientes. Há também o Elastic Net, que surge como resultado de uma modificação na norma da restrição, como será descrito formalmente adiante.

A regressão Ridge, em especial, tem uma fórmula exata para o vetor de coeficientes, para cada valor do parâmetro de regularização t fixado. Isso torna o processo de obtenção da frente de Pareto mais fácil do que no caso do Lasso, para o qual não existe uma expressão analítica em função do parâmetro de regularização fixado. Estes fatos são explorados e explicados adiante.

2.1 DEFINIÇÕES

Começaremos apresentando os conceitos básicos e as definições usadas na formulação do problema de regressão. Uma vez que o Lasso está associado ao MQO, sendo uma variação deste método, o definimos em primeiro lugar.

Considere $\mathbf{A}$ uma matriz $m \times n$ representando os dados para as variáveis de entrada e $\mathbf{b}$ um vetor coluna de dimensão m representando as respectivas respostas. Ao realizar uma regressão linear, gostaríamos de determinar um vetor coluna $\mathbf{x} = (x_1, x_2, \dots x_n)$, de dimensão n, também denominado vetor de parâmetros, tal que, dado um vetor linha $\mathbf{a}_i = (a_{i1}, a_{i2}, \dots a_{in})$ da matriz $\mathbf{A}$,

$$y(\mathbf{a}_i) = x_1 a_{i1} + x_2 a_{i2} + \cdots + x_n a_{in} \approx b_i \tag{2.1}$$

é uma boa aproximação para a resposta b_i relacionada a esta entrada, no sentido de reduzir o erro cometido pelo modelo. Obviamente, se os dados não forem descritos por uma

relação linear dos preditores, esse erro poderá ser considerável, tornando a aproximação ineficiente.

Portanto, ao realizar uma regressão linear usando MQO visamos reduzir o erro quadrado residual, que é dado por

$$SE = \sum_{i=1}^{m} (\mathbf{a}_i \cdot \mathbf{x} - b_i)^2 = \|\mathbf{Ax} - \mathbf{b}\|_2^2 \tag{2.2}$$

Isso implica que o problema de otimização relacionado ao MQO é

$$\min \|\mathbf{Ax} - \mathbf{b}\|_2^2 \quad , \quad \mathbf{x} \in \mathbb{R}^n, \tag{2.3}$$

que se trata de um problema de otimização irrestrito. As condições de otimalidade para este problema requerem que

$$\nabla \|\mathbf{Ax} - \mathbf{b}\|_2^2 = 2(\mathbf{Ax} - \mathbf{b}) = 0 \tag{2.4}$$

A partir desta condição, podemos chegar à fórmula bem conhecida que nos dá o vetor de coeficientes segundo o MQO,

$$\mathbf{x}_{ls} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{2.5}$$

Os métodos de regularização ficam definidos ao se adicionar uma restrição ao problema MQO, como por exemplo

$$\begin{aligned} &\min \|\mathbf{Ax} - \mathbf{b}\|_2^2 \quad , \quad \mathbf{x} \in \mathbb{R}^n \\ &\text{sujeito a } \|x\|_p^p < t, \end{aligned} \tag{2.6}$$

para algum valor de $p > 0$, que define o tipo de método de encolhimento. Por exemplo, $p = 2$ define o problema Ridge e $p = 1$ define o problema Lasso. O parâmetro de regularização $t \geq 0$ define o tamanho do encolhimento, e valores diferentes de t podem resultar em soluções diferentes para um determinado problema de regressão. É fácil ver que, se $t \geq \|x_{ls}\|_p^p$, o problema regularizado se reduz ao MQO. Dito isso, sempre consideraremos $0 \leq t < \|x_{ls}\|_p^p$.

O problema relacionado à regressão Ridge, citada anteriormente, pode ser definido como

$$\begin{aligned} &\min \|\mathbf{Ax} - \mathbf{b}\|_2^2, \mathbf{x} \in \mathbb{R}^n \\ &\text{sujeito a } \|x\|_2^2 < t. \end{aligned} \tag{2.7}$$

Este problema pode ser definido como um problema de otimização irrestrita, adicionando-se um fator de penalidade à função objetivo, como segue

$$\min \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2, \mathbf{x} \in \mathbb{R}^n. \tag{2.8}$$

Aqui, o parâmetro $\lambda > 0$ é único para cada parâmetro t na formulação anterior. A existência dessa relação um-para-um é provada em [17], mas nem sempre é possível determinar explicitamente essa relação, isto é, não há uma expressão geral que forneça λ em função de t ou vice-versa.

Assim como no problema MQO, podemos derivar a expressão (2.8) de forma a chegar em uma expressão para a solução, ou seja, dadas as matrizes $\mathbf{A}$, $\mathbf{b}$ e o parâmetro λ, o vetor de coeficientes fica definido por

$$\mathbf{x}_{\text{ridge}}(\lambda) = (A^T A + \lambda I)^{-1} A^T b \tag{2.9}$$

No caso do Lasso, que é definido por

$$\begin{aligned} &\min \|\mathbf{Ax} - \mathbf{b}\|_2^2, \mathbf{x} \in \mathbb{R}^n \\ &\text{sujeito a } \|x\|_1^1 < t. \end{aligned} \tag{2.10}$$

e pode ser escrito como o problema de otimização irrestrita

$$\min \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1^1, \mathbf{x} \in \mathbb{R}^n. \tag{2.11}$$

não é possível deduzir uma fórmula explícita devido à não-diferenciabilidade introduzida pela norma 1.

Este fato motiva a criação de métodos computacionais para a resolução deste problema. Podemos utilizar técnicas de otimização não diferenciável para encontrar $\mathbf{x}_{\text{lasso}}(\lambda)$ para um determinado λ.

O entendimento sobre o parâmetro λ para o Lasso é o mesmo apresentado na regressão Ridge.

O conjunto de todas as soluções $\mathbf{x}_{\text{lasso}}(\lambda)$, para $0 < \lambda < \infty$, é frequentemente chamado de *caminho de regularização*, e pode ser útil possuir tal conjunto ao se decidir qual modelo (ou seja, qual valor de λ) será utilizado. O processo de decisão pode ser feito por meio de uma técnica de reamostragem, como o método de validação cruzada e suas variações [1].

É possível mostrar que o caminho de regularização do Lasso é a frente de Pareto do problema bi-objetivo

$$\min \{\|Ax - b\|_2^2, \|x\|_1\}. \tag{2.12}$$

De fato, podemos utilizar o método das *somas ponderadas* para escalarizar a função vetorial que define o problema bi-obijetivo em (2.12) e encontrar suas soluções, visto que essas soluções podem ser todas encontradas por meio desse método (ver Teorema 10 no Capítulo 3). Temos então o problema

$$\min\ w_1\|\mathbf{Ax}-\mathbf{b}\|_2^2 + w_2\|\mathbf{x}\|_1^1 \quad , \quad \mathbf{x}\in\mathbb{R}^n. \tag{2.13}$$

para $w_1, w_2 \geq 0$. O problema de minimização (2.13) possui as mesmas soluções do problema

$$\min\ \|\mathbf{Ax}-\mathbf{b}\|_2^2 + \frac{w_2}{w_1}\|\mathbf{x}\|_1^1 \quad , \quad \mathbf{x}\in\mathbb{R}^n. \tag{2.14}$$

para $w_2 > 0$ que, por sua vez, possui as mesmas soluções do problema (2.11), isto é, a forma Lagrangeana do problema de minimização do Lasso, ao se fazer $\lambda = \frac{w_2}{w_1}$.

Portanto, encontrar a frente de Pareto do problema (2.14) é equivalente a encontrar a solução para cada valor de t no problema original. Além disso, podemos fazer uso de técnicas para gerar a frente de Pareto de forma a obter pontos uniformemente distribuídos, como a *escalarização de Tchebychev Along Rays* [5] e o método adaptativo Weighted Sum (soma ponderada), descrito em [9] e [10].

2.2 SUBGRADIENTE

Apresentamos agora algumas definições e resultados usando o conceito de subgradiente, que serão posteriormente utilizados nos métodos computacionais propostos para a geração da frente de Pareto do Lasso.

O subgradiente de uma função é uma generalização, para funções não diferenciáveis, do gradiente definido para funções diferenciáveis, no sentido de fornecer um hiperplano de apoio à função em um ponto dado. O subgradiente de uma função não suave em um ponto é um elemento do conjunto denominado *subdiferencial* da função neste ponto. Formalmente temos

Definição 1. Um subgradiente da função convexa $f : \mathbb{R}^m \to \mathbb{R}$ no ponto $x \in \mathbb{R}^m$ é o vetor $g \in \mathbb{R}^m$ tal que, para todo $y \in \mathbb{R}^m$,

$$f(y) \geq f(x) + g^T(y-x) \tag{2.15}$$

Definição 2. O subdiferencial da função convexa $f : \mathbb{R}^m \to \mathbb{R}$ no ponto $x \in \mathbb{R}^m$ é o conjunto $\partial f(x)$ contendo todos os subgradientes da função f neste ponto.

Se a função f for diferenciável no ponto x, $\partial f(x) = \{\nabla f(x)\}$. A Figura 2 ilustra as definições apresentadas acima.

No exemplo da Figura 2, a função f é diferenciável no ponto A e por esse motivo o subgradiente nesse ponto define um único plano tangente ao gráfico de f em A. No

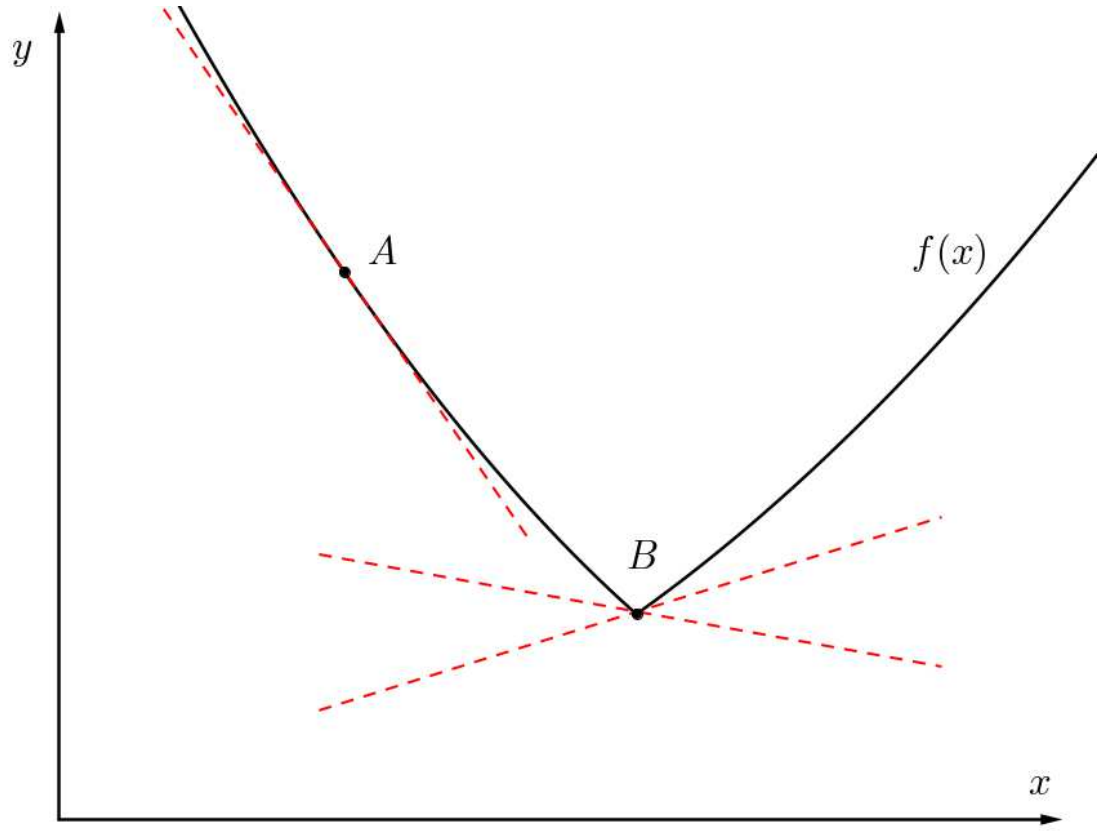

Figura 2 – Exemplo de subgradientes de uma função não diferenciável.

entanto, no ponto B onde f não é diferenciável, existem vários planos de apoio definidos por diferentes subgradientes.

Apresentaremos agora uma condição de otimalidade local para um problema de otimização cuja função objetivo é não suave:

Teorema 3. *O ponto $x \in \mathbb{R}^m$ é um mínimo (ou máximo) local da função $f : \mathbb{R}^m \to \mathbb{R}$ se, e somente se, $0 \in \partial f(x)$, onde $0 \in \mathbb{R}^m$.*

A ideia por trás deste resultado pode ser visualizada da seguinte forma: se o ponto for um mínimo local, então podemos passar por ele um plano de apoio com inclinação 0, de forma que o gráfico da função esteja acima deste plano (pelo menos localmente, em uma vizinhança em torno do ponto). Exploraremos esses conceitos no Capítulo 4, ao discutir os algoritmos para encontrar soluções para o problema de otimização do Lasso.

2.3 PROPRIEDADES ESTATÍSTICAS

Discutimos agora as principais propriedades estatísticas da regressão Lasso e os modelos que ela produz, com relação a demais métodos. Mais especificamente, apresentamos os princípios da parcimônia e a introdução de viés no estimador para reduzir o erro do estimador.

2.3.1 Parcimônia

Ao ajustar um modelo, muitas vezes obtemos variáveis com coeficientes próximos a zero e, em alguns casos, todas as variáveis de entrada obtêm um coeficiente diferente de zero, mesmo que não tenham significância para o modelo. Isso acontece mesmo com métodos de encolhimento como o Ridge. No entanto, o Lasso possui a propriedade de parcimônia, o que significa que a regressão tende a atribuir o valor zero a coeficientes de variáveis que, de outra forma, teriam um coeficiente pequeno mas diferente de zero fornecendo, portanto, modelos com menor número de variáveis.

Entender o significado de cada uma das variáveis é algo desejável em diversas aplicações, e isso só pode ser alcançado se houver um número razoável de variáveis no modelo. Além disso, se uma variável estiver presente, mas seu coeficiente for próximo de zero, essa variável pode não ser significativa para o modelo, podendo interferir na análise deste pelo decisor.

2.3.2 Introdução de Viés

O viés em um estimador é dado por $E(\mathbf{Y}) - \mu$, e representa o desvio da resposta esperada do estimador da média da resposta real. Nesse sentido, parece natural que um estimador preciso para uma dada variável não possua viés. No entanto, é possível que o melhor preditor para um determinado conjunto de teste, no sentido de erro obtido com um conjunto de testes, possua certo nível de viés.

De fato, o erro de teste quadrado do estimador pode ser escrito como

$$SME(\theta) = |E(\mathbf{Y}) - \mu| + Var(\mathbf{Y}) \tag{2.16}$$

onde o primeiro termo do lado direito é igual ao viés do estimador. Assim, para minimizar o erro devemos reduzir o viés e/ou a variância do estimador. Isso torna possível uma situação em que permitimos algum viés a fim de reduzir a variância, de forma que obtemos um estimador com uma média da soma dos erros (MSE) menor que a do melhor estimador sem viés. Nesses casos, esperamos que os métodos de encolhimento (incluindo o Lasso) tenham um desempenho melhor do que métodos com estimador sem viés, como o MQO, e isso justifica o uso do primeiro em situações específicas.

2.4 VARIAÇÕES DO LASSO

Apesar de ter vantagens estatísticas em diversos cenários, o Lasso pode enfrentar dificuldades em obter um modelo consistente em determinados problemas, a depender da natureza das variáveis de entrada. Visando solucionar alguns desses problemas, são propostas modificações ao modelo original do Lasso.

Um dos maiores problemas na utilização do Lasso é a introdução de viés excessivo em conjuntos com grande número de preditores com relação ao número de amostras no

conjunto de dados, uma vez que existe a tendência de atribuir zero a um número considerável de variáveis. Em [14] é introduzido o *lasso relaxado*, que consiste na aplicação do Lasso para encontrar um conjunto de variáveis menor que o inicial com coeficientes diferentes de zero e então aplicar novamente o Lasso sobre o conjunto reduzido de preditores. Para estimar o parâmetro de regularização, pode-se utilizar o método estatístico da *validação cruzada* em cada um dos estágios. A tendência é que durante a segunda aplicação do Lasso, seja escolhido um parâmetro de regularização que represente menor encolhimento, devido ao número reduzido de variáveis e, dessa forma, ocorra a redução do viés introduzido ao modelo.

Outra forma de lidar com o problema acima é alterar a função de penalidade do Lasso, de forma que o encolhimento seja menor em parâmetros com maior valor absoluto. Assim, mantemos a tendência do Lasso de anular parâmetros com valores próximos à zero, mas reduzimos o viés introduzido sobre o conjunto de todos os parâmetros do modelo.

Uma penalização com limitação do desvio absoluto suavizada (SCAD) é apresentada por [7], que substitui na expressão (2.11) o termo $\lambda\|\beta\|$ por $p_\lambda(\beta)$, definida pela expressão

$$p'_\lambda(\theta) = \lambda \left\{ I(\beta \leq \lambda) + \frac{(a\lambda - \beta)_+}{(a-1)\lambda} I(\beta > \lambda) \right\} \tag{2.17}$$

onde $a > 2$ é um parâmetro que deve ser definido à priori. Dessa forma, a penalidade assume a curva exibida na Figura 3. Vale observar que para valores de $|\beta|$ (o valor absoluto de um dos parâmetros em $\boldsymbol{\beta}$) maior que λa, não há penalidade aplicada ao parâmetro e, para $|\beta| < 2\lambda$ a penalidade aplicada é a mesma que no Lasso padrão. Há uma desvantagem nesse caso: o problema de otimização deixa de ser convexo, o que torna o processo de minimização mais custoso computacionalmente e impede o uso de alguns resultados teóricos úteis na resolução do problema.

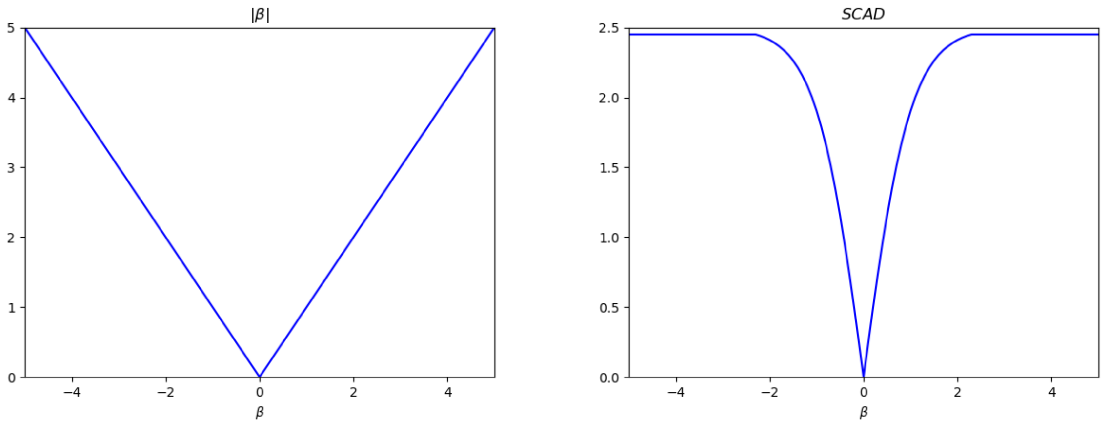

Figura 3 – Exemplo de penalidade do SCAD, em comparação com a restrição original do Lasso.

2.5 TRABALHOS RELEVANTES

O Lasso foi usado em muitas aplicações relacionadas a grande quantidade de dados em vários campos da ciência e engenharia. Em muitos casos, mostra-se que tem um desempenho melhor do que outros métodos, como MQO e Ridge, por exemplo.

Em [20] usa-se uma extensão do Lasso, forçando todos os coeficientes a assumirem valores não negativos, o que leva a melhorias na consistência tanto na seleção do modelo quanto na estimação sob certas condições, que é semelhante à *condição de irrepresentabilidade* para Lasso. Esta condição será discutida mais adiante dada a sua importância para a consistência da seleção de variáveis quando utilizando o Lasso. O método apresentado é então aplicado ao problema de rastreamento de índice no mercado de ações, problema que consiste em selecionar o subconjunto ótimo de ativos que são capazes de rastrear o índice em um determinado período de tempo. Tal problema é conhecido por ser NP-Difícil e diferentes abordagens podem ser encontradas na literatura, como programação matemática [16] e o uso de heurísticas [18].

Em [19] as regressões Lasso, Ridge e Elastic Net são aplicadas a problemas envolvendo dados genômicos, como predição da probabilidade de sobrevivência em pacientes com câncer e classificação de indivíduos obesos e magros, comparando o desempenho de cada método em relação aos demais.

Em [11] e [12] os autores geram intervalos de confiança úteis e válidos para os coeficientes do modelo por inferência pós-seleção aplicada ao Lasso. Em [11] é estabelecido um framework baseado em um resultado pelo qual é possível caracterizar a distribuição de um estimador após a seleção, enquanto [12] apresenta uma estratégia para gerar hipóteses de forma que reduza os custos de exploração de dados, evitando grandes intervalos, que não seriam úteis para analisar os dados.

3 OTIMIZAÇÃO MULTIOBJETIVO

Nesta seção, apresentamos os principais conceitos de otimização multiobjetivo (OM).

3.1 CONCEITOS GERAIS

Um problema de otimização é chamado de multiobjetivo se houver mais de uma função que se queira minimizar (ou maximizar). Normalmente, as funções estão relacionadas de forma que não podem ser independentemente minimizadas ao mesmo tempo e, além disso, a melhora do valor de uma função implica em piorar o valor da outra função.

Formalmente, um problema de otimização multiobjetivo (POM) é da forma

$$\begin{aligned} &\text{minimizar } \{f_1(x), f_2(x), \ldots, f_m(x)\} \\ &\text{sujeito a } \mathbf{x} \in S \end{aligned} \tag{3.1}$$

onde $m \geq 2$, as funções $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ são contínuas e $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ é o vetor das funções objetivo. Aqui S é chamada região viável e é definida pelas restrições do problema.

Para discutir melhor esses conceitos, temos que definir o que é considerado um ponto ótimo em um POM. Um ponto será considerado ótimo se não houver nenhum outro ponto no conjunto viável que melhore o valor de uma função sem piorar o valor de pelo menos uma outra função. Estes são chamados de ótimos de Pareto, em homenagem ao economista italiano Vilfredo Pareto (1848-1923). Formalmente temos a

Definição 4. Um vetor $\mathbf{x}^* \in S$ é dito *Pareto ótimo* se não existe outro vetor $\mathbf{x} \in S$ tal que $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ para todo $i = 1, \ldots, k$ e $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ para pelo menos um índice $i \in \{1, \ldots, n\}$.

Na definição 4 estabelece-se o conceito de um ótimo de Pareto no espaço das variáveis de decisão. Define-se também o conceito de ótimo de Pareto no espaço objetivo, conforme a

Definição 5. Um vetor $\mathbf{z}^* \in Z$ é dito *Pareto ótimo* no espaço objetivo se não existe outro vetor $\mathbf{z} \in Z$ tal que $z_i \leq z_i^*$ para todo $i = 1, \ldots, k$ e $z_i < z_i^*$ para pelo menos um índice $i \in \{1, \ldots, n\}$.

É fácil ver que esta definição é equivalente a dizer que um vetor $\mathbf{z}^* \in Z$ é Pareto ótimo se existe $\mathbf{x}^* \in S$ tal que $\mathbf{z}^* = f(\mathbf{x}^*)$. Também temos o conceito de *ótimo fraco de Pareto*, que intuitivamente é um vetor na região factível S tal que não existe outro vetor em S que melhore o valor de todas as coordenadas da função objetivo ao mesmo tempo. Formalmente, temos a

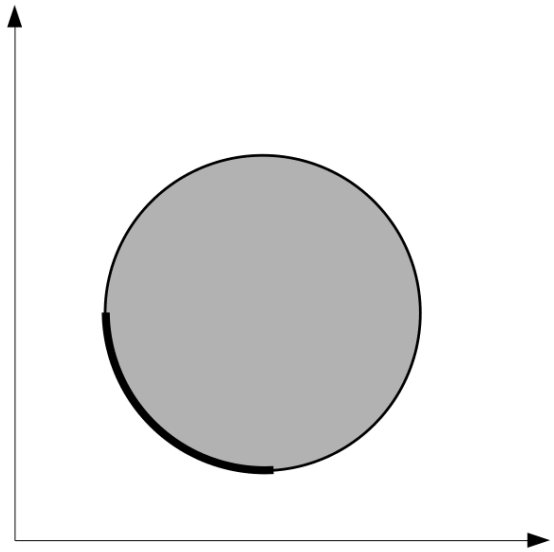

Figura 4 – Exemplo de frente de Pareto (linha destacada) no espaço objetivo.

Definição 6. Um vetor $\mathbf{x}^* \in S$ é *ótimo fraco de Pareto* se não existe outro vetor $\mathbf{x} \in S$ tal que $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ para todo $i = 1, \ldots, k$.

Também define-se o *ótimo fraco de Pareto* no espaço objetivo, como segue na

Definição 7. Um vetor $\mathbf{z}^* \in Z$ é *ótimo fraco de Pareto* no espaço objetivo se não existe outro vetor $\mathbf{z} \in Z$ tal que $z_i < z_i^*$ para todo $i = 1, \ldots, k$.

É possível ver que todo ótimo de Pareto é ótimo fraco de Pareto, e assim, de agora em diante, um ótimo fraco de Pareto ou um ótimo de Pareto serão chamados de *pontos de Pareto*, e o conjunto desses pontos será chamado de frente de Pareto.

Podemos visualizar os conceitos definidos acima em alguns exemplos. A figura 4 mostra um exemplo simples de uma frente de Pareto no espaço objetivo $\mathbb{R}^2$.

A Figura 5 mostra um exemplo de uma frente de Pareto no espaço objetivo $\mathbb{R}^2$, na qual os pontos na linha paralela ao eixo horizontal são pontos ótimos fracos de Pareto, enquanto a parte restante da frente contém apenas pontos ótimos de Pareto. Apesar dos exemplos mostrados, no caso geral a frente de Pareto não precisa ser conexa, nem convexa. A figura 6 mostra um exemplo de frente de Pareto não convexa no espaço objetivo $\mathbb{R}^2$.

3.2 SOLUCIONANDO PROBLEMAS MULTIOBJETIVO

Estabelecidos os conceitos acima, vamos apresentar métodos computacionais para a obtenção da solução de um problema multiobjetivo. Daremos ênfase aos métodos chamados *à posteriori*, isto é, aqueles que assumem que todos os pontos da frente de Pareto (ou uma parte destes) foram calculados. Os resultados apresentados nessa seção, bem como suas demonstrações, podem ser encontrados na íntegra em [15].

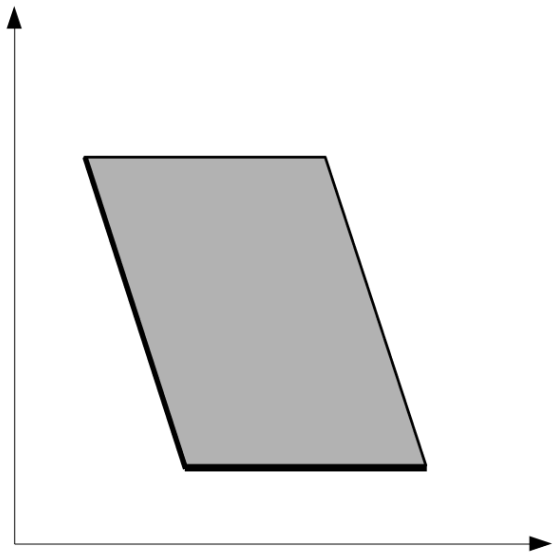

Figura 5 – Exemplo de frente de Pareto com ótimos de Pareto fraco (linha horizontal destacada) no espaço objetivo.

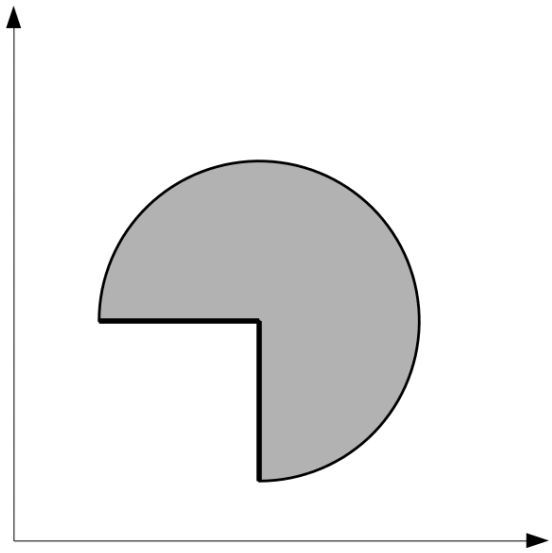

Figura 6 – Exemplo de frente de Pareto não convexa (linha destacada) no espaço objetivo.

3.2.1 Método da soma ponderada

O método da *soma ponderada* consiste em definir uma função escalar atribuindo-se pesos a cada uma das funções objetivo, e então encontrar os mínimos para essa função escalar. O problema fica definido como

$$\begin{aligned} &\text{minimizar } \sum_{i=1}^{k} w_i f_i(x) \\ &\text{sujeito a } \mathbf{x} \in S \end{aligned} \tag{3.2}$$

onde $w_i \geq 0$. Também é usual supor que os valores dos pesos utilizados estão normalizados, de forma que $\sum_{i=1}^{k} w_i = 1$. Apresentamos a seguir alguns resultados teóricos para este método. Primeiramente, temos os resultados quanto à otimalidade das soluções obtidas para cada conjunto de parâmetros $\mathbf{w} = (w_1, w_2, \ldots w_n)$.

Teorema 8. *A solução do problema (3.2) é um ótimo de Pareto fraco para qualquer conjunto de pesos $w_i \geq 0$ para $i = 1, 2, \ldots, n$.*

Para o caso da otimalidade de Pareto (estrita), temos o seguinte resultado.

Teorema 9. *A solução do problema (3.2) é um ótimo de Pareto se os coeficientes são estritamente positivos, isto é, $w_i > 0$ para $i = 1, 2, \ldots, n$.*

Um resultado particularmente importante para o problemas de otimização multi-objetivo linear, e para o problema de otimização associado ao Lasso, é o seguinte:

Teorema 10. *Se um problema de otimização multi-objetivo é convexo então, para toda solução ótima de Pareto $\mathbf{x}^* \in S$, existe um vetor $\mathbf{w}$ de pesos não-negativos com $\sum_{i=1}^{n} w_i = 1$ tal que $\mathbf{x}^*$ é uma solução para o problema (3.2).*

O resultado anterior garante que todos os ótimos de Pareto para um problema multi-objetivo convexo podem ser encontrados por meio do método da soma ponderada. Esse Teorema foi utilizados para mostrar a correspondência direta entre o problema de otimização bi-objetivo e o problema de otimização irrestrita (2.11) associado ao Lasso visto no Capítulo 2.

Além disso, utilizando o fato de que o problema associado ao Lasso é convexo, podemos utilizar o seguinte resultado para garantir que as soluções encontradas na aplicação ao Lasso são ótimos de Pareto (estritos).

Teorema 11. *Se um problema de otimização multi-objetivo é convexo então $w_i > 0$ para $i = 1, 2, \ldots, n$ é a condição necessária e suficiente para que uma solução seja ótima de Pareto (estrito).*

3.2.2 Método da ϵ-restrição

O método da ϵ-restrição consiste em otimizar apenas uma das funções objetivo do problema, utilizando as demais como restrições, limitando estas a um valor ϵ_i a ser definido. O método da ϵ-restrição é definido como

$$\begin{aligned} &\text{minimizar } f_i(x) \\ &\text{sujeito a } f_j(\mathbf{x}) \leq \epsilon_j \text{ para todo } j = 1, 2, ..., k, \quad i \neq j, \\ &\qquad \mathbf{x} \in S \end{aligned} \tag{3.3}$$

O problema acima é conhecido como o problema ϵ-restrito. Assim como no caso do método da soma ponderada, temos alguns resultados teóricos úteis, os quais apresentamos a seguir.

Teorema 12. *A solução do problema ϵ-restrito é um ótimo fraco de Pareto.*

Teorema 13. *Um vetor $\mathbf{x} \in S$ é ótimo de Pareto para um problema multi-objetivo se, e somente se, é também uma solução para o problema ϵ-restrito (3.3) para todo $i = 1, 2, ..., k$, onde, em cada caso, $\epsilon_j = f_j(\mathbf{x})$ para $j = 1, 2, ..., k,\ i \neq j$.*

Vale observar que, de acordo com o Teorema 13, todas as soluções ótimas de Pareto de qualquer problema de otimização multi-objetivo podem ser encontradas pelo método da ϵ-restrição. Nesse caso, não há hipótese sobre a convexidade do problema, como era necessário no método das somas ponderadas.

Os resultados seguintes são úteis para problemas com soluções únicas.

Teorema 14. *Se um vetor $\mathbf{x} \in S$ é a única solução para o problema ϵ-restrito (3.3) para algum $i \in \{1, 2, ..., k\}$ com $\epsilon_j = f_j(\mathbf{x})$ para $j = 1, 2, ..., k,\ i \neq j$, então este é ótimo de Pareto para o problema multi-objetivo.*

No teorema abaixo, consideramos $\boldsymbol{\epsilon}$ o vetor de dimensão $k - 1$ contendo todos os valores de ϵ_j para $j = 1, 2, ..., k,\ i \neq j$ na formulação do problema ϵ-restrito (3.3).

Teorema 15. *Se $\mathbf{x} \in S$ é a única solução para o problema ϵ-restrito (3.3), então $\mathbf{x}$ é ótimo de Pareto para qualquer vetor de restrições $\boldsymbol{\epsilon}$.*

3.2.3 Método hibrido

O método híbrido resulta da combinação dos métodos da soma ponderada e da ϵ-restrição apresentados anteriormente, de forma que o problema a ser resolvido é

$$\begin{aligned} &\text{minimizar } \sum_{i=1}^{k} w_i f_i(x) \\ &\text{sujeito a } f_j(\mathbf{x}) \leq \epsilon_j \text{ para todo } j = 1, 2, ..., k, \\ &\qquad \mathbf{x} \in S, \end{aligned} \tag{3.4}$$

onde $w_i > 0$ para todo $i = 1, 2, ...k$. Para esse método, apresentamos os seguintes resultados sobre a otimalidade das soluções.

Teorema 16. *A solução para o problema hibrido (3.4) é um ótimo de Pareto para qualquer vetor de restrições* $\boldsymbol{\epsilon} \in \mathbb{R}^k$.

Teorema 17. *Se* $\mathbf{x} \in S$ *é um ótimo de Pareto para o problema multi-objetivo, então* $\mathbf{x}$ *é uma solução para o problema hibrido (3.4) com*

$$\boldsymbol{\epsilon} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x}))$$

De forma geral, o método hibrido proporciona a capacidade de encontrar qualquer solução ótimo de Pareto independentemente da convexidade do problema, uma vantagem trazida do método da ϵ-restrição, mas ao mesmo tempo não possui a desvantagem de ter-se que lidar com vários problemas de otimização distintos, pois temos uma única função objetivo, como no método da soma ponderada. Computacionalmente, no entanto, o método hibrido é parecido com o método da ϵ-restrição, visto que é necessário estimar os valores para cada um dos parâmetros de restrição [15].

4 ALGORITMOS

Neste capítulo apresentamos os conceitos necessários para obter aproximações da frente de Pareto do problema do Lasso de forma mais eficiente. Apresentaremos também o algoritmo proposto para realizar uma aproximação da frente de Pareto com um número reduzido de pontos calculados.

4.1 INTERPOLAÇÃO E APROXIMAÇÃO

De forma a reduzir o tempo necessário para construir a frente de Pareto do Lasso, desejamos obter uma curva que aproxima tal frente a partir de um conjunto de pontos que sabemos pertencer a esta frente. Esse conjunto de pontos deve ser calculado, e desejamos minimizar o seu tamanho, visto que o custo computacional está em grande parte associado à quantidade de pontos necessários para a aproximação.

Um caminho natural para a obtenção de uma aproximação de uma curva a partir de um número finito de pontos pertencentes a essa curva é a *interpolação*. Em geral, métodos de interpolação têm como objetivo definir uma curva que se adéque a um conjunto de pontos, e para que seja possível realizar uma aproximação da curva verdadeira com tais métodos é necessário saber a priori a natureza da curva. Para ilustrar esse ponto, considere o seguinte exemplo: suponha que desejamos interpolar o conjunto de pontos exibidos na Figura 7 de forma a recuperar a curva aos quais pertencem.

A Figura 8 exibe diferentes tipos de interpolação polinomial, a saber, linear, quadrática e cúbica, a partir dos 5 pontos mostrados na Figura 7. Como os pontos foram gerados a partir da função cúbica $y(x) = x^3 - 10x + 1$, a curva que melhor se ajusta é aquela obtida por meio da interpolação polinomial de ordem 3.

Por outro lado, se aqueles pontos pertencessem ao gráfico de uma função linear por partes, a curva que melhor se ajustaria a esses pontos seria também linear por partes. Para determinação da curva que melhor se ajusta ao conjunto de pontos é desejável um certo conhecimento prévio da função cujo gráfico contém esses pontos e que desejamos aproximar por meio da interpolação.

A seguir apresentamos os métodos de interpolação adotados nesse trabalho para fins de comparação no estudo computacional realizado.

4.1.1 Interpolação Linear

A interpolação linear consiste geometricamente em ligar dois pontos no espaço objetivo $\mathbb{R}^n$ por um seguimento de reta. Formalmente, sejam $\mathbf{x}^*(t_1)$ e $\mathbf{x}^*(t_2)$ duas soluções ótimas do problema bi-objetivo do Lasso para os parâmetros de regularização t_1 e t_2, respectivamente, com $t_1 < t_2$. Então, para qualquer t tal que $t_1 \leq t \leq t_2$, a aproximação

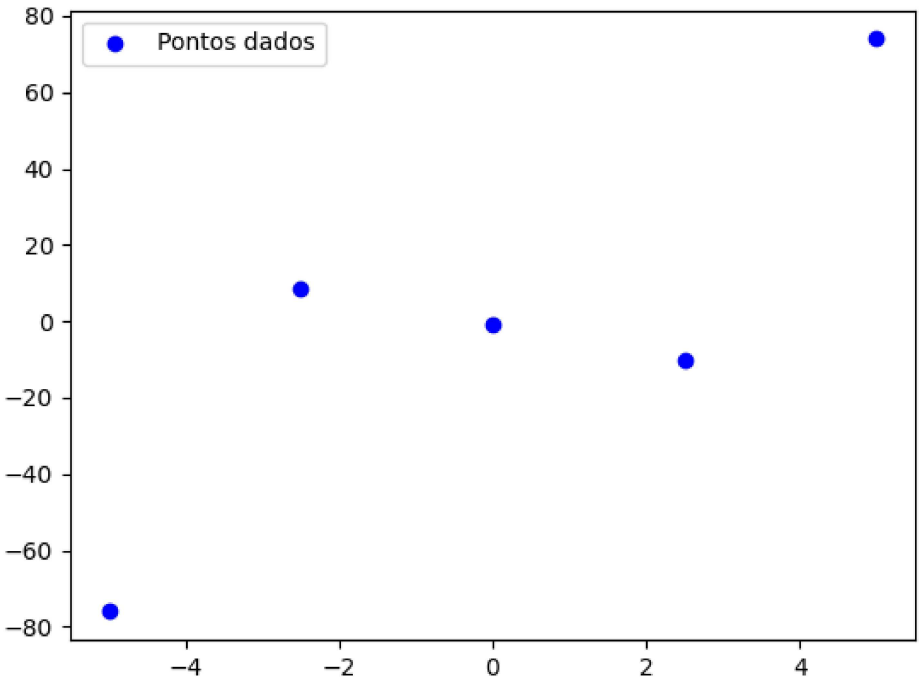

Figura 7 – Conjunto de dados gerados.

de $\mathbf{x}^*(t)$ é dada por

$$\hat{\mathbf{x}}^*(t) = \frac{t - t_1}{t_2 - t_1}\mathbf{x}^*(t_2) + \frac{t_2 - t}{t_2 - t_1}\mathbf{x}^*(t_1) \tag{4.1}$$

Para a geração da frente de regularização do Lasso será calculado $\mathbf{x}^*(t)$ para diversos valores de t, de forma que a interpolação linear poderá ser feita para cada par de valores de t adjacentes. A Figura 9 apresenta um exemplo de interpolação dos pontos na frente de regularização utilizando-se interpolação linear.

Apesar de muito simplística, a interpolação linear possui bom desempenho quando se trata de funções lineares por partes. Uma interpolação ótima para funções Lispschitz é encontrada em [2]. No entanto, tal método depende da estimativa de uma constante de Lispschitz à priori, o que não é factível no caso em estudo, pois seria necessário gerar um número muito grande de pontos da frente de regularização para se determinar a constante de Lispschitz, o que inviabilizaria a interpolação linear.

4.1.2 Polinômio de Lagrange

O Polinômio de Lagrange é o polinômio que interpola todos os pontos de um conjunto de dados com o menor grau possível. Sejam $\mathbf{x}^*(t_1), \mathbf{x}^*(t_2), \ldots, \mathbf{x}^*(t_k) \in \mathbb{R}^n$ pontos da frente de regularização, com $t_1 < t_2 < \cdots < t_k$. Para $t \in [0, \infty)$, temos a aproximação

$$L(t) = \sum_{i=1}^{k} \mathbf{x}^*(t_i) l_i(t) \qquad l_i(t) = \prod_{j=1, j\neq i}^{k} \frac{t - t_j}{t_i - t_j} \tag{4.2}$$

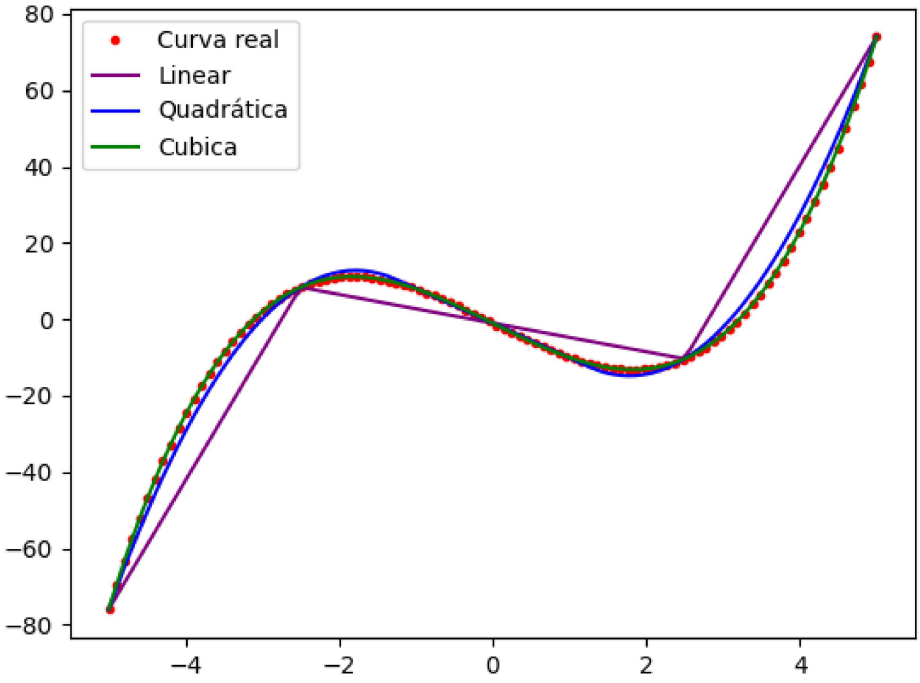

Figura 8 – Interpolação por meio de diferentes métodos.

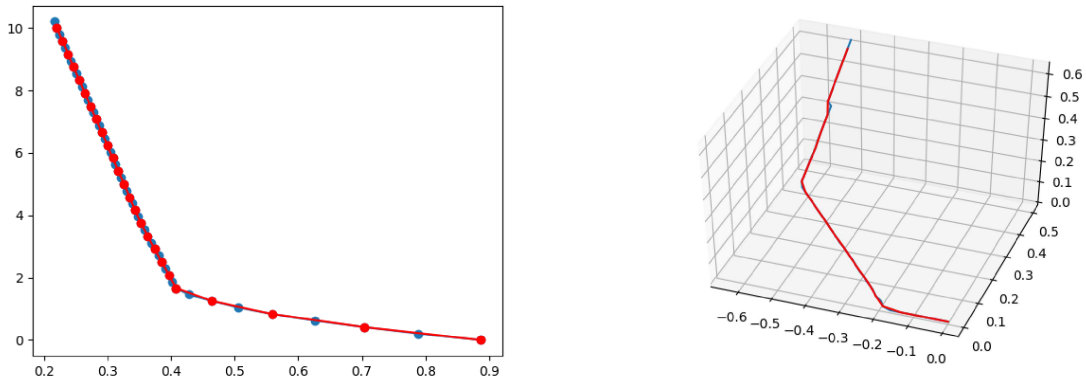

Figura 9 – Exemplo de aplicação da interpolação linear à frente de regularização do Lasso.

Assim temos $\hat{\mathbf{x}}^*(t_i) = \mathbf{x}^*(t_i)$ para todo $i = 1, \ldots, k$, onde $\hat{\mathbf{x}}^*$ é a curva definida pela interpolação de Lagrange. No entanto, caso o grau do polinômio resultante seja alto, podem ocorrer oscilações excessivas na curva obtida, comportamento conhecido como Fenômeno de Runge. Como o polinômio pode ter grau $k - 1$ no máximo, um número maior de pontos leva, em geral, a ocorrência desse tipo de oscilação. O exemplo exibido na Figura 10 mostra a interpolação de uma frente de Pareto onde ocorre o fenômeno de oscilação próximo às extremidades.

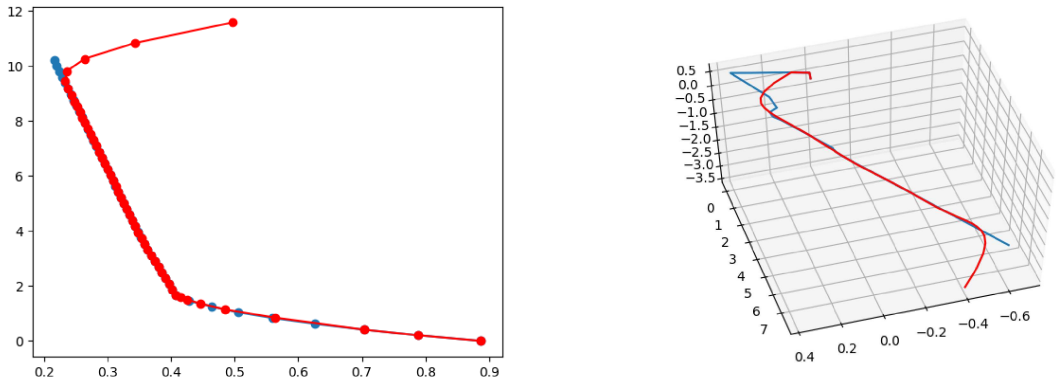

Figura 10 – Exemplo da ocorrência do fenômeno de Runge na utilização do Polinômio de Lagrange.

4.1.3 Regressão Polinomial

A Regressão Polinomial é uma forma de aplicação da regressão linear clássica, onde consideramos as m primeiras potências de uma única variável de entrada. No nosso caso em tela, tal variável é o parâmetro de regularização t, e os dados utilizados para ajustar o modelo são os pontos $\mathbf{x}^*(t_1), \mathbf{x}^*(t_2), \ldots, \mathbf{x}^*(t_k)$ da frente de Pareto. O modelo é representado por

$$x_{j,i} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \cdots + \beta_m t_i^m \tag{4.3}$$

Considerando o conjunto de dados completo, podemos escrever

$$\begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \\ \vdots \\ x_{i,n} \end{bmatrix} = \begin{bmatrix} 1 & t1 & t1^2 & \ldots & t1^m \\ 1 & t2 & t2^2 & \ldots & t2^m \\ 1 & t3 & t3^2 & \ldots & t3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & tn & tn^2 & \ldots & tn^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} \tag{4.4}$$

Para encontrar os coeficientes ótimos β_i^*, aplicamos a expressão conhecida para ajuste do modelo com MQO, encontrando $\boldsymbol{\beta}^* = \left(\mathbf{T}^\top \mathbf{T}\right)^{-1} \mathbf{T}^\top x$, onde $\mathbf{T}$ é a matriz dos dados mostrada acima. A complexidade para obter essa aproximação é maior que a dos demais métodos apresentados anteriormente, embora ainda seja significativamente menor do que a obtenção dos pontos da frente de Pareto em si.

A Figura 11 exemplifica o uso da regressão polinomial para obter uma aproximação da frente de Pareto.

4.1.4 FFT

A Transformada de Fourier Discreta (FFT) pode ser adaptada para representar funções diferenciáveis por partes, tal como $\mathbf{x}^*$, conforme descrito em [6]. O uso para a

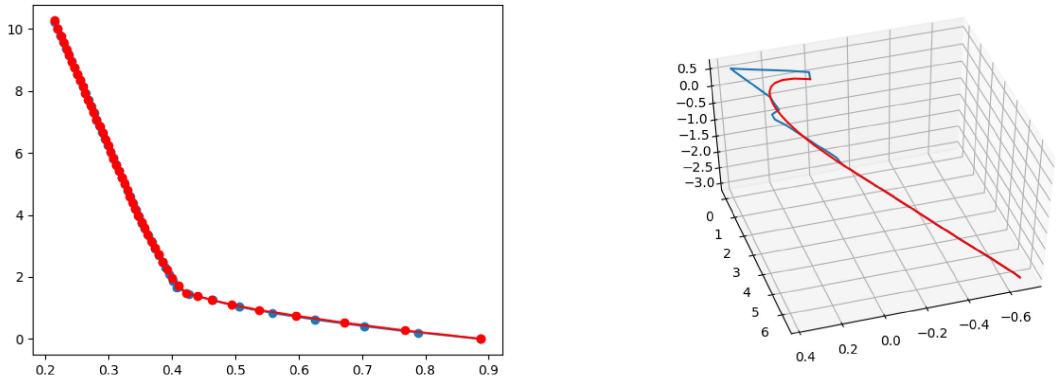

Figura 11 – Exemplo de aplicação da Regressão Polinomial à frente de Pareto do Lasso.

aproximação de funções pode ser feito com uma amostragem do conjunto de dados, seguido da passagem ao domínio das frequências e remoção das frequências mais elevadas, de forma a filtrar possíveis ruídos. Utiliza-se então a transformada inversa para retornar ao domínio da variável independente, neste caso, o parâmetro de regularização t.

A Figura 12 apresenta um exemplo de aproximação usando FFT com a abordagem descrita acima.

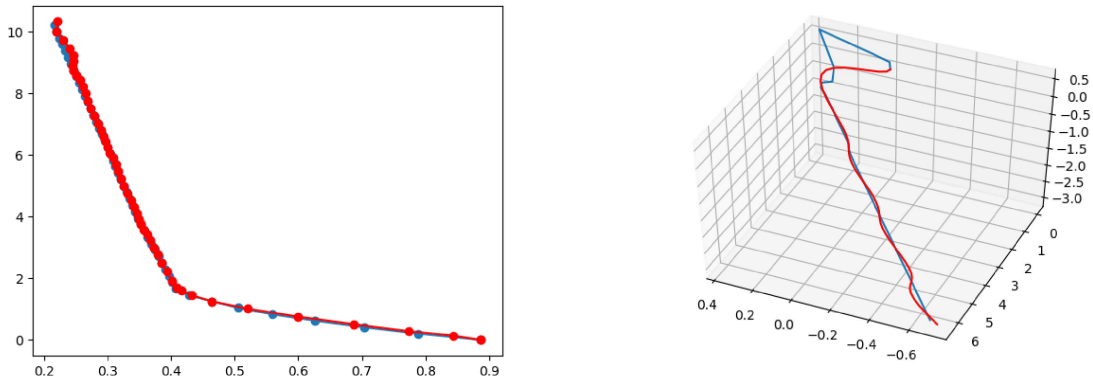

Figura 12 – Exemplo de aplicação do FFT para frente de Pareto do Lasso.

4.2 ALGORITMO PARA GERAÇÃO DA FRENTE REGULARIZAÇÃO COM ESTIMATIVA DE PONTOS INICIAIS

A frente de regularização definida pela curva $\mathbf{x}^*(t) : \mathbb{R}_+ \mapsto \mathbb{R}^n$, composta pelas soluções do problema de minimização apresentado em (2.10), é linear por partes [17]. O algoritmo que vamos propor nesta seção utiliza tal propriedade para gerar pontos inicias para os algoritmos que serão escolhidos para a solução do problema (2.10).

A Figura 13 mostra um exemplo de frente de regularização do Lasso em $\mathbb{R}^3$, onde é possível visualizar a propriedade de linearidade por partes que é explorada.

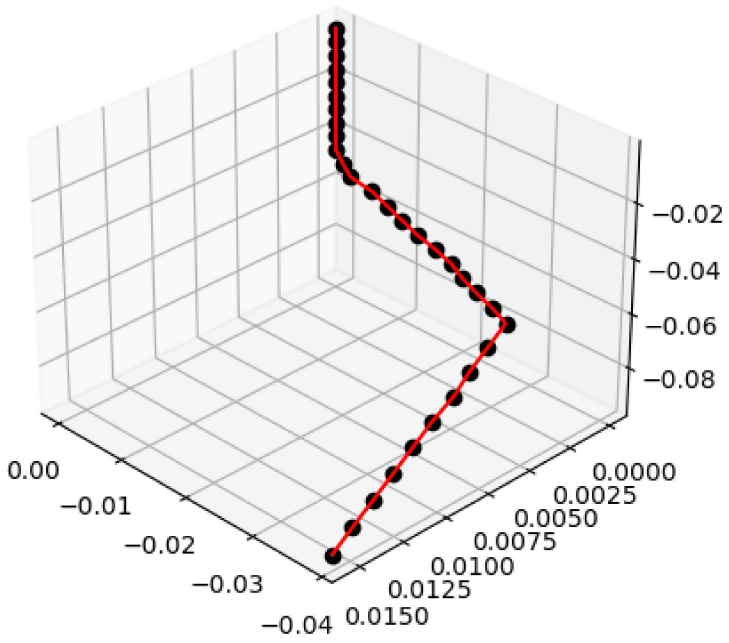

Figura 13 – Exemplo de pontos na frente de regularização em 3 dimensões.

Em outras palavras, utiliza-se o conhecimento de que $\mathbf{x}^*$ é linear por partes para auxiliar na escolha do próximo ponto inicial x_0 para o algoritmo que minimização que determinará $\mathbf{x}^*(t)$, para cada t escolhido previamente.

A seguir descrevemos o algoritmo proposto para a geração da frente de regularização do Lasso, que proporciona uma melhora na performance do método de minimização responsável pela obtenção de $\mathbf{x}^*(t)$:

Denotamos por i o índice da iteração e por t_i o valor do parâmetro de regularização utilizado naquela iteração.

Passo 1 Iniciar o conjunto das soluções do problema (2.10) que chamaremos de $\mathbf{X}$ com pelo menos dois pontos da frente de regularização, podendo um deles ser o vetor nulo.

Para cada valor de t escolhido:

Passo 2 Obter a direção $\mathbf{v}$ com base nos valores de $\mathbf{x}^*(t_{i-1})$ e $\mathbf{x}^*(t_{i-2})$, sendo

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|},$$

onde

$$\mathbf{u} = \mathbf{x}^*(t_{i-1}) - \mathbf{x}^*(t_{i-2}).$$

Passo 3 Obter o tamanho do passo α na direção $\mathbf{v}$, dado por

$$\alpha = t_i - t_{i-1}.$$

Passo 4 Determinar x_0 como a aproximação linear

$$x_0 = \mathbf{x}^*(t_{i-1}) + \alpha\mathbf{v}.$$

Passo 5 Utilizar o método de minimização escolhido com a aproximação inicial x_0 para obter $\mathbf{x}^*(t_i)$.

A descrição em formato de pseudocódigo do algoritmo é mostrada abaixo.

Algoritmo 1: Algoritmo para geração da frente regularização com estimativa de pontos iniciais

Resultado: Aproximação para a frente de regularização do Lasso
$X \leftarrow \{\mathbf{x}^*(t_{i-2}), \mathbf{x}^*(t_{i-1})\}$;
para *cada valor de* t_i **faça**
  $x_0 \leftarrow \mathbf{0}$;
  se $|X| \geq 2$ **então**
    $v \leftarrow \mathbf{x}^*(t_{i-1}) - \mathbf{x}^*(t_{i-2})$;
    $\alpha \leftarrow t_i - t_{i-1}$;
    $x_0 \leftarrow \mathbf{x}^*(t_{i-1}) + \alpha v$;
  fim
  $\mathbf{x}^*(t_i) \leftarrow$ minimizar_lasso(t_i, x_0);
fim

No Algoritmo 1, a ideia central é prover uma estimativa inicial x_0 precisa para o algoritmo de otimização que irá determinar $\mathbf{x}^*(t)$, com o intuito de reduzir o número de iterações ou o esforço computacional para a obtenção dos pontos da frente de regularização. Para o sucesso do algoritmo 1, o algoritmo de otimização escolhido deve ser capaz de encontrar o valor de $\mathbf{x}^*(t)$ mesmo no caso em que a aproximação inicial possua certo erro.

Nas subseções seguintes, apresentamos algoritmos para a execução do Passo 5 do Algoritmo 1.

4.2.1 Métodos de Regiões de Confiança

O método das regiões de confiança é um método iterativo que aproxima da função objetivo por uma função quadrática ou linear, delimitando uma região onde o erro cometido

pela aproximação é relativamente pequeno [4].

O método calcula, em cada iteração, o ótimo da função quadrática aproximada considerando apenas a região de confiança. O processo é repetido considerando o novo ponto ótimo como centro da região de confiança.

Um critério de parada comumente utilizado neste tipo de algoritmo é a comparação da razão de melhoria (ou razão de descida) do valor da função objetivo com um valor pré determinado, de forma que se o método não apresenta melhorias significativas a cada iteração, o processo iterativo é interrompido, e a solução atual é dada como ótimo local.

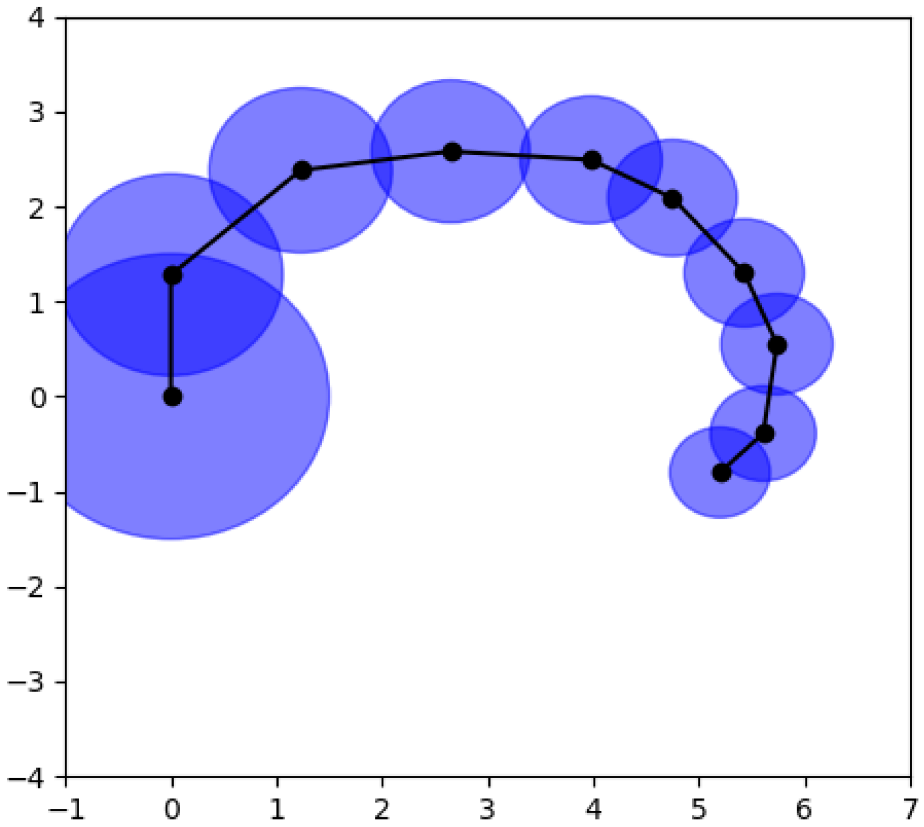

Figura 14 – Exemplo de conjunto de iterações do método de Regiões de Confiança.

A Figura 14 apresenta um exemplo de execução do método de Regiões de Confiança para um problema irrestrito. Observe que, em pontos onde a confiança é menor, o raio da região de confiança também é reduzido, de forma a tornar o processo de escolha da direção mais preciso. No Capítulo 5, esse método será identificado por *RC*.

4.2.2 Programação Quadrática Sequencial por Mínimos Quadrados

Os métodos de Programação Quadrática Sequencial (PQS) se utilizam de aproximações sucessivas do problema de otimização restrito original por meio de problemas de otimização restritos nos quais a função objetivo é quadrática, e as restrições são lineares. A ideia por trás destes métodos é que a solução de tais sub-problemas aproximados é mais fácil que a solução do problema original, cuja solução é um ponto limite da sequência formada pelas soluções dos problemas aproximados [21].

O método que abordaremos neste trabalho, em particular, realiza uma transformação dos sub-problemas de minimização utilizando uma decomposição de Cholesky sobre a matriz Hessiana de tal forma que estes sub-problemas possam ser resolvidos efetivamente pelo método dos mínimos quadrados. No Capítulo 5, esse método será identificado por *SLSQP*, seu nome na biblioteca de otimização utilizada.

4.2.3 Otimização Restrita por Aproximação Linear

O método de Otimização restrita por aproximação linear utiliza, assim como os métodos mencionados anteriormente, aproximações do problema de otimização inicial. Nesse caso, a aproximação é feita por problemas de programação linear que, em cada iteração do algoritmo, são resolvidos para se obter uma nova solução candidata. A aproximação para o problema linear é melhorada a cada iteração, de forma que as soluções candidatas convirjam para a solução do problema original [3].

Uma grande vantagem deste método é não necessitar de conhecimento prévio sobre a derivada da função objetivo, o que facilita a aplicação do método em uma gama de problemas. No Capítulo 5, esse método será identificado por *COBYLA*, seu nome na biblioteca de otimização utilizada.

4.3 ALGORITMO SEM MINIMIZAÇÃO

Utilizando as definições apresentadas anteriormente para o problema de otimização do Lasso, podemos obter uma abordagem para obter a solução para um parâmetro de regularização t representada por $\mathbf{x}^*(t)$, sem a utilização de métodos de otimização.

Para isso, considere a formulação em (2.11). Seja $f_1(x) = \|Ax - b\|_2^2$ e $f_2(x) = \|x\|_1$. Dessa forma o problema do Lasso fica definido como

$$\text{minimizar } f_1(x) + \lambda f_2(x), \qquad x \in \mathbb{R}^n \tag{4.5}$$

Dado que a função deste problema não é diferenciável, a condição de otimalidade para um ponto $\mathbf{x} \in \mathbb{R}^n$ é de que

$$\mathbf{0} \in \partial \left[f_1(x) + \lambda f_2(x) \right] = \{ \partial f_1(\mathbf{x}) + \lambda \partial f_2(\mathbf{x}) \} \tag{4.6}$$

Sabemos que $f_1(\mathbf{x})$ é diferenciável, de forma que seu único subgradiente em $\mathbf{x}$ é

$$\partial f_1(\mathbf{x}) = \{\nabla f_1(\mathbf{x})\} = \{2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b}\} \tag{4.7}$$

Por outro lado, o subgradiente de f_2 é dado por

$$\partial f_2(\mathbf{x}) = \partial\|\mathbf{x}\|_1 = \{\mathbf{v} \in \mathbb{R}^n; \|\mathbf{v}\|_\infty \leq 1, \mathbf{v}^T\mathbf{x} = \|\mathbf{x}\|_1\} \tag{4.8}$$

Para $\mathbf{v} \in \partial f_2(x)$, podemos escrever a condição de otimalidade apresentada acima como

$$2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} + \lambda\mathbf{v} = \mathbf{0} \tag{4.9}$$

Supondo que $\mathbf{A}^T\mathbf{A}$ é inversível, podemos reescrever a equação resolvendo para $\mathbf{x}$, como segue:

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\left[\mathbf{A}^T\mathbf{b} - \frac{\lambda}{2}\mathbf{v}\right] \tag{4.10}$$

Observe que $\mathbf{x}_{ls} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$, de forma que

$$\mathbf{x} = \mathbf{x}_{ls} - \frac{\lambda}{2}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v} \tag{4.11}$$

Sabemos que a solução para o parâmetro de regularização t deve satisfazer $\|\mathbf{x}\|_1 = t$. Aplicando a definição do subgradiente de f_2 em (4.8), podemos escrever

$$\|\mathbf{x}\|_1 = \mathbf{v}^T\mathbf{x} = \mathbf{v}^T\left[\mathbf{x}_{ls} - \frac{\lambda}{2}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v}\right] = t \tag{4.12}$$

Podemos então determinar uma expressão para λ, como a seguir

$$\lambda = \frac{2(\mathbf{v}^T\mathbf{x} - t)}{\mathbf{v}^T\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v}} \tag{4.13}$$

A partir das equações (4.11) e (4.13) podemos perceber que não é possível obter uma expressão para $\mathbf{x}$, pois existe uma dependência com o subgradiente $\mathbf{v}$ para obtenção de λ e, para determinar $\mathbf{v}$ dependemos do valor de $\mathbf{x}$.

Uma abordagem para eliminar esse impasse é adotar $\mathbf{v} = (\delta(x_1), \delta(x_2), \ldots \delta(x_n))$, onde

$$\delta(x) = \begin{cases} 1, \text{ se } x > 0, \\ -1, \text{ se } x < 0, \\ d \in [-1, 1], \text{ se } x = 0 \end{cases} \tag{4.14}$$

Como não podemos afirmar a princípio quais das coordenadas de $\mathbf{x}$ são nulas, devemos escolher $\mathbf{v} \in [-1, 1]^n$. A partir da escolha de $\mathbf{v}$ podemos determinar o valor de λ

e com isso de $\mathbf{x}$. No entanto, é necessário verificar se o vetor $\mathbf{v}$ escolhido atende a condição em (4.12).Esse processo será realizado repetidas vezes pelo algoritmo até que se encontre um subgradiente $\mathbf{v}$ que atenda às condições impostas, com alguma tolerância pré-definida.

Aqui, vale notar que caso possamos determinar o sinal de um conjunto de coordenadas de $\mathbf{x}$ de antemão, poderemos reduzir o espaço de busca pelo valor de $\mathbf{v}$, melhorando a performance do algoritmo responsável pelo cálculo da frente de regularização. No contexto do algoritmo para construção da frente faremos uso dessa propriedade da função δ definida.

Para proporcionar uma melhora na performance do algoritmo, é possível impor alguns limites sobre o valor de λ, de forma que possamos limitar o espaço de busca para $\mathbf{v}$. Mais especificamente, podemos determinar dois limites inferiores, dos quais assumiremos o maior, e um limite superior para λ.

Da condição de otimalidade (4.9) podemos escrever

$$\mathbf{v} = \frac{2}{\lambda}(\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x}). \tag{4.15}$$

Pela definição para o subgradiente de f_2 em (4.8), temos

$$\left\|\frac{2}{\lambda}(\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x})\right\|_\infty \leq 1 \tag{4.16}$$

$$\left\|\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x}\right\|_\infty \leq \frac{\lambda}{2} \tag{4.17}$$

Além disso, podemos utilizar a igualdade $\|\mathbf{v}\|_1 = t$ para escrever

$$\frac{2}{\lambda}(\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x})^T\mathbf{x} = \frac{2}{\lambda}\left[(\mathbf{A}^T\mathbf{b})^T\mathbf{x} - (\mathbf{A}^T\mathbf{A}\mathbf{x})^T\mathbf{x}\right] = t \tag{4.18}$$

Observe agora que, pela desigualdade triangular da norma e por (4.17) temos

$$\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty - \left\|\mathbf{A}^T\mathbf{A}\mathbf{x}\right\|_\infty \leq \left\|\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x}\right\|_\infty \leq \frac{\lambda}{2} \tag{4.19}$$

Agora, pela conhecida desigualdade entre as normas

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq n\|\mathbf{x}\|_\infty \tag{4.20}$$

e pela desigualdade geral

$$|\mathbf{x}^T\mathbf{y}| \leq \|\mathbf{x}\|\mathbf{y}\| \tag{4.21}$$

temos

$$\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty - \left\|\mathbf{A}^T\mathbf{A}\mathbf{x}\right\|_\infty \leq \frac{\lambda}{2} \tag{4.22}$$

$$\left\|\mathbf{A}^T\mathbf{b}\right\| \leq \left\|\mathbf{A}^T\mathbf{A}\mathbf{x}\right\|_\infty + \frac{\lambda}{2} \tag{4.23}$$

$$\leq \left\|\mathbf{A}^T\mathbf{A}\right\|_\infty \|\mathbf{x}\|_1 + \frac{\lambda}{2} \tag{4.24}$$

$$= t\left\|\mathbf{A}^T\mathbf{A}\right\|_\infty + \frac{\lambda}{2} \tag{4.25}$$

onde a última igualdade se dá pois $\|\mathbf{x}\| = t$ sempre que $\mathbf{x}$ é a solução do problema para o parâmetro de regularização t. Concluímos agora que o primeiro limite inferior é

$$\lambda \geq 2(\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty - t\left\|\mathbf{A}^T\mathbf{A}\right\|_\infty) \tag{4.26}$$

O segundo limite inferior pode ser encontrado ao observar que, pela desigualdade triangular da norma,

$$t = \|\mathbf{x}\|_1 = \left\|\mathbf{x}_{ls} - \frac{\lambda}{2}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v}\right\|_1 \leq \|\mathbf{x}_{ls}\|_1 - \frac{\lambda}{2}\left\|\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{v}\right\|_1. \tag{4.27}$$

Agora, observando que $\|v\|_1 \leq n$ (pois cada $v_i \leq 1$), e que $\|(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{v}\|_1 \leq \|(\mathbf{A}^T\mathbf{A})^{-1}\|_1\|\mathbf{v}\|_1$, podemos obter o segundo limite inferior

$$\lambda \geq \frac{2(\|\mathbf{x}\|_1 - t)}{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1}. \tag{4.28}$$

Antes de prosseguir para encontrar o limite superior, podemos determinar para quais valores de t o limite inferior 1 (4.26) é maior que o limite inferior 2 (4.28), conforme mostrado abaixo

$$2(\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty - t\left\|\mathbf{A}^T\mathbf{A}\right\|_\infty) \geq \frac{2(\|\mathbf{x}\|_1 - t)}{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1} \tag{4.29}$$

$$t \leq \frac{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1\|\mathbf{A}\mathbf{b}\|_\infty - \|\mathbf{x}\|_1}{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1\|\mathbf{A}^T\mathbf{A}\|_\infty - 1}. \tag{4.30}$$

Assim, para os valores de t determinados pela desigualdade (4.30), devemos adotar o limite (4.26) e para os demais valores devemos adotar o limite (4.28).

Para obter o limite superior, iniciamos por isolar λ na equação (4.18), como abaixo.

$$\lambda = \frac{2}{t}\left[(\mathbf{A}^T\mathbf{b})^T\mathbf{x} - (\mathbf{A}^T\mathbf{A}\mathbf{x})^T\mathbf{x}\right] \tag{4.31}$$

$$\leq \frac{2}{t}\left[|(\mathbf{A}^T\mathbf{b})^T\mathbf{x}| + |(\mathbf{A}^T\mathbf{A}\mathbf{x})^T\mathbf{x}|\right] \tag{4.32}$$

Agora, usando (4.21), podemos escrever

$$\lambda \leq \frac{2}{t}\left(\|\mathbf{A}^T\mathbf{b}\|_2\|\mathbf{x}\|_2 + \|\mathbf{A}^T\mathbf{A}\mathbf{x}\|_2\|\mathbf{x}\|_2\right) \tag{4.33}$$

Aplicando a desigualdade entre as normas (4.20) e as desigualdade gerais (4.34) e (4.35) abaixo

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x},\| \qquad \mathbf{A} \in \mathbb{R}^{n\times m}, \quad \mathbf{x} \in \mathbb{R}^m, \tag{4.34}$$

$$\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\|\|\mathbf{B},\| \qquad \mathbf{A} \in \mathbb{R}^{n\times m}, \quad \mathbf{B} \in \mathbb{R}^{m\times p}, \tag{4.35}$$

chegamos a

$$\lambda \leq \frac{2}{t}\left(\|\mathbf{A}^T\|_2\|\mathbf{b}\|_2\|\mathbf{x}\|_1 + \|\mathbf{A}\|_2^2\|\mathbf{x}\|_1^2\right) \tag{4.36}$$

onde podemos substituir $\|\mathbf{x}\|_1 = t$, obtendo o limite superior

$$\lambda \leq 2\left(\|\mathbf{A}\|_2\|\mathbf{b}\|_2 + t\|\mathbf{A}\|_2^2\right). \tag{4.37}$$

Com as condições e limites obtidos acima, podemos finalmente introduzir um algoritmo para obtenção dos pontos da frente de Pareto do Lasso sem necessidade de minimização, por meio da busca dos subgradientes válidos para tais condições. O algoritmo descrito assume o parâmetro de regularização t, e deve ser executado para todos os valores de t para os quais se deseja uma aproximação de $\mathbf{x}^*(t)$. Também assume-se que os valores para t utilizados em execuções consecutivas crescem monotonamente, de forma que $t = t_{-1} + \alpha$, onde t_1 é o valor do parâmetro de regularização da execução anterior e $\alpha > 0$. Por fim, também é necessário que tenhamos acesso ao subgradiente adotado para o parâmetro de regularização t_{-1}, denotado como $\mathbf{v}_{-1}$.

Passo 1 Calcular os limites inferior (LI) e superior (LS) para o parâmetro t, conforme abaixo

$$LS = 2\left(\|\mathbf{A}\|_2\|\mathbf{b}\|_2 + t\|\mathbf{A}\|_2^2\right)$$
$$\text{Se } t \leq \frac{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1\|\mathbf{A}\mathbf{b}\|_\infty - \|\mathbf{x}\|_1}{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1\|\mathbf{A}^T\mathbf{A}\|_\infty - 1},$$
$$LI = 2(\left\|\mathbf{A}^T\mathbf{b}\right\|_\infty - t\left\|\mathbf{A}^T\mathbf{A}\right\|_\infty)$$
$$\text{Senão}$$
$$LI = \frac{2(\|\mathbf{x}\|_1 - t)}{n\|(\mathbf{A}^T\mathbf{A})^{-1}\|_1}.$$

Passo 2 Obter o subgradiente $\mathbf{v}$ da seguinte forma: Caso seja a primeira iteração, isto é, não existem valores de $\mathbf{x}^*(t)$ calculados anteriormente, escolher $\mathbf{v} \in [-1, 1]^n$ sob uma distribuição uniforme. Caso contrário, tome $\mathbf{v} \in B_\infty(\mathbf{v}_{-1}, h_k) \subset [-1, 1]^n$, onde $\mathbf{v}_{-1}$ é o subgradiente adotado na iteração anterior (para o parâmetro de regularização t_{-1} anterior), e h_k é o parâmetro que representa o raio da bola utilizada na obtenção de $\mathbf{v}$.

Antes de prosseguir com a explicação desse passo, definimos uma sub-iteração como um conjunto de repetições do Passo 2 seguidas por uma execução do passo Passo 6. Denotaremos o número da sub-iteração por k, a ser iniciado com o valor 1 na primeira sub-iteração. Em cada sub-iteração, adotamos

$$h_k = 2^{k-1}h_0,$$

onde h_0 é uma constante a ser escolhida antes da execução do algoritmo. O Passo 2 deve ser repetido no máximo M vezes e, após esse número, o algoritmo é desviado ao Passo 6.

A ideia nesse passo é iniciar a busca pelo subgradiente $\mathbf{v}$ a partir da último subgradiente encontrado, isto é, para o valor do parâmetro de regularização anterior, aqui denotado como t_{-1}. Dessa forma, aumentamos as chances de encontrar um subgradiente que satisfaça as condições para t, considerando que $t = t_{-1} + \alpha$, onde $\alpha > 0$ é um incremento relativamente pequeno se comparado à ordem de grandeza de t.

Passo 3 Verificar as condições sobre λ. Se $\lambda > UB$ ou $\lambda < LB$, v é descartado como candidato ao subgradiente e retornamos ao Passo 2. Caso contrário, seguimos ao Passo 4.

Passo 4 Verificar a condição de otimalidade em (4.12) para $\mathbf{v}$ e t. Caso tenhamos

$$t - \epsilon_k \leq \mathbf{v}^T \left[\mathbf{x}_{ls} - \frac{\lambda}{2} \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{v} \right] \leq t + \epsilon_k,$$

onde $\epsilon_k = 2^{k-1}\epsilon_0$, aceitamos

$$\mathbf{x} = \left[\mathbf{x}_{ls} - \frac{\lambda}{2} \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{v} \right]$$

como aproximação para $\mathbf{x}^*(t)$. Caso contrário, retornamos para o Passo 2. A constante ϵ_0 é definida à priori para a execução do algoritmo.

Passo 5 Remoção de pontos indesejados na frente de Pareto. Verificamos se o ponto obtido atende às condições de um ótimo de Pareto quando comparado aos demais pontos já calculados, pois podem haver flutuações devido à tolerância ϵ adotada. Para isso, verificamos se existe algum ponto $\mathbf{x}'$ tal que $f_1(\mathbf{x}') < f_1(\mathbf{x}')$ e $f_2(\mathbf{x}') < f_2(\mathbf{x}')$. Em caso positivo, descartamos o ponto encontrado e retornamos ao Passo 2.

Passo 6 Caso não seja encontrado nenhum ponto válido após M tentativas do Passo 2, deve-se então iniciar uma nova sub-iteração, incrementando o valor de k em uma unidade, e permitindo que o passo 2 seja executado por mais M vezes. Podemos definir uma condição de parada sobre o número de sub-iterações para limitar o tempo que será utilizado para aproximar $\mathbf{x}^*(t)$. Se $k > k_{\max}$ (pré-definido), então o algoritmo é encerrado para o parâmetro t.

Aqui cabe uma discussão sobre o encerramento do algoritmo. Caso a condição especificada no Passo 6 seja atingida, o algoritmo não obterá uma aproximação para $\mathbf{x}^*(t)$. No entanto, mesmo que não seja imposta tal condição, o algoritmo encontra uma solução pois a tolerância ϵ é ajustada a cada sub-iteração e torna-se mais flexível, ao custo da solução encontrada possuir um erro maior associado. Por esse motivo, em algumas situações é conveniente ter um limite para k.

5 RESULTADOS COMPUTACIONAIS

Neste capítulo são descritos os experimentos realizados e os resultados obtidos. Os testes têm o objetivo de determinar a performance dos algoritmos 1 e 2, além de avaliar os ganhos no tempo e perda de precisão no uso de diferentes métodos de interpolação a partir dos pontos obtidos pelos algoritmos 1 e 2.

Os testes estão divididos em três seções principais. A primeira compara a performance e a precisão do algoritmo 2 com os algoritmos RC, SLSQP e COBYLA. A segunda compara o tempo gasto pelo algoritmo 1 com o tempo gasto pelo respectivo algoritmo de otimização utilizado pelo próprio algoritmo 1. A terceira avalia a perda de precisão ao realizar a interpolação dos pontos calculados, por meio de diferentes métodos, visando reduzir o esforço computacional na geração da frente de Pareto.

Para implementação dos testes foi utilizada a linguagem Python, em conjunto com as bibliotecas Numpy, Scipy e Pandas, dentre outras bibliotecas auxiliares. Todos os testes foram executados utilizando um computador com sistema operacional Linux, processador Intel Core i5 10210U de frequência máxima de 4.20 GHz e 20GB de memória RAM (2666 MHz) disponível.

5.1 GERAÇÃO DOS DADOS

As constantes que definem o problema de regressão são $\mathbf{A} \in \mathbb{R}^{n \times m}$ e $\mathbf{b} \in \mathbb{R}^n$, que representam as variáveis de entrada e a respectiva resposta esperada.

Como esperamos que exista uma relação linear entre os preditores, ou pelo menos entre um subconjunto destes, e a variável de resposta, podemos gerar o conjunto de testes tomando aleatoriamente os parâmetros x_i na expressão da regressão mostrada em (2.1) e então, para cada amostra do conjunto tomamos aleatoriamente os valores das variáveis de entrada e calculamos a variável de resposta, com a adição de um erro aleatório normalmente distribuído.

Em todos os testes foram consideradas 3 variáveis de entrada e 100 amostras, para simplificar a análise dos diferentes algoritmos. Portanto, $\mathbf{A} \in \mathbb{R}^{3 \times 100}$ e $\mathbf{b} \in \mathbb{R}^{100}$ e o vetor de coeficientes $\mathbf{x} \in \mathbb{R}^3$.

A Figura 15 apresenta um exemplo da frente de Pareto gerada por cada um dos algoritmos, para um certo conjunto de dados.

5.2 COMPARAÇÃO ENTRE MÉTODOS DE OTIMIZAÇÃO

Nesta seção serão feitas comparações do erro cometido pelos algoritmos, bem como o tempo gasto por cada um deles. A métrica utilizada para o erro é a diferença de hipervolume entre as frentes de Pareto obtidas por meio de cada método.

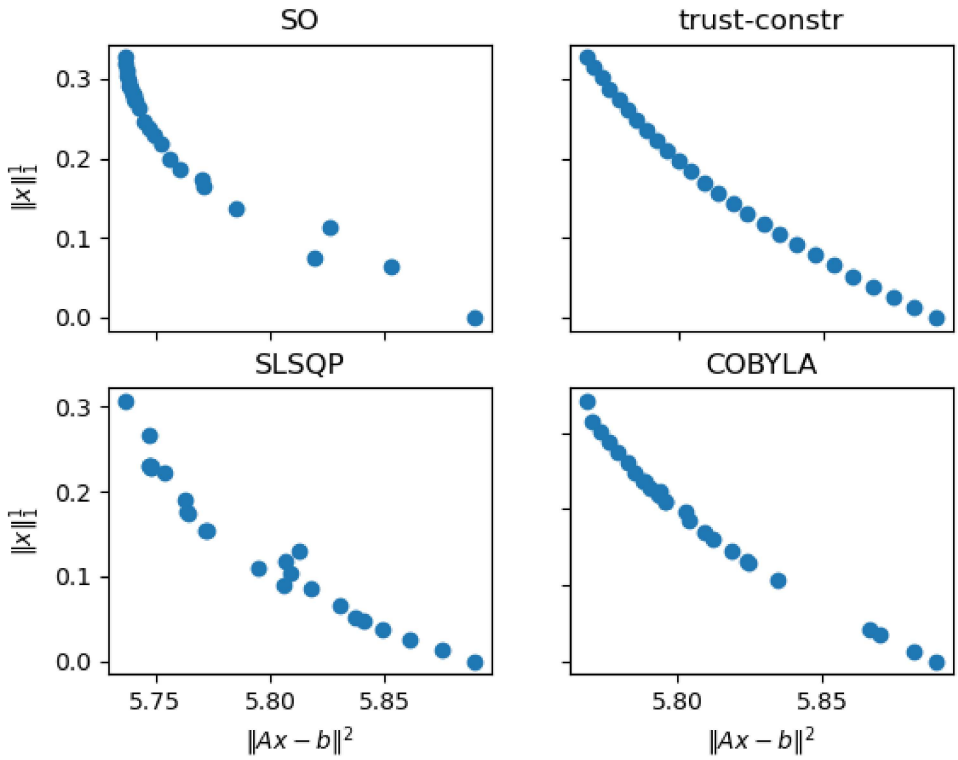

Figura 15 – Geração da frente de Pareto (sem remoção de pontos) para os diferentes métodos de otimização.

A distância de hipervolume é uma métrica comumente utilizada para comparação de frentes de Pareto, e consiste em calcular a área determinada pelas frentes de Pareto com relação a um ponto de referência, e então determinar a diferença entre as áreas obtidas. A fim de introduzir de forma intuitiva a métrica de hipervolume, a Figura 16 apresenta graficamente duas frentes de Pareto e seus respectivos hipervolumes.

Veja que o hipervolume é calculado com base em um ponto de referência $P = (p_1, p_2) \in \mathbb{R}^2$, o qual deve ser escolhido de forma que não exista qualquer ponto da frente de Pareto $x = (x_1, x_2)$ tal que $x_1 > p_1$ ou $x_2 > p_2$. Para o problema do Lasso, podemos assumir que $p_1 = \|\mathbf{b}\|_2^2$ e $p_2 = \|\mathbf{x}_{ls}\|_1$, pois todas as soluções do problema devem respeitar tais limites. Sendo assim, é possível obter a área da região R_1, delimitada pela frente de Pareto F_1 em vermelho, bem como a área da região R_2 delimitada pela frente F_2 em azul . Calculamos então a diferença entre as áreas das regiões, de forma a obter a métrica

$$H(F_1, F_2) = A(R_2) - A(R_1) \tag{5.1}$$

Veja que, se adotamos F_2 como a frente de Pareto ótima, $H(F_1, F_2)$ se torna uma métrica do erro cometido pela frente F_1. Nesse caso, cada uma das frentes geradas pelos algoritmos avaliados será colocado na condição de F_1 para obtenção do erro médio após 1000 repetições do teste. Além disso, usaremos também uma métrica de *erro relativo máximo*, como forma de determinar o erro máximo cometido dentre todos os pontos da

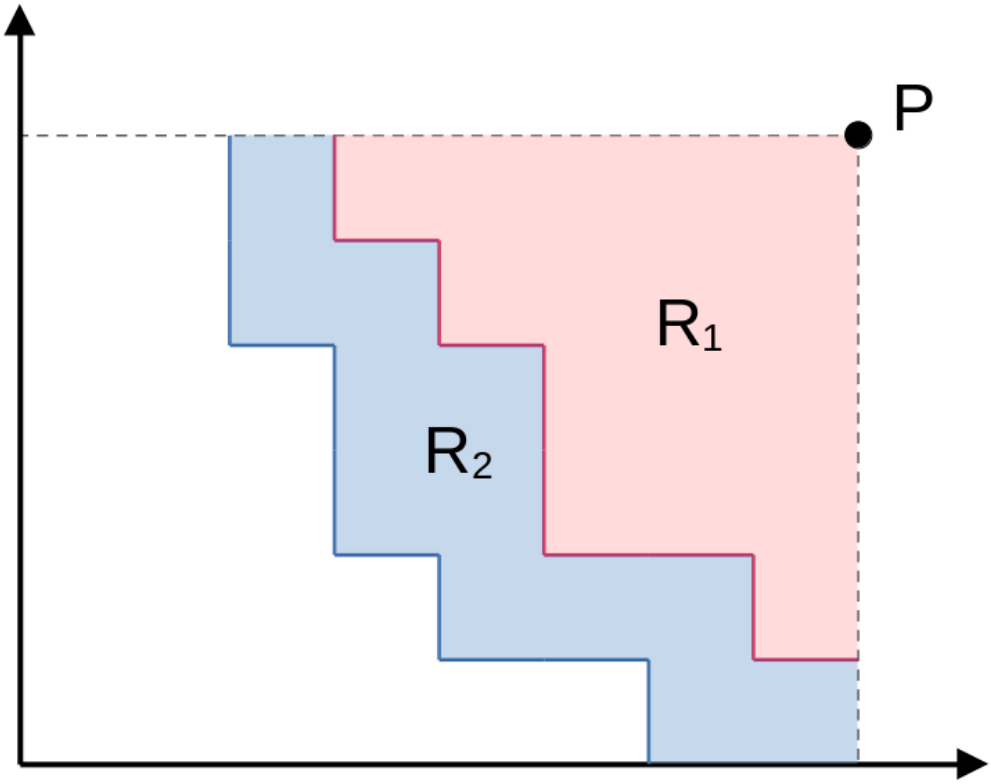

Figura 16 – Representação gráfica do cálculo do hipervolume para duas frentes de Pareto, considerando o ponto de referência P.

frente de Pareto, com relação à frente ótima. Dadas as frentes F_1 e F_2 contendo k pontos, definimos a métrica de erro relativo máximo como

$$M(F_1, F_2) = \max_{1 \leq i \leq k} |F_{2,i} - F_{1,i}|. \tag{5.2}$$

Para esta métrica, também é calculada a média após 1000 repetições do teste. A Tabela 1 apresenta os resultados para cada um dos algoritmos de geração da frente de Pareto.

×	SO	RC	SLSQP	COBYLA
Erro médio	0.0023	0.0163	0.0057	0.0074
Erro máximo	0.1031	0.5574	0.8599	0.5173

Tabela 1 – Média dos erros relativos e dos erros relativos máximos após 1000 repetições para cada um dos métodos de geração da frente de Pareto.

A seguir apresentamos a média ao final de 1000 repetições dos tempos de execução para cada um dos algoritmos apresentados, para os diferentes cenários de teste.

Cabe observar que o algoritmo 2 apresenta a melhor acurácia dentre todos os estudados, embora seja superado pelo algoritmos SLSQP e COBYLA quanto ao tempo de execução médio necessário para obtenção de todos os pontos da frente de Pareto.

×	SO	RC	SLSQP	COBYLA
Tempo médio	5.8837	23.0764	3.7831	0.0929

Tabela 2 – Média do tempo de execução após 1000 repetições para cada um dos métodos de geração da frente de Pareto.

5.3 OTIMIZAÇÃO DA APROXIMAÇÃO INICIAL

Nesta seção é feita a avaliação da eficiência obtida ao se empregar o Algoritmo 1, disposto na seção 4.2, com relação à utilização dos algoritmos de otimização RC, SLSQP e COBYLA de forma direta, isto é, sem a otimização da aproximação inicial x_0. Espera-se que os algoritmos encontrem as mesmas soluções que encontrariam sem a otimização de x_0, mas de forma mais rápida, visto que será necessário um número menor de iterações para a convergência dos mesmos.

Para esse teste, são feitas execuções empregando-se RC, SLSQP e COBYLA diretamente com os algoritmos de otimização utilizando como aproximação inicial $x_0 = \mathbf{0}$ para cada valor de t. Em seguida, comparamos o tempo necessário para que os algoritmos encontrem a solução utilizando a aproximação inicial otimizada x_0 descrita na Seção 4.2. Os resultados obtidos, apresentados na Tabela 3, são a média após 100 repetições do teste, com conjunto de dados distintos.

×	RC	SLSQP	COBYLA
Convencional	65.4993	10.1275	0.2568
Otimizado	60.2407	8.6578	0.2420
Ganho (%)	8.03	14.51	5.75

Tabela 3 – Tempos de execução médio com diferentes aproximações iniciais e ganho obtido.

Nota-se que o método que possui maior ganho com a utilização de uma solução inicial x_0 otimizada é o SLSQP. Para os demais métodos também existe ganho, embora sejam menos expressivos, sendo menores que 10%.

5.4 INTERPOLAÇÃO DE PONTOS

Para este teste, utilizaremos todos os métodos de interpolação apresentados na Seção 4.1, bem como todos os métodos para geração da frente de Pareto, apresentados nas seções 4.2 e 4.3. Primeiramente, apresentamos para um único problema as interpolações realizadas para as frentes de Pareto geradas por cada um dos algoritmos SO, RC, SLSQP e COBYLA, utilizando todos os métodos de interpolação citados na Seção 4.1. As Figuras de 17 a 32 apresentam os pontos gerados e a aproximação obtida para cada par de algoritmo de geração da Frente e método de interpolação.

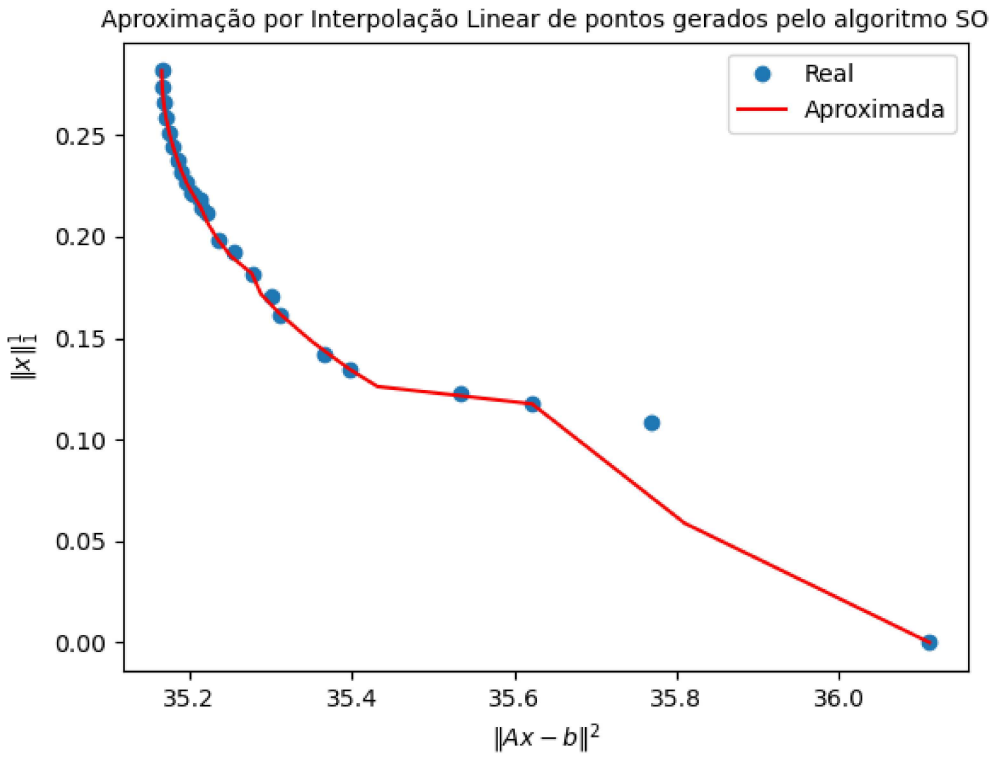

Figura 17 – Frente de Pareto gerada pelo algoritmo SO e interpolada pelo método Linear.

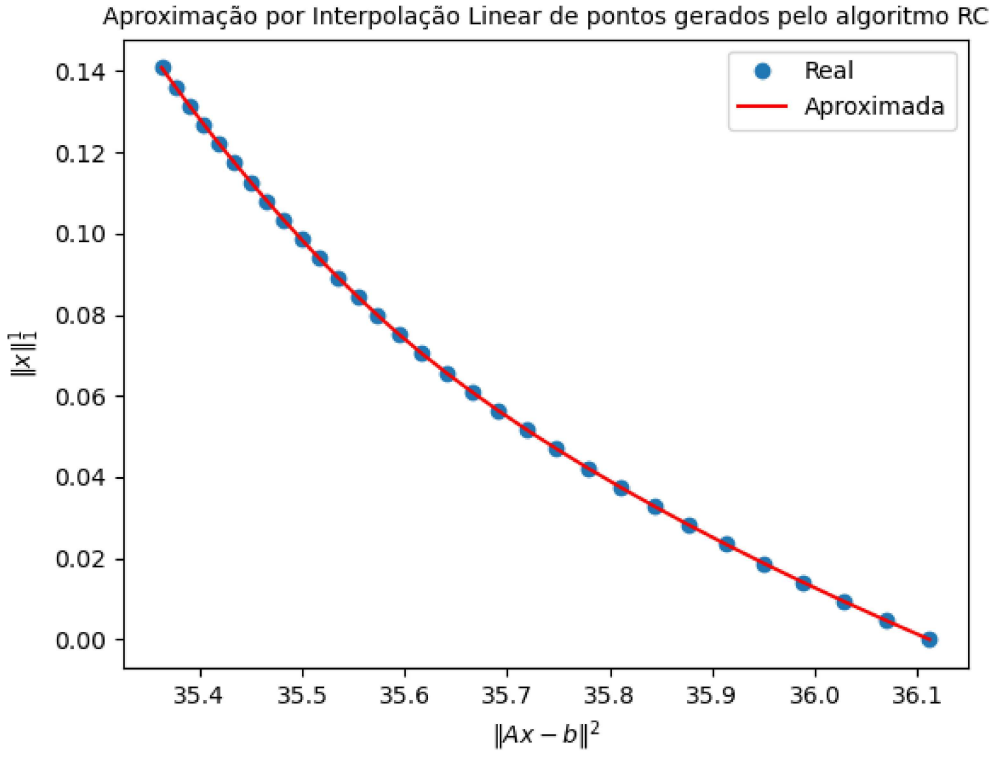

Figura 18 – Frente de Pareto gerada pelo algoritmo RC e interpolada pelo método Linear.

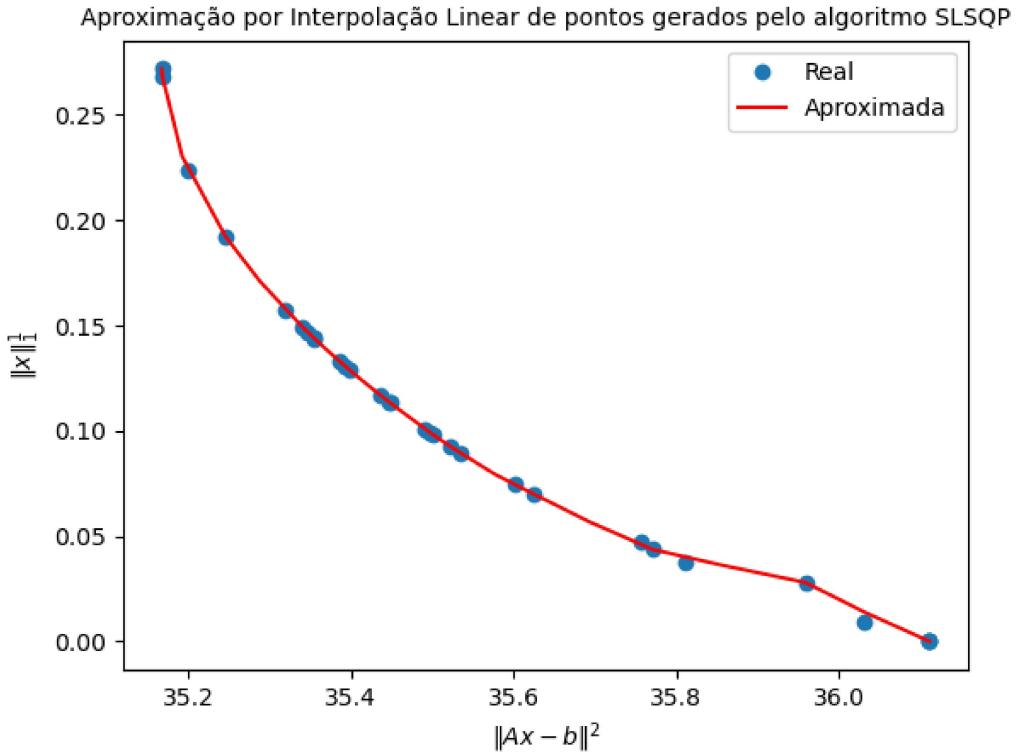

Figura 19 – Frente de Pareto gerada pelo algoritmo SLSQP e interpolada pelo método Linear.

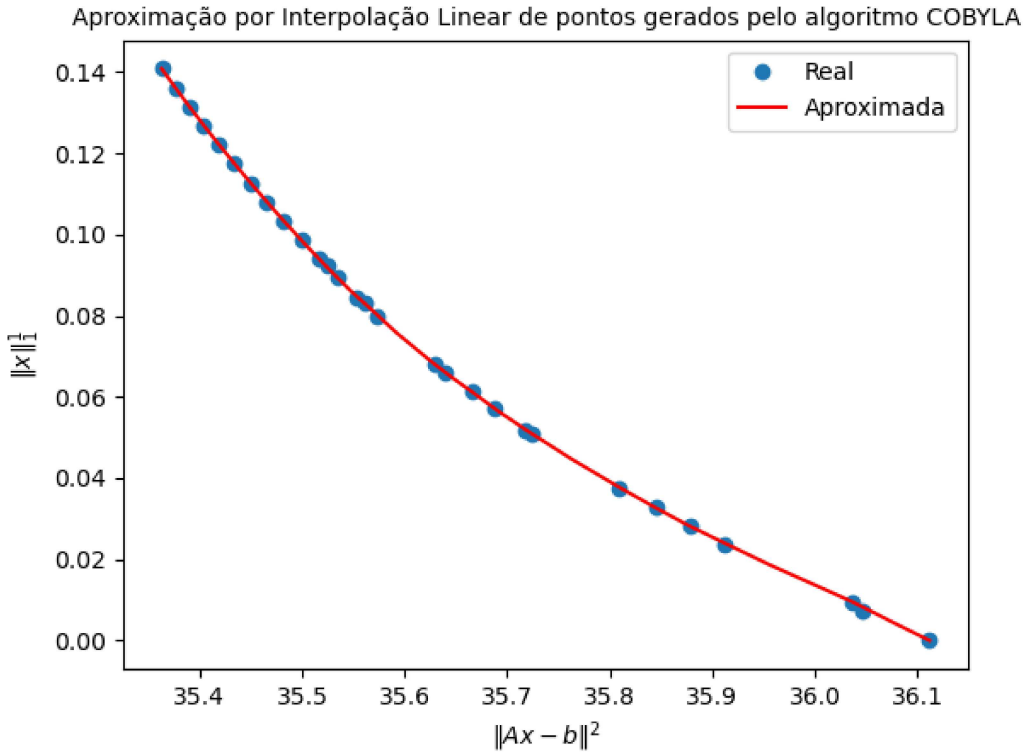

Figura 20 – Frente de Pareto gerada pelo algoritmo COBYLA e interpolada pelo método Linear.

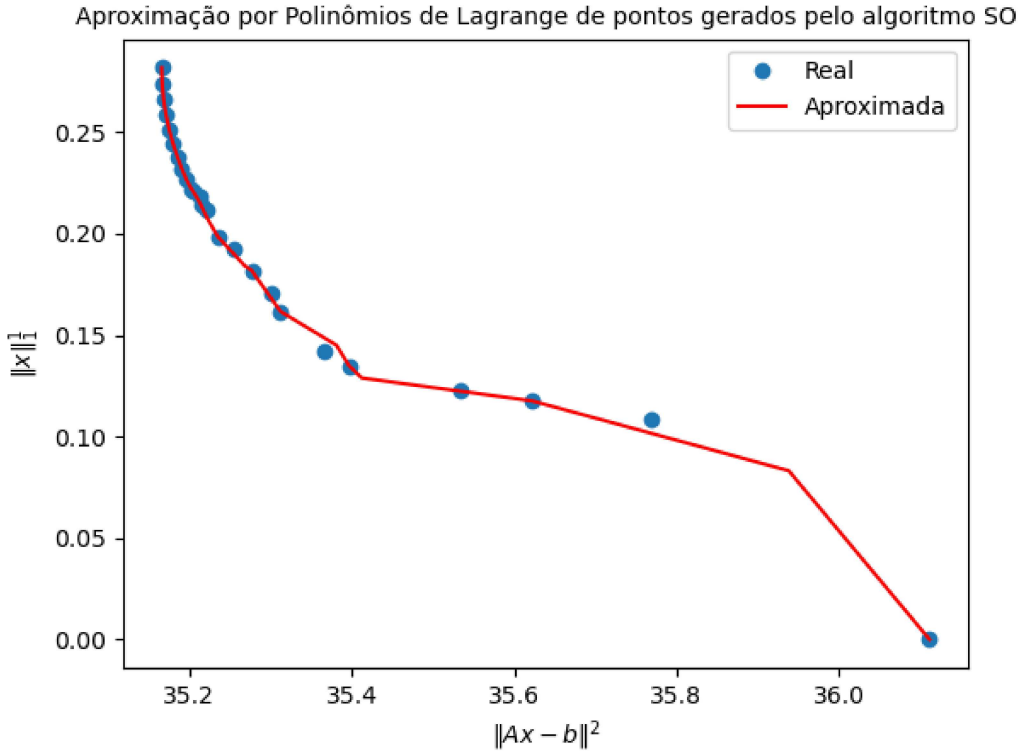

Figura 21 – Frente de Pareto gerada pelo algoritmo SO e interpolada pelo método Lagrange.

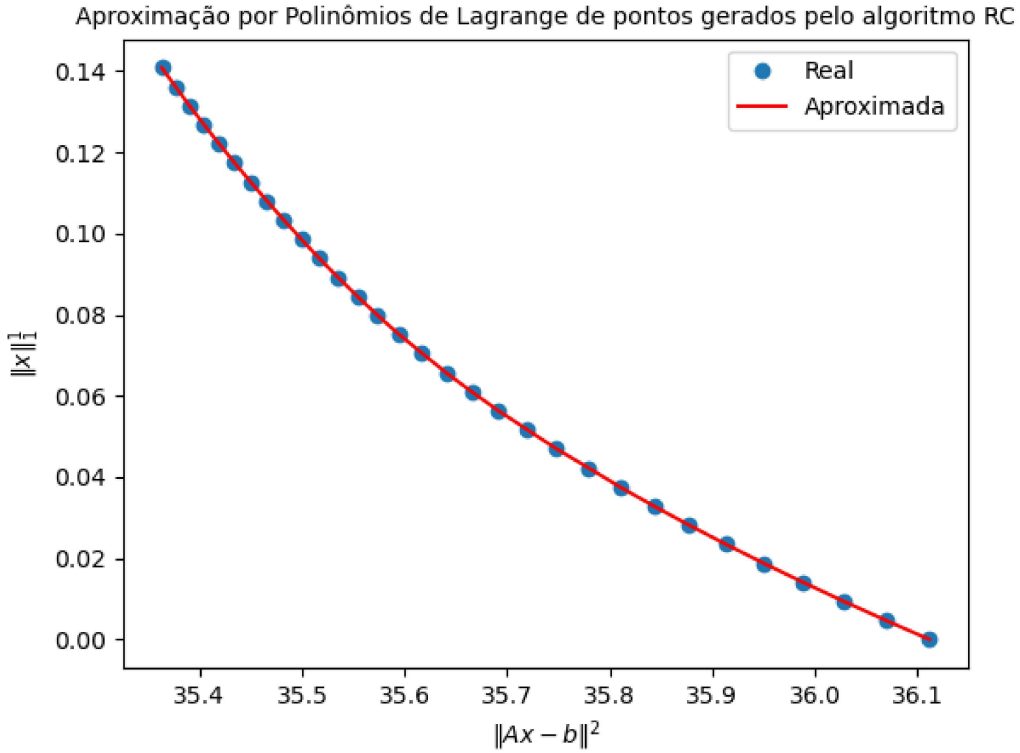

Figura 22 – Frente de Pareto gerada pelo algoritmo RC e interpolada pelo método Lagrange.

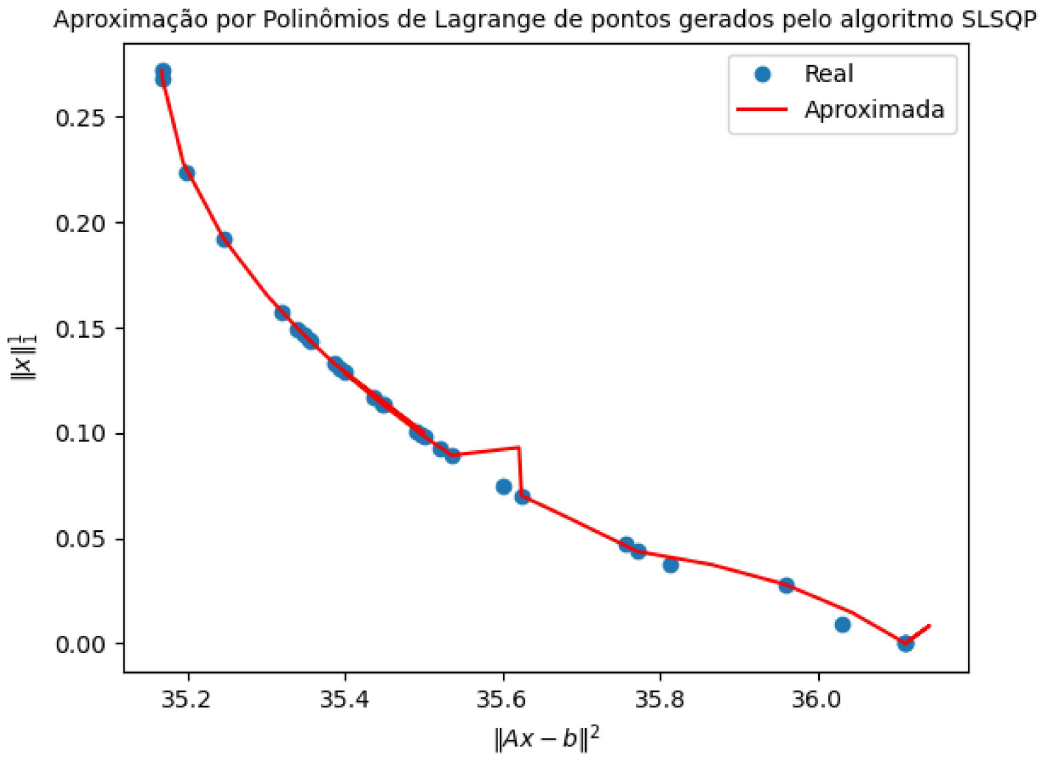

Figura 23 – Frente de Pareto gerada pelo algoritmo SLSQP e interpolada pelo método Lagrange.

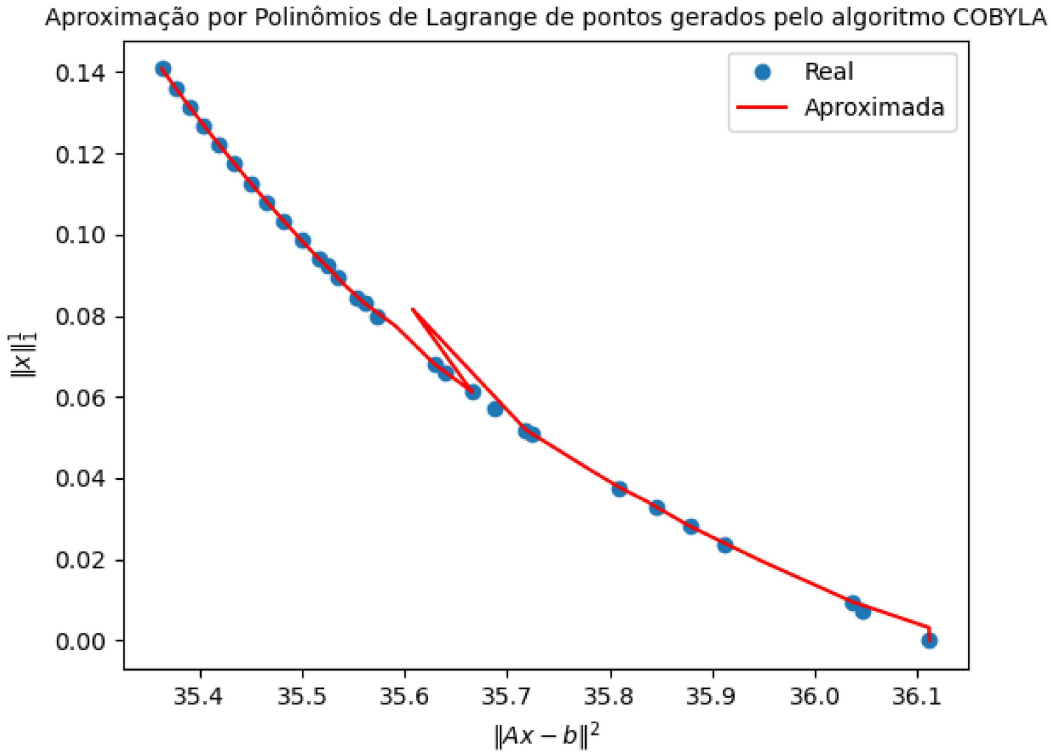

Figura 24 – Frente de Pareto gerada pelo algoritmo COBYLA e interpolada pelo método Lagrange.

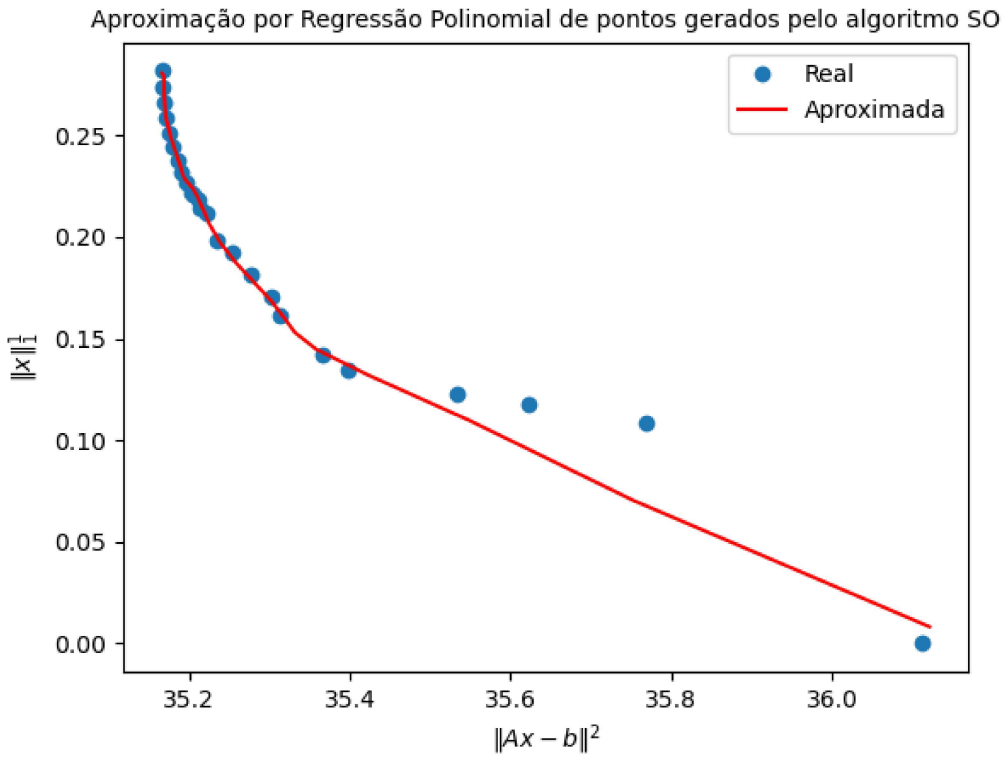

Figura 25 – Frente de Pareto gerada pelo algoritmo SO e interpolada pelo método Polinomial.

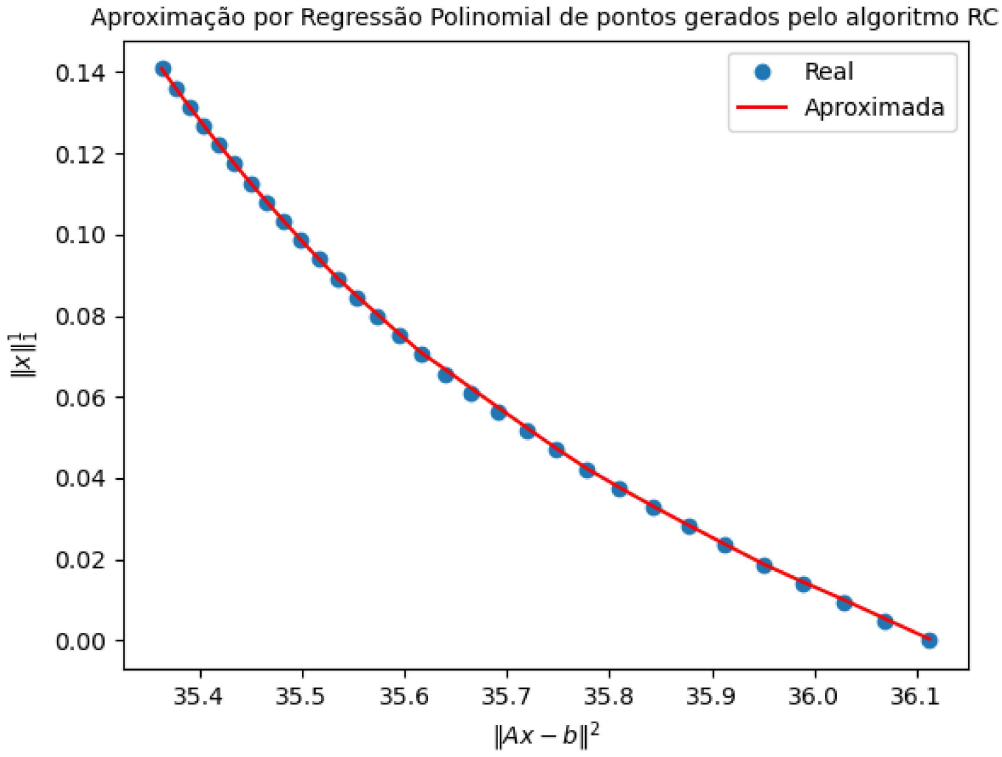

Figura 26 – Frente de Pareto gerada pelo algoritmo RC e interpolada pelo método Polinomial.

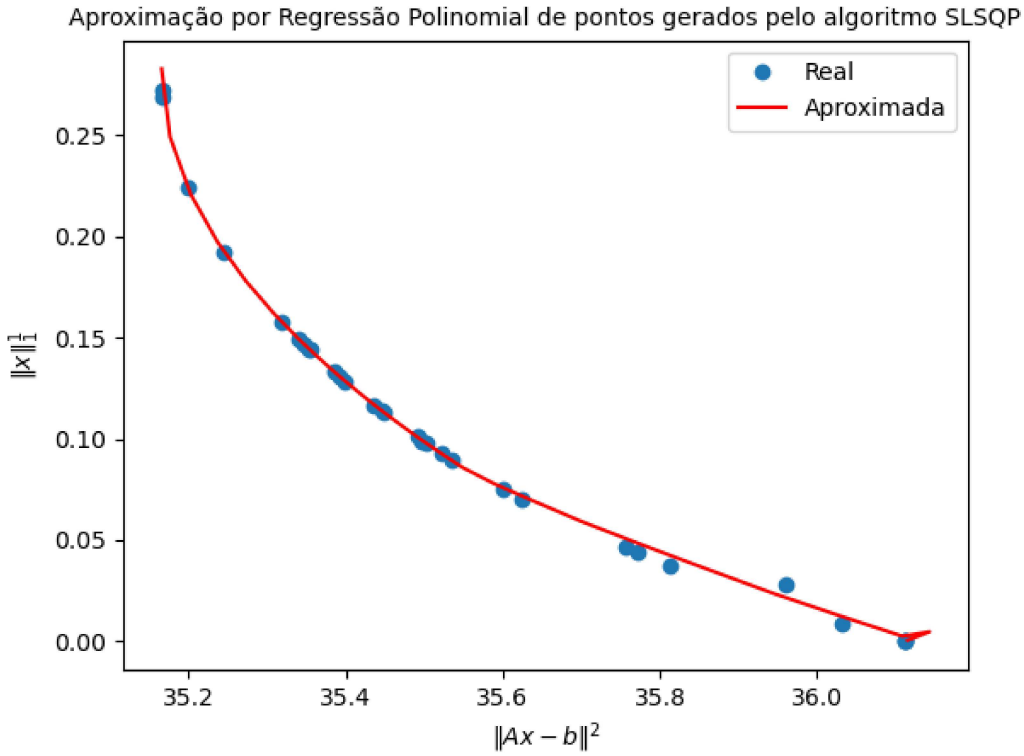

Figura 27 – Frente de Pareto gerada pelo algoritmo SLSQP e interpolada pelo método Polinomial.

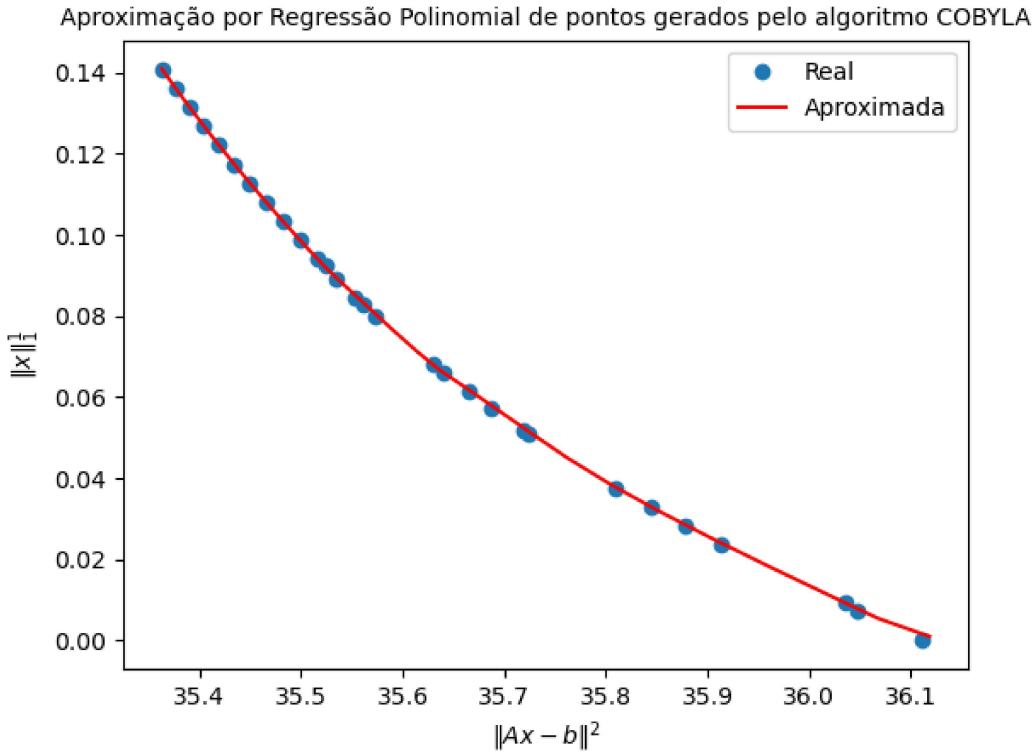

Figura 28 – Frente de Pareto gerada pelo algoritmo COBYLA e interpolada pelo método Polinomial.

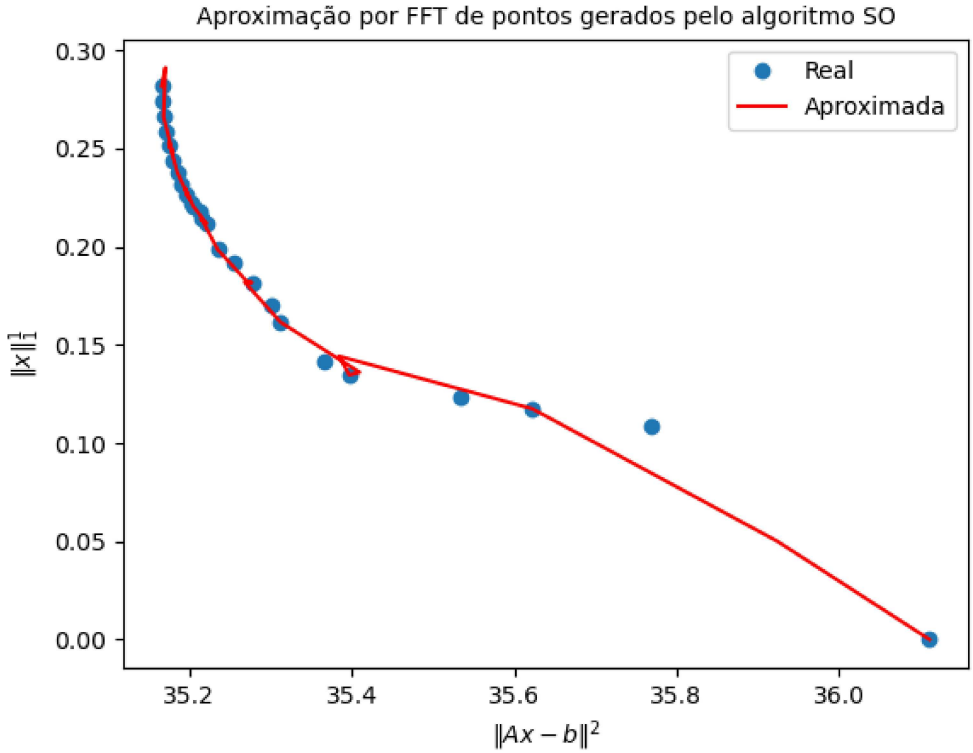

Figura 29 – Frente de Pareto gerada pelo algoritmo SO e interpolada pelo método FFT.

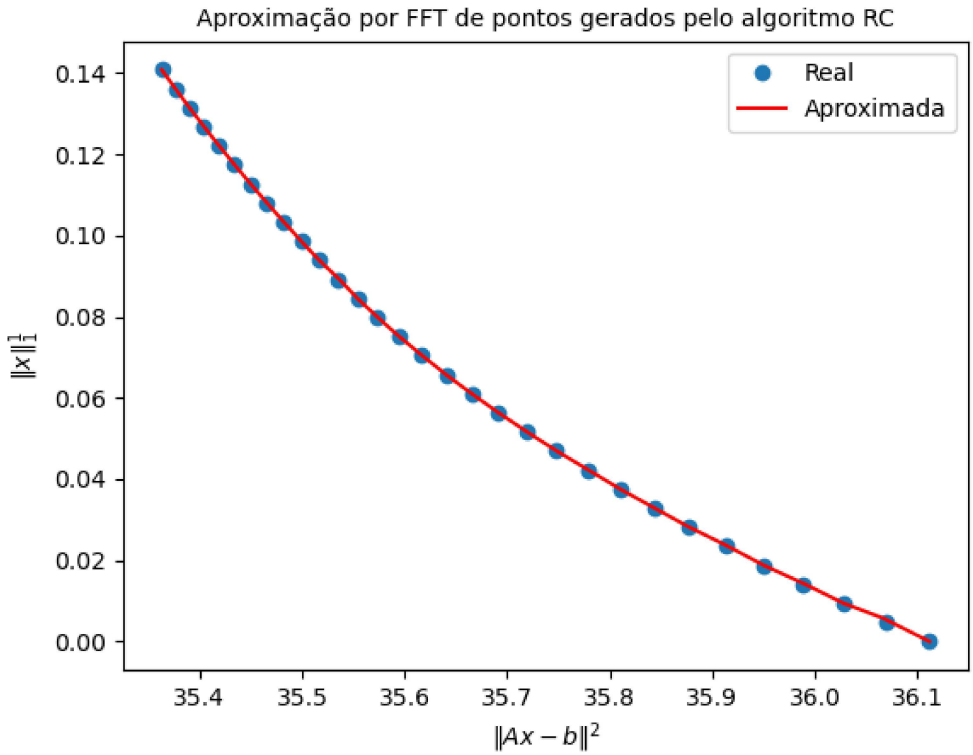

Figura 30 – Frente de Pareto gerada pelo algoritmo RC e interpolada pelo método FFT.

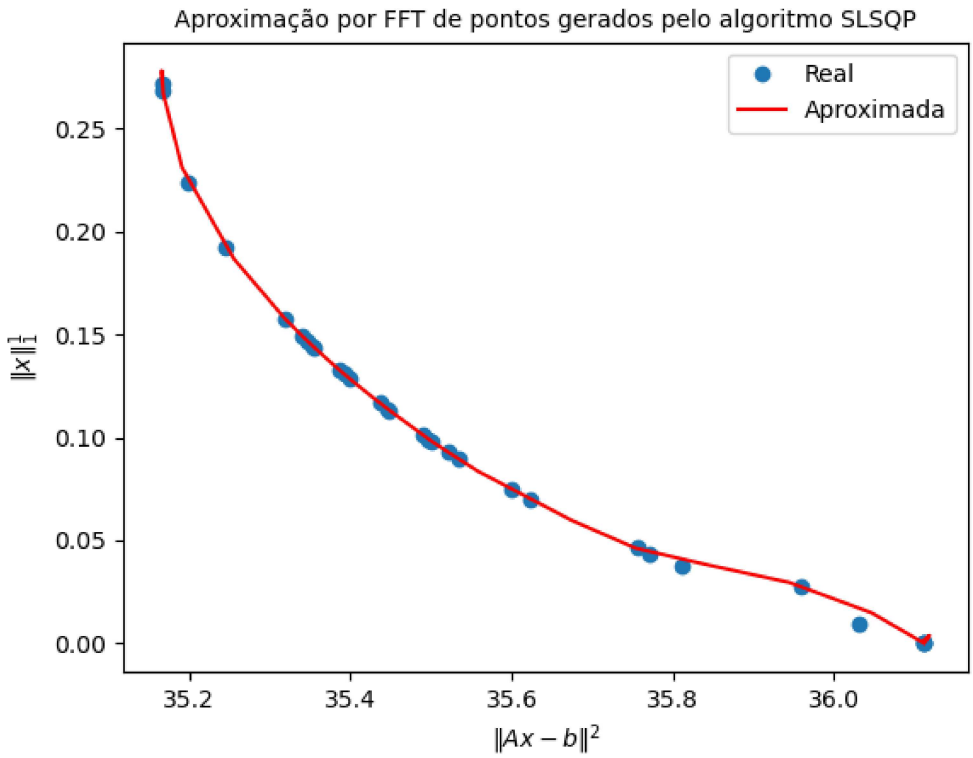

Figura 31 – Frente de Pareto gerada pelo algoritmo SLSQP e interpolada pelo método FFT.

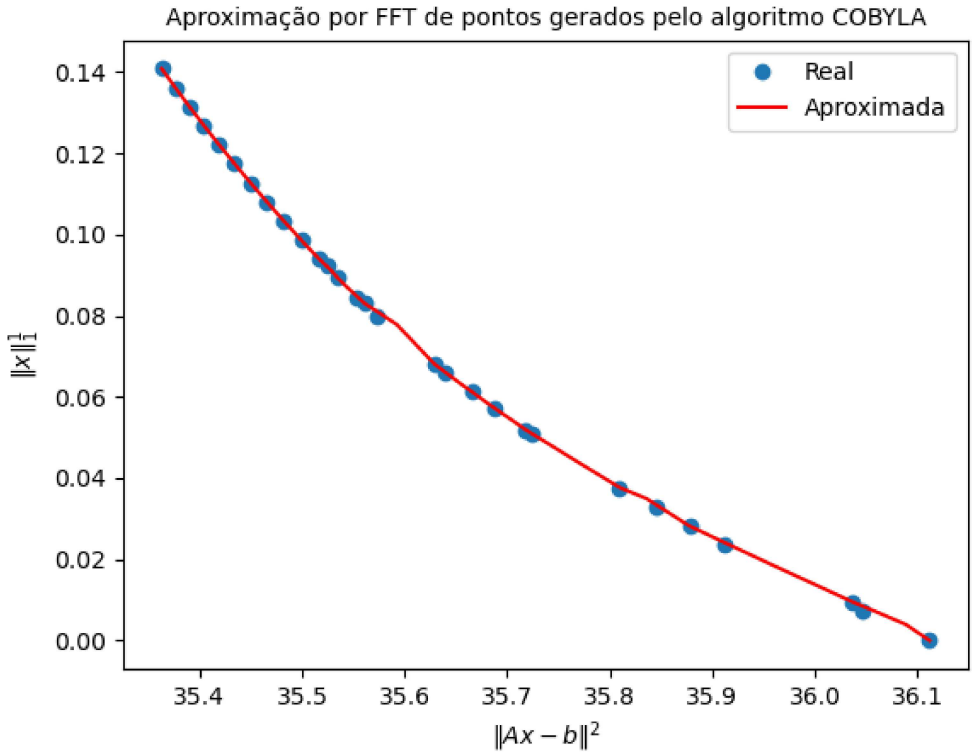

Figura 32 – Frente de Pareto gerada pelo algoritmo COBYLA e interpolada pelo método FFT.

A fim de estudar a perda de precisão ocorrida durante a interpolação, foram calculados 100 pontos da frente de Pareto e então foi utilizada apenas um subconjunto dos pontos calculados, uniformemente espaçados, para a interpolação. Após a interpolação é feita a comparação entre a curva gerada pela interpolação e a frente original utilizando a mesma distância de hipervolume apresentada na Seção 5.2, de forma a determinar o erro cometido.

Chamaremos de fator de remoção dos pontos a proporção $p \in (0, 1]$ dos pontos que será omitida do conjunto utilizado para a interpolação. Os valores utilizados no teste são 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 e 0.9. O teste será repetido para cada um dos algoritmos de geração dos pontos apresentados nas seções 4.2 e 4.3, sendo eles identificados por SO, RC, SLSQP e COBYLA, a fim de investigar como os métodos de interpolação se comportam com pontos gerados pelos diferentes algoritmos citados.

A Tabela 4 apresenta os erros para os fatores de remoção dos pontos, utilizando o método SO para geração dos pontos.

p	Linear	Lagrange	Polinomial	FFT
0.1	0.0002	0.0019	0.0094	0.0048
0.2	0.0007	0.0166	0.0122	0.0069
0.3	0.0014	0.0258	0.0117	0.0081
0.4	0.0017	0.0094	0.0097	0.0066
0.5	0.0041	0.0097	0.0123	0.0082
0.6	0.0042	0.0127	0.0103	0.0103
0.7	0.0057	0.0105	0.0109	0.0129
0.8	0.0095	0.0090	0.0102	0.0177
0.9	0.0218	0.0141	0.0135	0.0273

Tabela 4 – Erro médio entre todas as execuções para cada método de interpolação utilizando o algoritmo SO.

Os dados da Tabela 4 são apresentados na Figura 33.

Ao observar os dados apresentados na Tabela 4 e visualizados nas Figura 33, foi possível ver que o erro associado à interpolação com Polinômio de Lagrande é expressivamente maior quando comparado às demais interpolações para valores pequenos de p. Tal erro está relacionado ao chamado fenômeno de Runge mencionado na Seção 4.1.2, isto é, à oscilação causada pelo alto grau do polinômio resultante, visto que há um número elevado de pontos para interpolação.

Cabe observar aqui que todos os métodos apresentaram um aumento no erro para valores maiores de p, embora seja claro que, para o maior valor do parâmetro de remoção de pontos, o método de aproximação por Regressão Polinomial é aquele que possui o menor erro cometido dentre os demais métodos, superando inclusive a aproximação linear. Uma possível explicação para esse fato é que a Regressão Polinomial se ajusta melhor

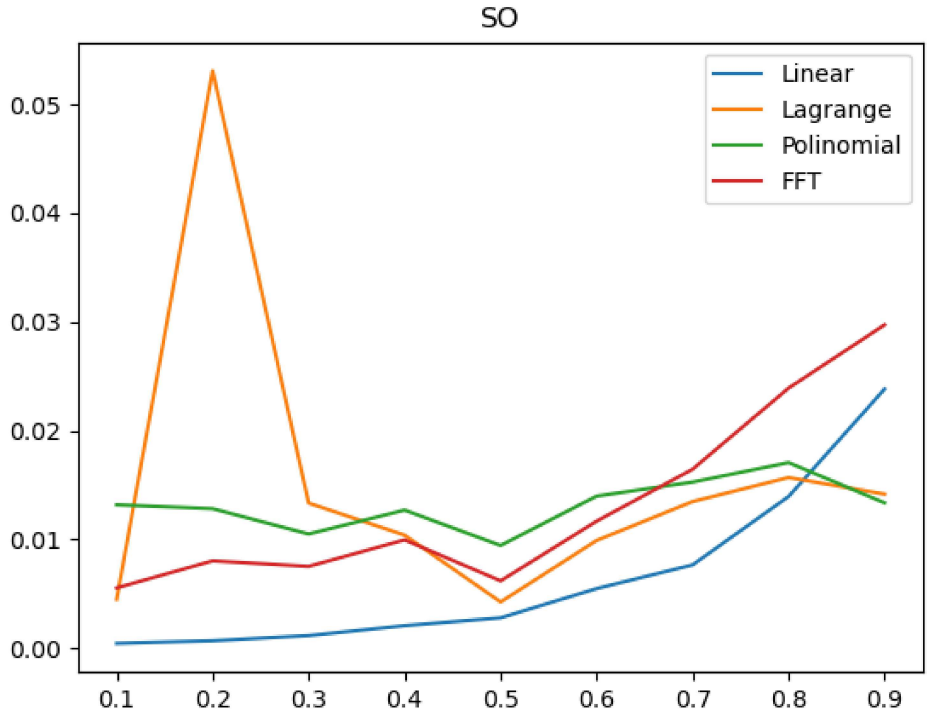

Figura 33 – Evolução do erro com o aumento de p para os diferentes métodos de interpolação utilizando o algoritmo SO.

nos pontos onde há uma mudança de direção na frente de regularização, ao contrário da aproximação linear. Além disso, a regressão polinomial não sofre com o fenômeno de Runge dos polinômios de Lagrange.

A Tabela 5 apresenta os erros para os fatores de remoção dos pontos, utilizando o método RC para geração dos pontos.

p	Linear	Lagrange	Polinomial	FFT
0.1	0.0001	0.0001	0.0002	0.0025
0.2	0.0001	0.0001	0.0008	0.0032
0.3	0.0001	0.0002	0.0004	0.0030
0.4	0.0001	0.0001	0.0002	0.0020
0.5	0.0001	0.0001	0.0003	0.0007
0.6	0.0001	0.0001	0.0002	0.0048
0.7	0.0002	0.0006	0.0017	0.0074
0.8	0.0001	0.0006	0.0002	0.0004
0.9	0.0005	0.0029	0.0014	0.0022

Tabela 5 – Erro médio entre todas as execuções para cada método de interpolação utilizando o algoritmo RC.

Os dados da Tabela 5 são apresentados na Figura 34.

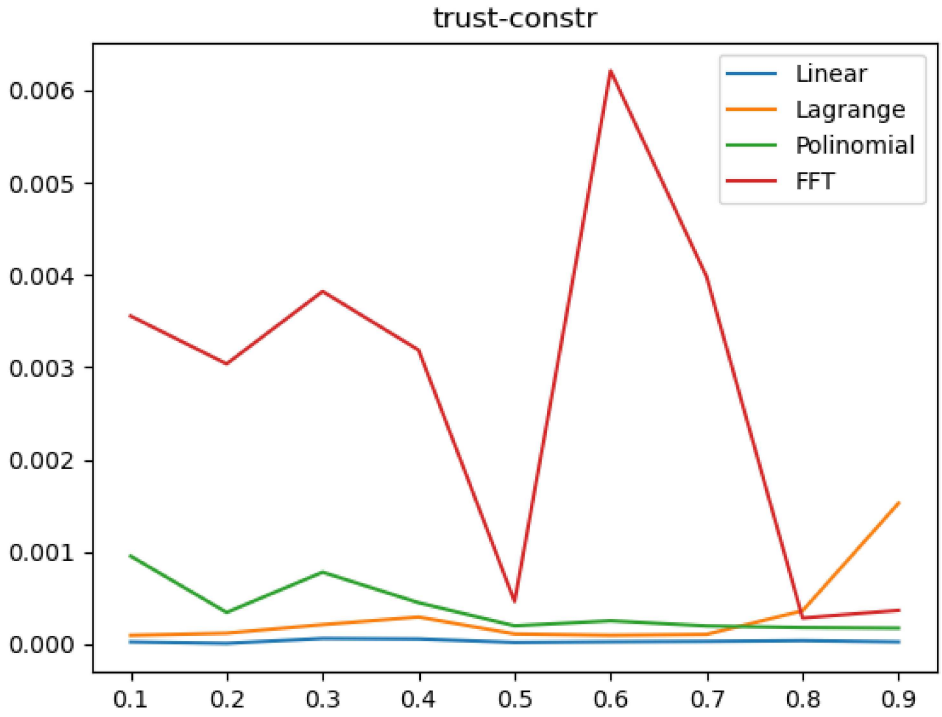

Figura 34 – Evolução do erro com o aumento de p para os diferentes métodos de interpolação utilizando o algoritmo RC.

Ao observar os dados apresentados na Tabela 5 e visualizados nas Figura 34, foi possível notar que o método de interpolação FFT possui o erro cometido muito maior que o dos demais métodos para a maioria dos valores do fator de remoção dos pontos. Isso se deve à natureza do método adaptado a partir do algoritmo FFT que interpola os pontos, que produz curvas que não se adéquam à natureza dos pontos gerados pelo método RC. Além disso, o erro se torna mais perceptível neste caso devido ao baixo erro cometido pelos demais métodos de interpolação.

A Tabela 6 apresenta os erros para os fatores de remoção dos pontos, utilizando o método SLSQP para geração dos pontos.

Os dados da Tabela 6 são apresentados na Figura 35.

Ao observar os dados apresentados na Tabela 6 e visualizados nas Figura 35, foi possível identificar qualitativamente os melhores métodos para interpolação a partir de pontos gerados pelo algoritmo SLSQP, sendo a interpolação Linear aquela que cometeu o menor erro para todos os fatores remoção dos pontos. Em seguida, as aproximações por Polinômio de Lagrange e por FFT apresentaram resultados parecidos, enquanto a aproximação por regressão polinomial se mostrou a pior que as demais em termos do erro cometido.

A Tabela 7 apresenta os erros para os fatores de remoção dos pontos, utilizando o

p	Linear	Lagrange	Polinomial	FFT
0.1	0.0004	0.0024	0.0159	0.0050
0.2	0.0014	0.0127	0.0202	0.0071
0.3	0.0015	0.0054	0.0142	0.0058
0.4	0.0016	0.0050	0.0129	0.0051
0.5	0.0026	0.0062	0.0210	0.0089
0.6	0.0039	0.0121	0.0150	0.0099
0.7	0.0047	0.0150	0.0145	0.0114
0.8	0.0077	0.0247	0.0256	0.0163
0.9	0.0130	0.0359	0.0192	0.0317

Tabela 6 – Erro médio entre todas as execuções para cada método de interpolação utilizando o algoritmo SLSQP.

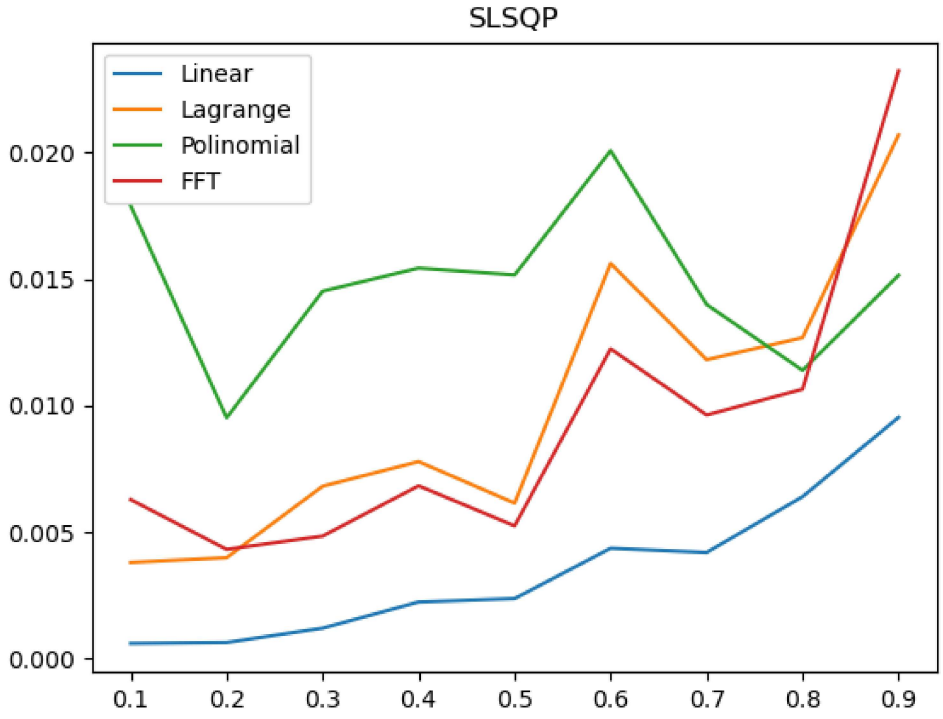

Figura 35 – Evolução do erro com o aumento de p para os diferentes métodos de interpolação utilizando o algoritmo SLSQP.

método COBYLA para geração dos pontos.

Os dados da Tabela 7 são apresentados na Figura 36.

Ao observar os dados apresentados na Tabela 7 e visualizados nas Figura 36, pudemos perceber que não existe grande diferença de performance entre os métodos de interpolação. No entanto, para todos os fatores de remoção dos pontos, a interpolação Linear se mostrou a mais eficiente, no sentido de provocar menor erro na aproximação.

O tempo necessário para obter a interpolação por meio de cada um dos métodos e

p	Linear	Lagrange	Polinomial	FFT
0.1	0.0002	0.0023	0.0070	0.0035
0.2	0.0008	0.0085	0.0088	0.0054
0.3	0.0008	0.0048	0.0059	0.0039
0.4	0.0015	0.0056	0.0083	0.0045
0.5	0.0009	0.0028	0.0042	0.0025
0.6	0.0021	0.0084	0.0070	0.0059
0.7	0.0032	0.0109	0.0073	0.0074
0.8	0.0056	0.0152	0.0108	0.0138
0.9	0.0120	0.0305	0.0114	0.0279

Tabela 7 – Erro médio entre todas as execuções para cada método de interpolação utilizando o algoritmo COBYLA.

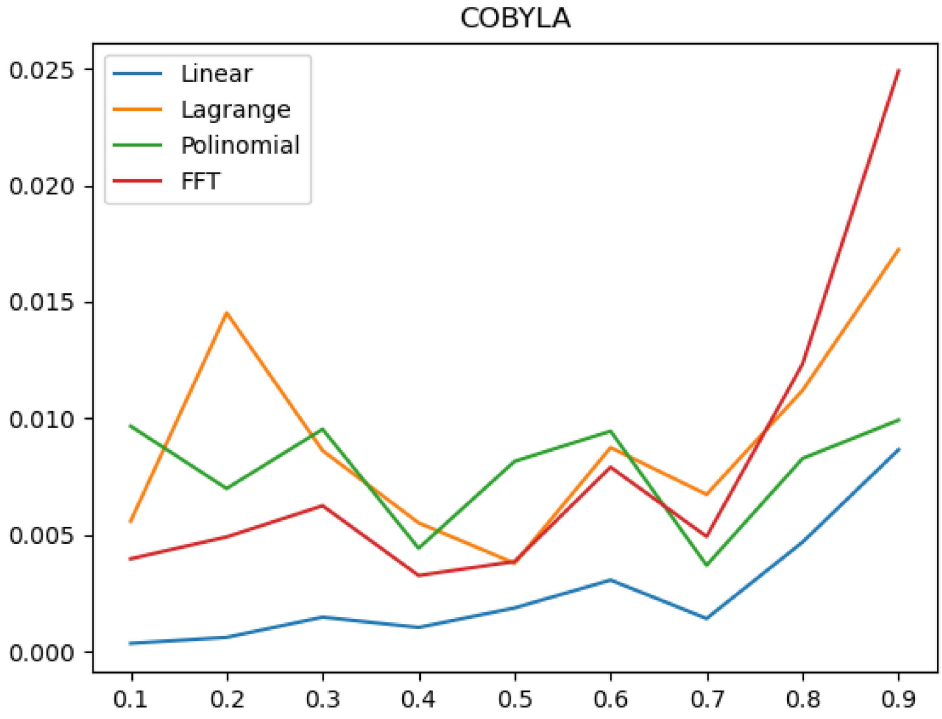

Figura 36 – Evolução do erro com o aumento de p para os diferentes métodos de interpolação utilizando o algoritmo COBYLA.

em cada um dos cenários é mostrado na Tabela 8.

Pôde-se observar que o tempo para cálculo da aproximação em todos os casos foi menor que o tempo necessário para a geração da frente de Pareto, principalmente quando utilizando o algoritmo sem otimização (SO) e o método das Regiões de Confiança (RC), como mostra a Tabela 2.

Esse resultado indica que, do ponto de vista do custo computacional, é vantajoso reduzir o número de pontos de Pareto gerados pelos algoritmos e realizar a interpolação

p	Linear	Lagrange	Polinomial	FFT
0.1	0.0024	1.0631	0.0032	0.0069
0.2	0.0017	0.6074	0.0020	0.0040
0.3	0.0015	0.4543	0.0020	0.0036
0.4	0.0012	0.2898	0.0018	0.0028
0.5	0.0011	0.1920	0.0018	0.0023
0.6	0.0009	0.1274	0.0017	0.0019
0.7	0.0008	0.0713	0.0017	0.0015
0.8	0.0006	0.0276	0.0016	0.0010
0.9	0.0004	0.0075	0.0014	0.0006

Tabela 8 – Tempo em segundos necessário para o cálculo da interpolação por meio de cada método.

destes. Naturalmente, a redução do número de pontos utilizados leva a um aumento do erro cometido na aproximação da frente. O real prejuízo causado por essa redução na precisão do algoritmo depende em grande parte da natureza da aplicação final dos modelos gerados, no sentido de que essa estratégia é funcional em aplicações onde é preferível a minimização do tempo de geração do modelo, mesmo que ao custo de alguma perda de precisão, por exemplo.

6 CONCLUSÃO

Durante os experimentos computacionais foi possível avaliar diferentes aspectos do processo de obtenção da frente de Pareto para o problema do Lasso por meio da utilização de algoritmos de otimização e por meio da utilização das propriedades do problema.

Primeiramente, foi possível validar o funcionamento do algoritmo SO, que não utiliza técnicas de minimização para obter aproximações para a frente de Pareto do Lasso, comparando-o com algoritmos que utilizam a otimização para alcançar o mesmo objetivo. Durante os teste de comparação, também verificou-se que existem algoritmos de otimização que executam em tempo menor do que o algoritmo SO, tal como COBYLA e SLSQP, apesar da menor precisão encontrada por estes últimos.

Em um segundo momento, foi possível verificar que o Algoritmo 1, proposto para melhorar a aproximação inicial para os algoritmos de otimização em cada valor do parâmetro de regularização, de fato proporcionou uma melhora nos tempos de execução de todos os algoritmos apresentados, sendo o resultado mais notável para o algoritmo SLSQP. Esse resultado é esperado devido à natureza da frente de regularização do Lasso, que, por ser linear por partes, proporciona uma forma de prever as coordenadas do próximo ponto com certo grau de precisão. Além disso, a taxa de melhora para cada algoritmo está relacionado ao funcionamento interno dos mesmos, e a fatores que incluem, por exemplo, a condição de parada adotada.

Por fim, durante os testes de interpolação foi possível observar a viabilidade em se usar modelos aproximados para geração da frente de Pareto. Foi possível ver que, mesmo com a utilização de apenas 80% dos pontos para geração da frente de Pareto, é possível obter um erro que corresponde a menos de 2% do valor das funções objetivo em cada ponto ao se utilizar o método de interpolação linear com pontos gerados pelo algoritmos de Região de Confiança (RC). Esse tipo de combinação pode ser usada para reduzir o custo computacional do processo de obtenção da frente de Pareto para o Lasso, principalmente observando que o método de RC é o mais custoso computacionalmente dentre os estudados.

Vale notar que algumas combinações entre algoritmo gerador dos pontos e método de interpolação dos pontos podem produzir resultados com erro considerável na aproximação, tornando os modelos interpolados pouco úteis, pois não representam adequadamente o conjunto de dados.

6.1 TRABALHOS FUTUROS

Durante a realização deste trabalho, foram identificados possíveis pontos de melhoria nos algoritmos e hipóteses que carecem de investigação mais profunda, que pretendemos trabalhar futuramente.

- Durante a busca pelo subgradiente (Passo 2 do algoritmo sem minimização), podem ser testadas novas formas de determinar a região de busca, adotando um raio diferente por coordenada como alternativa para a bola na norma infinito utilizada, por exemplo, ou alterando a função do decaimento do raio da bola a cada iteração.
- Verificar a viabilidade de descrever um algoritmo de seleção dos pontos ótimos, para que seja reduzido o número de iterações necessárias para construir a frente de Pareto. Tal algoritmo pode explorar as propriedades estatísticas do Lasso, bem como as propriedades geométricas da frente de regularização do problema multi-objetivo.
- Investigar a escalabilidade do algoritmo SO quando o número de variáveis de entrada aumenta.

REFERÊNCIAS

1 ARLOT, Sylvain; CELISSE, Alain. A survey of cross-validation procedures for model selection. 2010.

2 BELIAKOV, Gleb. Interpolation of Lipschitz functions. Journal of computational and applied mathematics, v. 196, n. 1, p. 20-44, 2006.

3 BONET-MONROIG, Xavier et al. Performance comparison of optimization methods on variational quantum algorithms. arXiv preprint arXiv:2111.13454, 2021.

4 CONN, Andrew R.; GOULD, Nicholas IM; TOINT, Philippe L. Trust region methods. Society for Industrial and Applied Mathematics, 2000.

5 DUTTA, Joydeep; KAYA, C. Yalçın. A new scalarization and numerical method for constructing the weak Pareto front of multi-objective optimization problems. Optimization, v. 60, n. 8-9, p. 1091-1104, 2011.

6 FAN, Guo-Xin; LIU, Qing Huo. Fast Fourier transform for discontinuous functions. IEEE Transactions on Antennas and Propagation, v. 52, n. 2, p. 461-465, 2004.

7 FAN, Jianqing; LI, Gang; LI, Runze. An overview on variable selection for survival analysis. Contemporary Multivariate Analysis And Design Of Experiments: In Celebration of Professor Kai-Tai Fang's 65th Birthday, p. 315-336, 2005.

8 HASTIE, Trevor et al. The elements of statistical learning: data mining, inference, and prediction. New York: springer, 2009.

9 KIM, Il Yong; DE WECK, Oliver L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Structural and multidisciplinary optimization, v. 29, p. 149-158, 2005.

10 KIM, Il Yong; DE WECK, O. L. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. Structural and multidisciplinary optimization, v. 31, n. 2, p. 105-116, 2006.

11 LEE, Jason D. et al. Exact post-selection inference, with application to the lasso. 2016.

12 LIU, Keli; MARKOVIC, Jelena; TIBSHIRANI, Robert. More powerful post-selection inference, with application to the lasso. arXiv preprint arXiv:1801.09037, 2018.

13 MARTÍNEZ, J. M.; SANTOS, S. A. Métodos computacionais de otimizaçao. 20 Colóquio Brasileiro de Matemática-IMPA. 1955.

14 MEINSHAUSEN, Nicolai. Relaxed lasso. Computational Statistics & Data Analysis, v. 52, n. 1, p. 374-393, 2007.

15 MIETTINEN, Kaisa. Nonlinear multiobjective optimization. Springer Science & Business Media, 1999.

16 MOEINI, Mahdi. Solving the index tracking problem: a continuous optimization approach. Central European Journal of Operations Research, v. 30, n. 2, p. 807-835, 2022.

17 TIBSHIRANI, Robert. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), v. 58, n. 1, p. 267-288, 1996.

18 VARSEI, Mohsen et al. A heuristic approach to the index tracking problem: a case study of the tehran exchange price index. Asian academy of management Journal, v. 18, n. 1, p. 19, 2013.

19 WALDRON, Levi et al. Optimized application of penalized regression methods to diverse genomic data. Bioinformatics, v. 27, n. 24, p. 3399-3406, 2011.

20 WU, Lan; YANG, Yuehan; LIU, Hanzhong. Nonnegative-lasso and application in index tracking. Computational Statistics & Data Analysis, v. 70, p. 116-126, 2014.

21 XIE, Jupeng et al. Energy consumption optimization of central air-conditioning based on sequential-least-square-programming. In: 2020 Chinese Control And Decision Conference (CCDC). IEEE, 2020. p. 5147-5152.