

UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MATHEUS MACHADO VIEIRA

UMA ANÁLISE QUALITATIVA DO SEQUENTIAL CUTTING PLANE
-
DA ORIGEM ÀS INFLUÊNCIAS MODERNAS

GARANHUNS
2021

MATHEUS MACHADO VIEIRA

**UMA ANÁLISE QUALITATIVA DO SEQUENTIAL CUTTING PLANE
-
DA ORIGEM ÀS INFLUÊNCIAS MODERNAS**

Monografia de Trabalho de Conclusão de Curso de graduação, submetido à Universidade Federal do Agreste de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, sob orientação do professor Dr^o Gersonilo Oliveira da Silva.

GARANHUNS

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

V658a Vieira, Matheus Machado
 Uma Análise Qualitativa do Sequential Cutting Plane: Da Origem às Influências Modernas / Matheus Machado
Vieira. - 2021.
 81 f. : il.

 Orientador: Gersonilo Oliveira da Silva.
 Inclui referências.

 Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Ciência da Computação, Garanhuns, 2021.

 1. Otimização contínua. 2. Problemas não lineares. 3. Programação. I. Silva, Gersonilo Oliveira da, orient. II. Título

CDD 004

MATHEUS MACHADO VIEIRA

UMA ANÁLISE QUALITATIVA DO SEQUENTIAL CUTTING PLANE
-
DA ORIGEM ÀS INFLUÊNCIAS MODERNAS

Monografia de Trabalho de Conclusão de Curso de graduação, submetido à Universidade Federal do Agreste de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, sob orientação do professor Dr^o Gersonilo Oliveira da Silva.

Garanhuns, 21 de Junho de 2021

Orientador: **Prof. Dr. Gersonilo Oliveira da Silva**

Universidade Federal do Agreste de Pernambuco

Prof. Dr. Igor Medeiros Vanderlei

Universidade Federal do Agreste de Pernambuco

Profa. Dra. Maria Aparecida Amorim Sibaldo

Universidade Federal do Agreste de Pernambuco

Resumo

O método SCP pode ser posto em uma condição histórica renascentista, uma vez que ele se propõe a usar métodos lineares que há muito foram deixados de lado. Mesmo usando algo que por vezes foi considerado uma tecnologia defasada, o método foi capaz de atingir bons resultados. Esse trabalho tem como foco o estudo dos fundamentos, funcionamento e implicações do método em um contexto contemporâneo. Um produto desse estudo foi a confecção de um código que tenta equivaler-se ao método proposto, que data de 2010. Tanto o estudo quanto a confecção do código resultaram em um ferramental teórico e prático a respeito não só do SCP, mas também à outros métodos, como o do gradiente conjugado e o método sequencial quadrático. Análises sobre o funcionamento do método e a conjuntura das ferramentas usadas em sua construção foram feitas a fim de compreender não só as possibilidades advindas do método, mas também seu percurso e o que ele significa na história da otimização.

Palavras-chave: Otimização contínua. Problemas não lineares. Programação.

Abstract

The SCP method can be placed in a renaissance historical condition, since it purports to use linear methods that have long been discarded. Even using something that was sometimes considered outdated technology, the method was able to achieve good results. This work focuses on the study of the fundamentals, functioning and implications of the method in a contemporary context. The product of this study was the making of a code that tries to match the proposed method, which dates from 2010. Both the study and the making of the code resulted in theoretical and practical tools regarding not only the SCP, but also other methods, such as conjugate as the gradient and the sequential quadratic method. Analyzes of the method's functioning and the conjunction of the tools used in its construction were carried out in order to understand not only the possibilities arising from the method, but also its path and what it means in the history of optimization.

Keywords: Continuous optimization. Nonlinear Problems. Programming.

LISTA DE FIGURAS

Figura 1 – Função de Bohachevsky com ótimo em $(0, 0)$	67
Figura 2 – Função Perm com ótimo em $(1, 0.5)$	67
Figura 3 – Função Trid com ótimo em $(2, 2)$	67
Figura 4 – Função Sum Squares com ótimo em $(0, 0)$	68

LISTA DE TABELAS

Tabela 1 – Funções de teste irrestritas.	67
Tabela 2 – Tempo de execução de diversas etapas.	70

LISTA DE ABREVIATURAS E SIGLAS

ARC	Atualização da Região de Confiança
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CPU	Central Processing Unit
DFP	Davidon–Fletcher–Powell
EMFCQ	Extended Mangasarian-Fromovitz Constraint Qualification
KKT	Karush-Kuhn-Tucker
KKTE	KKT Estacionário
MINLP	Mixed Integer Nonlinear Programming
SCP	Sequential Cutting Plane, Sequential Convex Programming
SPL	Sequência de Problemas Lineares
SQP	Sequential Quadratic Programming
VFM	Verificação da Função de Mérito

LISTA DE ALGORITMOS

Algoritmo 1 – SCP - <i>Sequential Cutting Plane</i>	56
Algoritmo 2 – SPL - <i>Sequência de Problemas Lineares</i>	59
Algoritmo 3 – VFM - <i>Verificação da Função de Mérito</i>	64
Algoritmo 4 – ARC - <i>Atualização da Região de Confiança</i>	65

SUMÁRIO

1 – Introdução	12
2 – Revisão Literária	14
2.1 A programação sob o viés matemático	14
2.1.1 Funções à Uma Variável Real	14
2.1.2 Funções à Várias Variáveis Reais	19
2.2 As primeiras três décadas da programação	24
2.2.1 L. V. Kantorovich	24
2.2.2 G. B. Dantzig	26
2.3 Programação não linear - Um possível início?	28
2.3.1 J. E. Kelley Jr.	28
2.3.2 R. E. Griffith, R. A. Stewart	29
2.3.3 S. P. Han	29
2.3.4 Richard H. Byrd, et al.	30
2.4 Os pilares arquitetônicos do SCP	30
2.4.1 Ferramentas usadas pelo SCP	31
2.4.1.1 Condições de Karush-Kuhn-Tucker	31
2.4.1.2 Qualificações de Mangasarian e Fromovitz	32
2.4.2 Sobre a escolha do uso de problemas lineares	33
2.4.2.1 LANCELOT	34
2.4.2.2 DONLP2	35
2.5 Método do gradiente conjugado	35
2.5.1 Método do gradiente descendente	36
2.5.2 Construindo o método do gradiente conjugado	37
2.6 Problemas lineares duais	41
2.7 Cálculos numéricos da Hessiana	43
2.7.1 Diferença finita	44
2.7.2 DFP e BFGS	45
2.7.3 Passo complexo	47
2.7.4 Números hiper-duais	47
3 – Percurso Metodológico	49
3.1 Delineamento da Pesquisa	50
3.2 Coleta e Tratamento de Dados	51
4 – Análise e discussão dos resultados	52

4.1	O modernismo sob o ponto de vista da programação	52
4.2	Introdução ao SCP	54
4.2.1	Aplicação SCP em MINLP	54
4.3	Análise Qualitativa do algoritmo - SCP	55
4.3.1	O Algoritmo	56
4.3.1.1	Subiterações lineares	57
4.3.1.2	Função Mérito	62
4.3.1.3	Regiões de confiança	64
4.4	Análise Quantitativa do algoritmo - SCP	65
4.4.1	Problemas de Surjanovic e Bingham	66
4.4.2	Problemas de Hock e Schittkowski	69
4.4.3	Avaliação numérica do SCP	70
4.5	As vantagens e desvantagens computacionais do SCP	71
5	– Conclusão	74
5.1	Trabalhos Futuros	74
5.2	Considerações Finais	75
	Referências	76

1 Introdução

Os processos industriais potencializados no início do século XX guiado pelos fronts das duas grandes guerras trouxe um solavanco ao desejo de solução de problemas de carácter otimizável. Dois personagens em cenários distintos se sobressaem dando partida a corrida armamentista dos métodos de otimização.

Por outro lado, de um ponto de vista teórico os métodos de otimização nascem junto com o Cálculo Diferencial advindo do Principia (Newton) e reformulado por seus colaboradores, Euler, Cauchy, Weierstrass, Lagrange, entre outros. O mecanismo em si ainda precisava de um interlocutor, o computador, a máquina que decifraria os enigmas da fenomenologia. Com a evolução dos sistemas processuais booleanos advieram a cepa de uma nova era, o modernismo pragmático das máquinas digitais possibilitaram um novo horizonte a ser concebido pelas mentes mais audazes e, convenientemente, mais bem preparadas de toda a história humana, os cientistas do meio do século XX.

Apenas depois de quase 30 anos de avanços teóricos e práticos é que pudemos contar com métodos que se equivaleriam ao clássico método de Newton no que se refere a solução de um problema de busca linear. O efeito desse avanço nos colocou em um movimento que só fez acelerar. Desde o combalido método Simplex ([DANTZIG, 1963](#)), enviesando entre os métodos de Newton e seus correlacionados, estamos em uma busca contínua por um modo de encontrar ótimos de funções de formas cada vez mais eficientes. Por essa busca e pelo sucesso dela é que temos hoje computadores funcionais em dispositivos móveis e uma janela de aplicabilidades dos modelos de otimização evoluindo dia após dia.

É nesse cenário que viemos apresentar um método que foi gerado para ser aplicado a um espectro grande de modelos pela amplitude de seus parâmetros. Visando uma aplicação à problemas não lineares e em que as funções que representam o modelo não sejam necessariamente conexas, o *Sequential Cutting Plane* (SCP) se propõe como uma ferramenta robusta e metamorfa. Pois o algoritmo no qual está erguido faz uso de um modo sofisticado dos elementos que compõem todos os métodos clássicos, e mais, ele se supõe ser ainda mais eficaz que cada um deles.

Entendo que só se poderia apresentar os aspectos técnicos do SCP mediante uma abreviada introdução ao teor histórico que fomentou as ideias por trás do ideal do algoritmo e, que também, seria necessário apresentar estruturas fundamentais à otimização matemática, em seu próprio território, dentre outras estruturas que se aliciam à técnica, seja pelo suporte ao mecanismo teórico, seja pela praticidade de linguagem. Apresentamos no Capítulo 2 um

relato histórico e teórico que sustenta o SCP em seus alicerces mais amplos. Iniciando com a ideia prática de resolver um problema e embarcando numa discussão indo até o problema de fazer estimativas de um ente matemático clássico que se revela indomesticável quase, a Hessiana.

Seguindo uma estrutura textual que possibilite a compreensão do que foi produzido ao longo de seis meses de dedicação erguemos o método científico que utilizamos no Capítulo 3 e, em seguida, no Capítulo 4 apresentamos nosso objeto central, a análise analítica do método mediante uma sentença prática. Fazendo uso de lógica de programação foi estruturado um algoritmo auxiliar que serviu para imergir aos instrumentos intrínsecos do SCP e revelar seus mecanismos e manobras. Disso seguimos para uma modelagem de um método construído com um algoritmo inédito, que traduz o SCP de uma forma que possibilite a sua codificação em uma linguagem que consiga compilar suas rotinas.

Além de se aproximar com muita eficiência de um algoritmo que revele o SCP numa plataforma pública e de fácil acesso, os frutos desse trabalho transbordam na área teórica, no que diz respeito a formação de recursos humanos, e no requisito prático, no que diz respeito a elaboração de um código que lê o SCP via lógica de programação.

Ademais, desejamos que esse trabalho possa vir a servir de inspiração a outros trabalhos que evoluam ainda mais os avanços que atingimos e que possam trazer uma estrutura mais sólida para a área da otimização contínua.

2 Revisão Literária

2.1 A programação sob o viés matemático

Visando uma exposição técnica e concisa do conteúdo dessa seção foi elaborado pelo professor orientador desse trabalho o texto que se apresenta.

Desde que começamos a produzir materiais usando de maquinários nos confrontamos com o problema de otimização. Em primeira escala o problema já tinha uma abordagem matemática sólida iniciada em consequência da evolução da matemática analítica que foi fortificada pelos instrumentos advindos do Cálculo Diferencial e Integral.

Mesmo os métodos mais simples, matematicamente, podem ser de alto custo computacional. Esse o objeto desse texto. Apresentar os elementos da otimização de funções reais à uma variável e, sem seguida, à várias variáveis, desconsiderando completamente o fato de que em nosso tempo tais processos são feitos quase que exclusivamente por computadores e seus métodos. E por esse motivo precisam de uma abordagem dirigida ao fato de que o computador precisa entender o que teoricamente pode ser fácil de definir, mas que para os processos de otimização computacionais são delicados, como por exemplo a estimativa de cálculo com números infinitesimais, que adotamos teoricamente como algo concreto, passa a ser um desafio computacional.

Faremos nossa exposição usando de dois momentos. No primeiro definimos o que significa otimizar e apresentamos os elementos que cercam essa definição. Num segundo momento apresentamos critérios de otimalidade que definem condições sob as quais podemos garantir a existência de soluções para os problemas propostos.

Para o que vamos apresentar estamos considerando que o conjunto $\mathbb{R}^n$ é um Espaço Vetorial Normado, com a norma provinda de um produto interno. Representamos a norma por: $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$. Não faremos distinção de notação para as possíveis variações de n .

2.1.1 Funções à Uma Variável Real

Damos início a essa seção apresentando o que devemos entender por processo de otimização. Tornou-se comum o uso de modelagem matemática para a determinação de como os processos de certos fenômenos podem ser simulados. Essa simulação molda o que significamos como sendo uma solução para um problema que está dentro de parâmetros estabelecidos a partir do fenômeno. O instrumento que usamos para representar o modelo é o que matematicamente entendemos como sendo uma função. Que definimos a seguir:

Definição 1 (Função Real). *Uma função real é uma relação entre dois subconjuntos $I, J \subset \mathbb{R}$. Para que seja considerada uma função a relação deve satisfazer as seguintes condições:*

1. *Para todo $x \in I$ deve existir um número real $f(x)$;*
2. *Para todo $x \in I$, deve existir um único $f(x) \in J$.*

O conjunto I é denominado o domínio de $f(x)$ e J a imagem de $f(x)$.

Existem diversos estudos que podemos fazer para que possamos analisar o comportamento de uma função assumindo a variação dos valores de x dentro do subconjunto I . Os avanços teóricos obtidos desde a construção do Cálculo Diferencial nos permite fazer a análise de variação infinitesimal de uma função $f(x)$ que satisfaça determinadas condições.

Para que possamos caracterizar o ente *derivada de uma função real* vamos construir sucintamente uma cadeia de resultados que revelam a natureza desse ente. Um instrumento que usaremos frequentemente é o que denominamos de *intervalo*, que podemos compreender como sendo um subconjunto de $\mathbb{R}$ definido pelos extremos a, b . O conjunto $I = (a, b)$ é o intervalo aberto definido pelos valores de x , tais que $a < x < b$. Equivalentemente, o intervalo fechado $\bar{I} = [a, b]$ é definido pelos valores de x , tais que $a \leq x \leq b$.

Iremos precisar de uma noção íntima ao estudo das derivadas que é a de aproximação infinitesimal. Dado um valor real c , desejamos poder gerar uma entidade que represente uma vizinhança de c , isto é, um conjunto de valores reais que estão suficientemente próximos de c . Para isso é clássico o uso de variáveis de acréscimo e decréscimo infinitesimal. O infinitésimo deve ser compreendido como uma porção arbitrariamente pequena, ou seja, ao assumir que um valor real $\epsilon \approx 0$, estou supondo que a diferença entre ϵ e 0 é tal pequena que podemos considerar que esses valores representam grandezas indistinguíveis. Vamos denotar por uma ϵ -vizinhança de c , o subconjunto $V_c = (c - \epsilon, c + \epsilon)$.

Antes de apresentar o conceito de derivada, devemos apresentar o conceito de continuidade que precede o de derivada e o fomenta de uma forma intrínseca.

Definição 2 (Função Contínua). *Dizemos que uma função real definida em um subconjunto I , um intervalo, é uma função contínua se, e somente se, para todo valor real $\tilde{x} \in I$ pudermos escolher arbitrariamente uma variável infinitesimal, $\epsilon > 0$, que implique na existência de outra variável infinitesimal $\delta > 0$ tais que:*

$$\|x - \tilde{x}\| < \delta \Rightarrow \|f(x) - f(\tilde{x})\| < \epsilon \quad (1)$$

A construção do entendimento do que vem a ser uma função contínua é de grande interesse ao que estamos construindo. Devemos perceber que dizemos que uma função é contínua em um valor real específico $\tilde{x}$, isso confere um caráter local à definição. De fato podemos definir continuidade pontualmente. Mas escolhemos apresentar essa definição que implicitamente é dada para todos os valores de um intervalo real. Ademais, o fato de $\|x - \tilde{x}\| < \delta$ significa na prática que estamos considerando estimativas para o valor de $\tilde{x}$ com um erro de no máximo δ . Que faz implicar que as estimativas para o valor de $f(\tilde{x})$ no máximo vão diferir em ϵ , que descrevemos como: $\|f(x) - f(\tilde{x})\| < \epsilon$.

Essas noções definem um ente que denotamos por limite, e que se apresenta como a seguir:

$$\lim_{x \rightarrow \tilde{x}} f(x) = f(\tilde{x}) \quad (2)$$

essa equação representa o fato de que ao considerarmos x tender à $\tilde{x}$ obtemos por implicação que $f(x)$ tende à $f(\tilde{x})$. Como dissemos que isso ocorre para todos os valores de $\tilde{x} \in I$, então ocorre que o limite de $f(x)$ existe para qualquer valor de $x \in I$. Tecnicamente isso tem uma outra interpretação. Se considerarmos uma sequência de valores reais $\{x_k\}$, em I tais que a partir de um determinado índice K tenhamos $\|x_k - \tilde{x}\| < \delta$. O fato de $f(x)$ ser contínua em I implica que se considerarmos a sequência de valores reais gerados $\{f(x_k)\}$ será tal que $\|f(x_k) - f(\tilde{x})\| < \epsilon$. Em outras palavras, ao considerar uma sequência de valores reais $\{x_k\}$ convergindo para o valor $\tilde{x}$ temos que os valores associados pela imagem da função real contínua $\{f(x_k)\}$ é uma sequência de valores reais convergentes à $f(\tilde{x})$.

Estamos em posição de confrontar os seguintes entendimentos sobre o que desejamos como solução do que chamamos de otimização de funções reais que são os extremos de uma função real.

Definição 3 (Extremos de $f(x)$). *Considerando uma função real $f : I \rightarrow J$, definimos como extremo de $f(x)$ em I um ponto x^* que satisfaça:*

1. $f(x^*) < f(x)$ para todo $x \in I$, nesse caso dizemos que x^* é um mínimo, ou;
2. $f(x^*) > f(x)$ para todo $x \in I$, nesse caso dizemos que x^* é um máximo.

Obviamente que os pontos de mínimo ou máximo podem ser de carácter local ou global. Se considerarmos, por exemplo, uma lista de intervalos I_k tais que $I = \cup_k I_k$, e, ademais, $I_i \cap I_j = \emptyset$. Uma lista de tais intervalos denotamos por partição do intervalo I . Naturalmente, que ao aplicarmos a definição de extremos de $f(x)$ a cada intervalo I_k obtemos potenciais mínimos e máximos que numa visão geral devem ser considerados como locais.

Como um primeiro passo concreto em direção à nossa ambiciosa missão de concatenar a teoria de otimização em alguns comentários, nos deparamos a um resultado clássico, conhecido como o Teorema de Weierstrass, que enunciamos:

Teorema 1 (Teorema de Weierstrass). *Toda função contínua definida em um intervalo fechado assume seus extremos.*

Esse Teorema expressa de forma precisa o que estamos interessados em encontrar. Ele garante a existência de valores de máximo e de mínimo para toda função contínua desde que essa esteja restrita a um intervalo fechado. O modo como apresentamos o Teorema é sugestivo. Em verdade ele se expressa em uma linguagem topológica mais ampla, em termos de compactos. Basta, por enquanto, dizer que os compactos da reta são os intervalos fechados. O que justifica nossa abordagem.

Como primeiro efeito do Teorema de Weierstrass temos o seguinte critério de otimalidade:

Teorema 2 (Critério de Otimalidade I). *Uma função objetivo que seja contínua assume valores de ótimo em intervalos fechados. Isto é, se as regiões de busca dos valores de ótimos de uma função objetivo que seja contínua forem intervalos fechados, então em cada intervalo a função objetivo terá um mínimo local e um máximo local.*

Podemos observar que esse critério garante a existência de valores de ótimo. O que computacionalmente é suficiente, mas ineficiente teoricamente. O que nos leva a considerar um refinamento em nossa abordagem. A fim de fortalecer os critérios de otimalidade vamos considerar a diferenciabilidade.

Definição 4 (Função Diferenciável). *Considerando uma função real $f(x)$ definida num intervalo aberto I . Diremos que $f(x)$ é diferenciável em I se, e somente se, o seguinte limite existir para todo valor real $\tilde{x} \in I$:*

$$f'(\tilde{x}) = \lim_{h \rightarrow 0} \frac{f(\tilde{x} + h) - f(\tilde{x})}{h} \quad (3)$$

onde $f'(\tilde{x})$ denota a derivada de $f(x)$ em $\tilde{x}$.

Agora podemos usar da diferenciabilidade da função $f(x)$ para propor uma análise mais robusta do problema a ser otimizado. É fácil observar que se a função for diferenciável então ela é contínua. O que garante que podemos usar o critério de otimalidade para as funções que sejam sabidamente diferenciáveis. O que nos interessa portanto é encontrar mais informações sobre como encontrar os ótimos da função objetivo. O que nos leva ao seguinte fato:

Teorema 3 (Teorema de Fermat). *Dada uma função real $f : [a, b] \rightarrow J$. Se $f(x)$ for diferenciável em (a, b) , então seus extremos $x^* \in (a, b)$ satisfazem a condição $f'(x^*) = 0$.*

Esse Teorema de Fermat nos leva a um patamar mais sofisticado de análise, pois agora sabemos quais são os candidatos a pontos de extremos de toda função que seja diferenciável em (a, b) e que seja ao menos contínua em $[a, b]$. Os pontos x^* que satisfazem a equação $f'(x^*) = 0$ são comumente chamados de pontos críticos de $f(x)$. O que nos leva ao seguinte critério de otimalidade:

Teorema 4 (Critério de Otimalidade II). *Se a função objetivo for contínua no intervalo fechado $[a, b]$ e diferenciável no intervalo aberto (a, b) , então existem ótimos e esses podem ser determinados fazendo:*

1. $\min\{f(a), f(b), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$, ou;
2. $\max\{f(a), f(b), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$.

Para além dessas análises ainda somos capazes de refinar ainda mais os critérios de otimalidade se considerarmos que a função objetivo seja duas vezes diferenciável e que sua segunda derivada seja uma função contínua. A segunda derivada de uma função é precisamente o que parece ser, a derivada da função derivada. Isto é,

$$f''(\tilde{x}) = \lim_{h \rightarrow 0} \frac{f'(\tilde{x} + h) - f'(\tilde{x})}{h} \quad (4)$$

Para dar contexto ao que vamos apresentar devemos fazer a exposição de um resultado deveras valoroso que permite fazermos uma representação local de uma função duas vezes diferenciável da forma seguinte:

Teorema 5 (Teorema de Taylor). *Dada uma função real $f : [a, b] \rightarrow J$. Se $f(x)$ for duas vezes diferenciável em (a, b) , então para todo $c \in (a, b)$ temos que:*

$$f(x) = f(c) + f'(c)(x - c) + \frac{1}{2}f''(c)(x - c)^2 + r(x), \quad (5)$$

com $\lim_{x \rightarrow c} \frac{r(x)}{|x - c|^3} = 0$.

Portanto, ao aplicarmos o Teorema de Taylor a um ponto crítico de $f(x)$ obtemos que:

$$f(x) - f(x^*) \approx \frac{1}{2}f''(x^*)(x - x^*)^2 \quad (6)$$

O termo $(x - x^*)^2$ é definido positivo. Logo o sinal de variação da função com relação ao ponto crítico x^* é dado pela aplicação da segunda derivada ao ponto crítico, isto é, é dado por $f''(x^*)$. Esse fato sugere as seguintes definições:

Definição 5 (Função Convexa). *Dada uma função real duas vezes diferenciável em (a, b) . Diremos que $f(x)$ é convexa se sua segunda derivada for definida positiva, isto é, se $f''(x) > 0$ para todo $x \in (a, b)$.*

Como uma implicação direta da definição de função convexa temos que se uma função convexa possui um ponto crítico em x^* então esse ponto é um ponto de mínimo global de $f(x)$. Esse fato sugere que as funções que são mais facilmente otimizadas são as que são convexas. Equivalentemente, podemos definir funções côncavas, exigindo que a segunda derivada seja definida negativa, isto é, $f''(x) < 0$ para todo x no domínio de diferenciabilidade da função.

Uma implicação desses argumentos é o seguinte critério de otimalidade:

Teorema 6 (Critério de Otimalidade III). *Se a função objetivo for contínua convexa no intervalo fechado $[a, b]$ e duas vezes diferenciável no intervalo aberto (a, b) , então existe um ótimo e esse pode ser encontrado fazendo: $\min\{f(a), f(b), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$.*

Esse último critério traz uma expressão definitiva sobre o ambiente que desejamos simular o fenômeno. Os modelos de otimalidade mais eficazes são aqueles em que a função objetivo é contínua convexa num intervalo fechado. Salientando que o intervalo fechado é o melhor ambiente para a confecção de valores de ótimo tendo em vista o fato de serem conjuntos compactos.

Podemos concluir que a otimização de funções à uma variável real é algo tangível, visto que podemos equilibrar o modelo para uma restrição de seu domínio natural ou amplificar se necessário. Restando a tarefa de decifrar essas configurações à uma linguagem computacional acessível e razoavelmente controlável.

2.1.2 Funções à Várias Variáveis Reais

Nessa seção exporemos o contexto de otimização para funções à várias variáveis reais. A primeira coisa que faremos é definir um ente.

Definição 6 (Função Real à várias variáveis). *Uma função real à várias variáveis é uma relação entre dois conjuntos $\mathcal{U} \subset \mathbb{R}^n$ e $J \subset \mathbb{R}$. Para que seja considerada uma função a relação deve satisfazer as seguintes condições:*

1. *Para todo $x \in \mathcal{U}$ deve existir um número real $f(x)$;*
2. *Para todo $x \in \mathcal{U}$, deve existir um único $f(x) \in J$.*

O conjunto $\mathcal{U}$ é denominado o domínio de $f(x)$ e J a imagem de $f(x)$.

Diante ao novo contexto, para que possamos caracterizar o ente *derivada de uma função real à várias variáveis* vamos construir sucintamente uma cadeia de resultados que revelam a natureza desse ente. Agora passamos a ter como domínio no mínimo um subconjunto do plano $\mathbb{R}^2$. Que vamos usar como protótipo de domínio para o argumento inicial. Diferente do intervalo que é um subconjunto da reta real, num subconjunto do plano real temos uma liberdade infinita de direções. Por causa disso definir e compreender conceitos como limite e derivada passa antes pode um barema de construção geométrico de aproximação.

Para que possamos ter um lastro vamos começar definindo o que seria uma vizinhança de um ponto num subconjunto $\mathcal{U} \subset \mathbb{R}^2$. Dado um ponto $a \in \mathcal{U}$ definimos como a ϵ -vizinhança de a o conjunto: $\mathcal{B}_\epsilon(a) = \{x \in \mathcal{U} \mid \|x - a\| < \epsilon\}$. Esse conjunto é comumente denominado de bola aberta de centro a e raio ϵ . A partir da definição desse conjunto já podemos ver que temos infinitas maneiras de se aproximar do ponto a por pontos de $\mathcal{B}_\epsilon(a)$. Por exemplo, se considerarmos as retas que passam pelo ponto $(0,0)$ e que tem equação $y = m \cdot x$. O fato de podermos escolher arbitrariamente um valor para m indica nossa arbitrariedade de direções de aproximação ao ponto a . É nesse contexto que iremos confeccionar nossos instrumentos de análise.

Definição 7 (Função Contínua). *Dizemos que uma função real definida em um subconjunto $\mathcal{U} \subset \mathbb{R}^n$ é uma função contínua se, e somente se, para todo ponto $\tilde{x} \in \mathcal{U}$ pudermos escolher arbitrariamente uma variável infinitesimal, $\epsilon > 0$, que implique na existência de outra variável infinitesimal $\delta > 0$ tais que:*

$$\|x - \tilde{x}\| < \delta \Rightarrow \|f(x) - f(\tilde{x})\| < \epsilon \quad (7)$$

Tal como feito para funções reais à uma variável, temos que essas noções definem um ente que denotamos por limite, e que se apresenta como a seguir:

$$\lim_{x \rightarrow \tilde{x}} f(x) = f(\tilde{x}) \quad (8)$$

Essa equação representa o fato de que ao considerarmos x tendendo à $\tilde{x}$ obtemos por implicação que $f(x)$ tende à $f(\tilde{x})$. Como dissemos que isso ocorre para todos os valores de $\tilde{x} \in \mathcal{U}$, então ocorre que o limite de $f(x)$ existe para qualquer valor de $x \in \mathcal{U}$.

Tecnicamente isso tem uma outra interpretação. Se considerarmos uma sequência de pontos $\{x_k\}$, em $\mathcal{U}$ tais que a partir de um determinado índice K tenhamos $\|x_k - \tilde{x}\| < \delta$. O fato de $f(x)$ ser contínua em $\mathcal{U}$ implica que se considerarmos a sequência de pontos gerados $\{f(x_k)\}$ será tal que $\|f(x_k) - f(\tilde{x})\| < \epsilon$. Em outras palavras, ao considerar uma sequência de pontos $\{x_k\}$ convergindo para o valor $\tilde{x}$ temos que os pontos associados pela imagem da função real contínua $\{f(x_k)\}$ é uma sequência de pontos convergentes à $f(\tilde{x})$.

Definição 8 (Extremos de $f(x)$). *Considerando uma função real $f : \mathcal{U} \subset \mathbb{R}^n \rightarrow J$, definimos como extremo de $f(x)$ em $\mathcal{U}$ um ponto x^* que satisfaça:*

1. $f(x^*) < f(x)$ para todo $x \in \mathcal{U}$, nesse caso dizemos que x^* é um mínimo, ou;
2. $f(x^*) > f(x)$ para todo $x \in \mathcal{U}$, nesse caso dizemos que x^* é um máximo.

Como na primeira seção faremos aqui um passo concreto em direção à nossa teoria de otimização em alguns comentários. Apresentamos a versão geral do clássico Teorema de Weierstrass, que enunciamos:

Teorema 7 (Teorema de Weierstrass). *Toda função contínua definida em um compacto assume seus extremos.*

Esse Teorema expressa de forma precisa o que estamos interessados em encontrar. Ele garante a existência de valores de máximo e de mínimo para toda função contínua desde que essa esteja restrita a um compacto do $\mathbb{R}^n$. Como primeiro efeito do Teorema de Weierstrass temos o seguinte critério de otimalidade:

Teorema 8 (Critério de Otimalidade IV). *Uma função objetivo que seja contínua assume valores de ótimo em conjuntos compactos. Isto é, se as regiões de busca dos valores de ótimos de uma função objetivo que seja contínua forem conjuntos compactos, então a função objetivo terá um mínimo local e um máximo local.*

Podemos observar que esse critério garante a existência de valores de ótimo. O que computacionalmente é suficiente, mas ineficiente teoricamente. O que nos leva a considerar um refinamento em nossa abordagem. A fim de fortalecer os critérios de otimalidade vamos considerar a diferenciabilidade.

Definição 9 (Função Diferenciável). *Considerando uma função real $f(x)$ definida num subconjunto $\mathcal{U} \subset \mathbb{R}^n$. Diremos que $f(x)$ é diferenciável em $\mathcal{U}$ se, e somente se, existirem n*

números reais $\{A_1, \dots, A_n\}$, tais que:

$$f(x + v) - f(x) = A_1 \cdot v_1 + \dots + A_n \cdot v_n + r(v), \quad (9)$$

para todo $x \in \mathcal{U}$, e $v = (v_1, \dots, v_n)$ tal que $x + v \in \mathcal{U}$. Com $\lim_{v \rightarrow 0} \frac{r(v)}{\|v\|} = 0$.

Agora podemos usar da diferenciabilidade da função $f(x)$ para propor uma análise mais robusta do problema a ser otimizado. É fácil observar que se a função for diferenciável então ela é contínua. O que garante que podemos usar o critério de otimalidade para as funções que sejam sabidamente diferenciáveis. O que nos interessa portanto é encontrar mais informações sobre como encontrar os ótimos da função objetivo.

Segue direto da definição de função diferenciável que podemos calcular a derivada de $f(x)$ fazendo uso dos vetores canônicos de $\mathbb{R}^n$, obtendo que os números reais $\{A_1, \dots, A_n\}$ são o que denominamos de derivadas parciais da função $f(x)$ com relação às suas coordenadas, como segue:

$$\frac{\partial f}{\partial x_i}(x) = A_i \quad (10)$$

Tal como fizemos na seção anterior, podemos definir essas derivadas parciais a partir do cálculo de um limite, com a seguir:

$$\frac{\partial f}{\partial x_i}(x) = \lim_{h \rightarrow 0} \frac{f(x + h \cdot e_i) - f(x)}{h} \quad (11)$$

Podemos fazer uso desses argumentos para definir o que denotamos como a diferencial de $f(x)$ que seria o equivalente a derivada de $f(x)$ quando tínhamos uma variável.

Definição 10 (O Diferencial de $f(x)$). *Seja $f(x)$ uma função real à várias variáveis definida em um aberto $\mathcal{U} \subset \mathbb{R}^n$. Supondo que as n derivadas parciais existam e que sejam funções contínuas, temos o diferencial de $f(x)$ dado com respeito à uma base canônica, $\{dx_1, \dots, dx_n\}$, do Espaço Dual $\mathbb{R}^*$ pela expressão:*

$$Df(x) = \sum_{i=1}^n \frac{\partial f}{\partial x_i}(x) \cdot dx_i \quad (12)$$

Podemos definir os pontos críticos de $f(x)$ a partir do conjunto solução da equação: $Df(x) = 0$. Segue dessa definição que num ponto crítico todas as derivadas parciais são nulas, isto é, $\frac{\partial f}{\partial x_i}(x) = 0$. Temos para as funções à várias variáveis o equivalente ao Teorema de Fermat:

Teorema 9. *Seja $f(x)$ uma função real à várias variáveis definida em um aberto $\mathcal{U} \subset \mathbb{R}^n$. Se $f(x)$ for diferenciável em $\mathcal{U}$, então seus extremos $x^* \in \mathcal{U}$, satisfazem a condição $Df(x^*) = 0$.*

Esse Teorema nos leva a uma sofisticação da análise, pois agora sabemos quais são os candidatos a pontos de extremos de toda função que seja diferenciável em um aberto $\mathcal{U}$ de $\mathbb{R}^n$ e que seja ao menos contínua no fecho do conjunto $\mathcal{U}$, desde que esse seja um conjunto

compacto. Podemos entender o fecho de um conjunto como sendo a coleção de todos os pontos de acumulação do conjunto. Os pontos de fronteira são todos os pontos que definem o interior e o exterior do conjunto. Denotamos o conjunto pontos de fronteira, ou a fronteira do conjunto $\mathcal{U}$ por $\partial\mathcal{U}$. O que nos leva ao seguinte critério de otimalidade:

Teorema 10 (Critério de Otimalidade V). *Se a função objetivo for contínua no compacto $\bar{\mathcal{U}}$ e diferenciável no aberto $\mathcal{U}$, então existem ótimos e esses podem ser determinados fazendo:*

1. $\min\{f(x), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$ em $\mathcal{U}$ e $x \in \partial\mathcal{U}$, ou;
2. $\max\{f(x), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$ em $\mathcal{U}$ e $x \in \partial\mathcal{U}$.

Para além dessas análises ainda somos capazes de refinar ainda mais os critérios de otimalidade se considerarmos que a função objetivo seja duas vezes diferenciável e que sua segunda derivada seja uma função contínua. A segunda derivada de uma função é precisamente o que parece ser, a derivada da função derivada. Isto é,

$$D(Df)(x) = D^2f(x) \quad (13)$$

Para dar contexto ao que vamos apresentar devemos fazer a exposição de um resultado deveras valoroso que permite fazermos uma representação local de uma função duas vezes diferenciável da forma seguinte:

Teorema 11 (Teorema de Taylor). *Seja $f(x)$ uma função real à várias variáveis definida em um aberto $\mathcal{U} \subset \mathbb{R}^n$. Se $f(x)$ for duas vezes diferenciável em $\mathcal{U}$, então:*

$$f(x+v) = f(x) + Df(x) \cdot v + \frac{1}{2} D^2f(x) \cdot v^2 + r(x), \quad (14)$$

onde $v \in \mathbb{R}^n$, tal que $x+v \in \mathcal{U}$. Com $\lim_{v \rightarrow 0} \frac{r(v)}{\|v\|^2} = 0$.

Nessa versão do Teorema de Taylor apresentamos implicitamente um conceito deveras importante para a análise de funções reais à várias variáveis, a saber, a forma Hessiana. Denotada por $D^2f(x) \cdot v^2$ a forma Hessiana, tem o mesmo papel que a segunda derivada para uma função real à uma variável. Visto que podemos encontrar uma expressão de $f(x)$ numa vizinhança de um ponto crítico dada por:

$$f(x^*+v) - f(x^*) \approx \frac{1}{2} D^2f(x^*) \cdot v^2 \quad (15)$$

Precisamos encontrar uma maneira de analisar a variação da forma Hessiana. Uma que se ajusta muito bem por diversas razões é a forma matricial. A matriz da forma Hessiana, é denominada de matriz Hessiana, e tem a seguinte expressão:

$$[D^2f(x)] = H = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right) \quad (16)$$

Analogamente podemos definir a matriz associada ao funcional $Df(x)$ que é denominada de matriz Jacobiana e tem a expressão:

$$[Df(x)] = J = \left(\frac{\partial f}{\partial x_i}(x) \right) \quad (17)$$

Obviamente que a matriz Jacobiana é uma matriz linha, ou coluna. Enquanto que a matriz Hessiana é uma matriz quadrada de ordem n . Há um fato muito relevante que faz da matriz Hessiana um manjar da otimização que é o fato de ser uma matriz simétrica, quando consideramos condições simples para a função $f(x)$. A fim de que o Teorema de Shwarz seja satisfeito. Por ser simétrica podemos definir um produto interno sempre que ela for definida positiva. Isso gera um estrutura deveras importante.

Associado ao funcional $Df(x)$ temos um vetor que desempenha para a otimização um papel de príncipe. Esse vetor é denominado de vetor gradiente, representado por $\nabla f(x)$ e é definido a partir da seguinte equação:

$$Df(x) \cdot v = \langle \nabla f(x), v \rangle \quad (18)$$

Ao usarmos a definição de ângulo entre vetores no $\mathbb{R}^n$, provinda do produto interno, temos que:

$$\cos(\theta_{\nabla f(x), v}) = \frac{\langle \nabla f(x), v \rangle}{\|\nabla f(x)\| \cdot \|v\|} \quad (19)$$

Podemos considerar, sem perda de generalidade que $\|v\| = 1$, encontrando que:

$$\cos(\theta_{\nabla f(x), v}) \|\nabla f(x)\| = \langle \nabla f(x), v \rangle = Df(x) \cdot v \quad (20)$$

Portanto, podemos concluir que a direção para v na qual o funcional $Df(x)$ assume o maior valor é precisamente $v = \nabla f(x)$ visto que é nessa direção que o ângulo $\theta_{\nabla f(x), \nabla f(x)} = 0$ tornando o $\cos(\theta_{\nabla f(x), \nabla f(x)}) = 1$, logo: $Df(x) \cdot v = \|\nabla f(x)\|$.

Além desse fato, o gradiente é facilmente calculado visto que ele é definido pelas derivadas parciais:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (21)$$

Definição 11 (Função Convexa). *Dada uma função real duas vezes diferenciável em $\mathcal{U}$. Diremos que $f(x)$ é convexa se sua forma Hessiana for definida positiva, isto é, se $H \cdot v^2 > 0$ para todo $x \in \mathcal{U}$.*

Como uma implicação direta da definição de função convexa temos que se uma função convexa possui um ponto crítico em x^* então esse ponto é um ponto de mínimo global de $f(x)$. Esse fato sugere que as funções que são mais facilmente otimizadas são as que são convexas. Equivalentemente, podemos definir funções côncavas, exigindo que a forma Hessiana seja definida negativa, isto é, $H \cdot v^2 < 0$ para todo x no domínio de diferenciabilidade da função.

Uma implicação desses argumentos é o seguinte critério de otimalidade:

Teorema 12 (Critério de Otimalidade VI). *Se a função objetivo for contínua convexa no $\bar{\mathcal{U}}$ e duas vezes diferenciável no aberto $\mathcal{U}$, então existe um ótimo global e esse pode ser encontrado fazendo: $\min\{f(x), f(x^*)\}$, onde x^* é ponto crítico de $f(x)$ e $x \in \partial\mathcal{U}$.*

Diferente das funções à uma variável real essas opções não chegam a esgotar as possibilidades de representação de fenômenos a partir de modelos matemáticos. Pode ocorrer de termos uma série de restrições que devem ser satisfeitas pelos valores do domínio da função objetivo em concomitância as condições implicadas pela função $f(x)$. Para essa estrutura, que se apresenta como uma generalização num sentido de representação de problemas em que a função objetivo seja contínua e ao menos duas vezes diferenciável, temos o seguinte teorema devido a Lagrange:

Teorema 13 (Multiplicadores de Lagrange). *Seja $f(x)$ uma função duas vezes diferenciável em um aberto $\mathcal{U}$ do $\mathbb{R}^{n+1}$. E seja $M = \phi^{-1}(c)$ uma hiperfície de $\mathbb{R}^n + 1$. Então a fim de que exista um ótimo para a restrição $f|M$ é suficiente e necessário que exista um número real λ tal que:*

$$\nabla f(x) = \lambda \cdot \nabla \phi(x) \quad (22)$$

O número real λ é denominado de multiplicador de Lagrange.

Podemos considerar o próprio Teorema dos Multiplicadores de Lagrange como um critério de otimalidade. Pois ele define um método que garante a existência de um ótimo para a restrição $f|M$ que nada mais é do que a função objetivo restrita às condições de otimalidade.

É possível encontrar outras descrições e argumentos que ampliem o leque da compreensão de como tratar da otimização de modelos matemáticos de forma matemática, assim como uma exposição mais abrangente de todos os temas expostos nesse texto nos seguintes livros: (HALMOS, 1978), (LANG, 1969), (LIMA, 1995), (LIMA, 1989), (LIMA, 2021), (LIMA, 1977) e (TAO, 2016).

2.2 As primeiras três décadas da programação

Na década de 1930 o mundo encontrava-se em um estado onde problemas de gerenciamento se tornaram muito complexos, e tão pouco estruturados, que modelos matemáticos e o poder computacional existente na época foram postos à prova. A partir daí o campo de pesquisa responsável por esses assuntos começou a tomar forma e ganhar mais importância. Dentre os vários nomes dados a essa área de pesquisa, podemos encontrar: pesquisa operacional, análise de sistemas, análise de custo-benefício e análise de custo-efetividade. Durante essa década, onde iniciava-se a Segunda Guerra Mundial, problemas como a alocação de recursos limitados entre atividades simultâneas sob certas restrições eram comuns.

2.2.1 L. V. Kantorovich

Em 1939, problemas de otimização de produções com base nas reservas da indústria (materiais, trabalhadores e equipamentos) se tornaram essenciais no sustento do estado soviético. Segundo Kantorovich (KANTOROVICH, 1960), existem duas formas de melhorar a eficiência de uma

fábrica. A primeira é melhorando a tecnologia empregada nos meios de produção, e a outra é melhorando a organização do planejamento e da produção.

Antes disso, Kantorovich havia percebido que uma grande quantidade de problemas, de diversas áreas, levavam à formulação de problemas de extremos. Mas ele também notou que esses problemas, embora similares à problemas da análise matemática, quando se tentava resolver usando os mesmos processos matemáticos, formalmente definidos, se tornavam inúteis, pois era deveras custoso no aspecto computacional. Dentre um dos exemplos está a resolução de milhares ou milhões de sistemas de equações.

Ele foi capaz de reduzir vários dos problemas que encontrou em três tipos. Partindo do problema de alocação de tempo de uso de máquinas para a produção de partes que compõem um dado produto, foi formulado um problema genérico considerando um número n de máquinas, e a produção de m partes diferentes para confecção do produto. Usando a i -ésima máquina para produzir apenas a k -ésima parte durante um dia inteiro de trabalho, seria produzido $\alpha_{i,k}$ partes. O problema consiste em dividir o tempo de trabalhado de cada máquina de forma que possa ser construído o máximo possível de produtos.

Considerando $h_{i,k}$ como a fração do dia de trabalho que a máquina i passou produzindo a parte k , como sendo a variável que desejamos otimizar. Por motivos óbvios, ela deve ser positiva. E também, por não podermos considerar que uma máquina esteja parada durante o dia de trabalho, temos duas restrições:

1. $h_{i,k} \geq 0$, para $i = 1, \dots, m$;
2. $\sum_{k=1}^m h_{i,k} = 1$, para $i = 1, \dots, m$;

Observando que a restrição 2. significa que a soma das frações $h_{i,k}$ deve resultar em 1, visto que a unidade de tempo é o dia. Portanto, temos que a quantidade da k -ésima parte produzida no dia é:

$$z_k = \sum_{i=1}^n \alpha_{i,k} h_{i,k}. \quad (23)$$

Para que se obtenha o máximo de produtos completos, é necessário que se tenha a mesma quantidade de cada tipo de parte, do contrário foi perdido tempo produzindo partes extras. Com isso temos o *Problema A*, cuja configuração é dada pelas restrições a seguir:

Encontrar $h_{i,k}$ tal que as partes z_k sejam todas iguais, isto é $z_{k_1} = z_{k_2}$ para $k_1, k_2 \in \{1, \dots, m\}$. Onde z_k é dado pela expressão (23) e adicionalmente $h_{i,k}$ satisfazendo as restrições descritas em (2).

Problemas como o uso racional de combustível e distribuição ótima de terras aráveis também foram usados como exemplos de problemas que podem ser reduzidos a mesma forma que o Problema A.

Caso tenhamos mais condições limitantes, temos um outro modelo esperado por Kantorovich. Considerando o mesmo problema de produção anteriormente definido, e agora com limite máximo de energia gasta C para todas as máquinas e, $c_{i,k}$ sendo o consumo de energia da máquina i produzindo a parte k por um dia inteiro de trabalho, temos o Problema A com

mais uma restrição, chamado de Problema B. Onde essa nova restrição é dada por:

$$\sum_{i=1}^n \sum_{k=1}^m c_{i,k} h_{i,k} \leq C \quad (24)$$

Podem existir vários significados sobre as quantidades em torno de C , como, por exemplo, a quantidade de pessoas disponíveis para trabalho, consumo de água ou qualquer recurso necessário para a produção. Da mesma forma podemos ter diversas restrições como esta, o que ainda é classificado como um problema tipo Problema B.

Ainda há um terceiro modelo de problema, quando uma mesma máquina pode produzir ao mesmo tempo mais de uma parte, e que ela pode ser configurada para operar de diferentes métodos, com cada método resultando em números diferentes de partes. Considerando $\gamma_{i,k,l}$ o número de k -ésimas partes, produzidas na i -ésima máquina, utilizando a l -ésima configuração, é preciso encontrar a fração de horas do dia trabalhado $h_{i,l}$ que uma máquina i deverá passar operando na configuração l . O número de k -ésimas partes é dado por $z_k = \sum_{i,l} \gamma_{i,k,l} h_{i,l}$. São aplicados as restrições da mesma forma para o Problema A e o Problema B. Com isso temos o Problema C:

$$\max \quad z_k \quad (25a)$$

$$\text{s.a} \quad z_i = z_j, \quad \forall i, j \leq m \quad (25b)$$

$$z_k = \sum_{i,l} \gamma_{i,k,l} h_{i,l}, \quad k = 1, \dots, m \quad (25c)$$

$$h_{i,l} \geq 0, \quad \forall i, l \quad (25d)$$

$$\sum_l h_{i,l} = 1, \quad \forall i, l \quad (25e)$$

Para a resolução de tais problemas, Kantorovich desenvolveu um método simples. A partir do Problema A, é certo que existe um conjunto de multiplicadores $\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ com cada um correspondendo a uma parte produzida, que são capazes de levar à solução do problema. A partir deles podemos observar cada máquina i e escolher os multiplicadores que geram os máximos entre $\{\lambda_1 \alpha_{i,1}, \lambda_2 \alpha_{i,2}, \dots, \lambda_m \alpha_{i,m}\}$. Em seguida, os multiplicadores λ_k observados nas máquinas i que resultaram em máximos são postos como $h_{i,k}$. Feito isso, fica muito mais fácil determinar os outros valores para $h_{i,k}$.

2.2.2 G. B. Dantzig

Ainda na década de 40, mas desta vez nos Estados Unidos da América, os mesmos problemas de alocação de recursos surgiam, uma vez dado o contexto de guerra. Dada necessidade de coordenação entre os diversos problemas que demandavam atenção, Dantzig ([DANTZIG; THAPA, 1997](#)), ao mesmo tempo que Kantorovich, desenvolveu modelos e métodos para a resolução de problemas lineares.

Para dar exemplos de uso, Dantzig elencou alguns problemas rotineiros para ele nas décadas de 30-40. Dentre os problemas, podemos falar sobre uma empresa vendedora de mesas. A empresa tem quatro modelos de mesas, onde cada mesa é construída em uma carpinteira e depois mandada para o acabamento. O número de horas de trabalho e o número de horas disponíveis para cada lugar são conhecidos. É certo a disponibilidade de matérias-primas e a venda das mesas fabricadas. O objetivo da empresa é saber quantas mesas de cada modelo irá encomendar de forma que maximize o lucro, e isso facilmente pode ser representado como um problema linear.

Dentre as várias contribuições dadas por Dantzig, podemos destacar três. A primeira sendo a padronização de um problema linear. Como já visto, o problema de otimização linear consiste em otimizar uma expressão linear com n variáveis, mas considerando um conjunto de m igualdades também lineares:

$$\min_x \quad c_1x_1 + \dots + c_nx_n \quad (26a)$$

$$\text{s.a} \quad a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \quad (26b)$$

$$a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \quad (26c)$$

$$a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \quad (26d)$$

Considerando em notação de matrizes e vetores:

$$\min_x \quad c^T x \quad (27a)$$

$$\text{s.a} \quad Ax = b \quad (27b)$$

$$x \geq 0 \quad (27c)$$

A restrição (27c) foi considerada parte da formulação padrão uma vez que os problemas consistiam em escolha de número de itens, o que limita a escolha a ser nula ou positiva, mas sem muitas dificuldades podemos tornar um problema com variáveis negativas nesta forma.

Uma segunda contribuição foram instruções práticas de como modelar problemas lineares. São consideradas duas abordagens para modelagem, por receita/atividade e por balanço de materiais. A primeira abordagem considera uma coleção de processos que são tomados durante a situação que se deseja otimizar. Já a segunda, uma coleção de restrições acerca de materiais ou recursos que serão usados nos processos. Dantzig recomenda, se possível, tomar as duas abordagens, mesmo que gerem o mesmo modelo, uma vez que o ganho de conhecimento da situação trazida por ambas abordagens é sempre de interesse.

A terceira contribuição, certamente uma das que lhe trouxe fama, foi o método simplex para resolução de problemas lineares. O método opera sobre o problema na forma canônica, que considera o sinal $\leq$ ao invés de $=$ em (131b). A transformação de um problema para o outro é trivial, basta duplicar cada linha de A e b e cada linha duplicada é multiplicado por -1 . As restrições definem um polítopo, um exemplo de polítopo de 3 dimensões é um poliedro.

Uma característica de um problema linear é que a solução é um vértice do politopo. Para ver isso basta considerar um poliedro e verificar que o ponto mais alto (ou mais baixo) do poliedro é um vértice. Caso qualquer ponto em uma face do poliedro seja uma solução, isso também faz com que qualquer vértice daquela face seja uma solução.

O método simplex, de forma geral, faz operações em uma matriz gerada a partir do problema até que se produza a solução. Uma forma de visualização para essas operações é considerar que cada operação move a solução atual de um vértice para um vértice vizinho.

2.3 Programação não linear - Um possível início?

Ao mesmo tempo que se refinavam as técnicas de otimização de problemas lineares, o que ocorreu alguns anos após os estudos iniciais de Dantzig e Kantorovich, mais esforços foram dirigidos à outras áreas da otimização. Problemas de otimização não linear já vinham sendo trabalhados há algum tempo, principalmente problemas de otimização não restrita. O método de Newton, descrito em *De analysi per aequationes numero terminorum infinitas*, que é um método capaz de encontrar raízes de funções diferenciáveis, tomou a liderança dentre os métodos de otimização usados para tais problemas, e encontrou um "rival" cerca de dois séculos depois, o método do gradiente descendente, proposto por Cauchy ([LEMARÉCHAL, 2012](#)) em 1847. Nesse contexto, após a fixação de métodos para resolução de problemas lineares, foi iniciada uma busca por métodos para problemas não lineares usando essas novas ferramentas. Alguns desses métodos servem como fundamentos para o SCP, outros explicam as circunstâncias históricas do SCP.

2.3.1 J. E. Kelley Jr.

Em 1960, a otimização restrita de problemas de mínimo encontrava pouco sucesso. No entanto, um pequeno subconjunto desses problemas estava gerando técnicas muito úteis, e um desses subconjuntos era a minimização de uma função convexa em um conjunto fechado e convexo, que foi alvo de muito interesse na época. O motivo de tanto interesse era que, como a função é convexa, todo ponto de mínimo local é mínimo global.

Ainda em 1960, J.E. Kelley ([KELLEY, 1960](#)), posto como responsável pela disseminação dos métodos de plano cortante desenvolve um método de otimização capaz de resolver problemas convexos a partir de suposições mais genéricas. O método proposto por Kelley é iterativo e consiste na resolução de problemas lineares.

Esse método faz parte do crescimento do estudo do problema de ajuste de curvas de Chebyshev ([KELLEY, 1959](#)), ([KELLEY, 1958](#)). O problema do ajuste de curvas pode ser resolvido minimizando a seguinte função:

$$\min_{\alpha} \sigma(\alpha, x) = f(x) - \sum_j \alpha_j g_j(x) \quad (28)$$

Em resumo, o objetivo desse problema de minimização é encontrar escalares para um certo conjunto de funções, que quando combinadas aproximam a função f . Dentre as

aplicações, podemos citar casos onde a computação de todas as funções g_j é menos custosa que a computação de f .

Outros pesquisadores, como Cheney e Goldstein ([CHENEY; GOLDSTEIN, 1959](#)) e Phillip Wolfe ([WOLFE, 1960](#)), das condições de Wolfe, chegaram independentemente a este método, todos inspirados pelo trabalho de Remez.

Em 1934, Remez ([REMEZ, 1934](#)) desenvolveu um método de ajuste de curvas, que constrói o polinômio com melhor aproximação para certas funções sob algumas condições. A forma de resolução é dada pela resolução de problemas lineares, análise de erro, e atualização do estado atual baseado em condições sobre o erro. Esses passos se tornaram base para toda a família de otimizadores pelo fato de gerar informações sobre direções do ótimo baseado numa análise inteligente do comportamento da função.

2.3.2 R. E. Griffith, R. A. Stewart

Em 1961, R. E. Griffith e R. A. Stewart ([GRIFFITH; STEWART, 1961](#)) foram responsáveis por uma publicação de caráter prático. Desenvolveram uma técnica de otimização não linear para sistemas de processamento contínuo. Um método responsável por otimizar um problema da companhia de petróleo Shell. Trabalhando de forma similar ao proposto por J.E. Kelley, usando resoluções de subproblemas lineares, o processo consiste em resolver os subproblemas lineares de forma que a solução deles venham a convergir para a solução do problema principal.

O método aceita problemas formulados respeitando um certo conjunto de regras, mas na prática nem sempre é necessário que sigam todas as funções visto que a solução dada é boa o suficiente. É dito que as melhores formulações para o problema necessitam de que as variáveis sejam separadas em lineares e não lineares; que as restrições envolvam tanto variáveis lineares e não lineares; que todas as funções sejam contínuas e, preferencialmente, bem comportadas. Um dos pontos positivos é que a escolha do ponto inicial para dar início ao algoritmo iterativo não afeta tanto o resultado final.

A forma como a convergência das soluções dos subproblemas lineares resolvidos é mostrada é bastante ingênua. O método sendo iterativo precisa de um ponto inicial, e tal ponto inicial é o ponto dado como solução do subproblema linear resolvido anteriormente. E como a região onde esse subproblema é resolvido respeita as restrições do problema principal, então a sequência de pontos gerados converge para a solução ótima dentro de polígonos gerados pela linearização das restrições, sendo a solução uma aresta desse polígono.

2.3.3 S. P. Han

Em 1977, S. P. Han ([HAN, 1977](#)) publicou um artigo no qual demonstrava um método que mudaria os rumos do estudo de otimização na época. Os métodos newtonianos que vinham sendo desenvolvidos eram de convergência a um ótimo local apenas, até que S.P. Han mostrou um formato que a convergência se tornaria a um ótimo global. Esse formato tem como objetivo resolver uma forma quadrática do problema original, encontrando uma direção de descida, e,

ao invés de analisar a otimização da função original, analisa-se uma função de mérito que leva em consideração propriedades locais da função, tornando o método capaz de convergir para um ótimo global, e mais, de forma superlinear.

Após a publicação desse artigo, uma grande quantidade de publicações foram feitas usando como base as técnicas de otimização sequencial quadrática (SQP), e até hoje, uma grande parte dos otimizadores usam de alguma forma tais técnicas.

2.3.4 Richard H. Byrd, et al.

Mesmo o SQP sendo uma das abordagens mais bem sucedidas para otimização em larga escala, como problemas com centenas de variáveis e restrições, existe um defeito, justamente neste caso, a abordagem se torna computacionalmente ineficiente. Por isso, Bryd, Gould, Nocedal e Waltz (BYRD et al., 2003), desenvolveram um novo método onde o problema quadrático não precisa ser resolvido completamente, em que a resolução de um problema linear limite o que precisa ser avaliado pelo problema quadrático, reduzindo assim o custo computacional da etapa mais pesada do método, a resolução de problemas quadráticos. O que os métodos anteriores vinham fazendo era avaliar as funções de restrição ao mesmo tempo que se fazia a computação da atualização do ponto da sequência naquele momento. A mudança que este método fez foi separar cada etapa da resolução e buscar uma melhor forma de resolver cada passo. O primeiro passo, feito por uma resolução de um problema linear reduz o conjunto de restrições e define uma direção de minimização. Feito isso é computado um modelo quadrático minimizando uma função de mérito que faz uso da direção dada pela resolução do problema linear. Por fim, o problema quadrático é resolvido usando o conjunto de restrições determinada na primeira etapa.

2.4 Os pilares arquitetônicos do SCP

O algoritmo SCP (STILL; WESTERLUND, 2010) apresentado é a extensão de um trabalho sobre as mesmas bases. Anteriormente, os autores haviam desenvolvido um método capaz de resolver apenas problemas convexos com restrições de desigualdades não lineares.

Assim como o algoritmo antes desse, o objetivo do *Sequential Cutting Plane* (SCP) é melhorar a performance da resolução de problemas onde todas as funções são computacionalmente baratas de calcular, já que não se busca minimizar a quantidade de funções computadas durante o processo. Portanto, se a computação de alguma função dentro do problema for computacionalmente custosa, então outro algoritmo deve fazer esse trabalho melhor.

Os autores pedem que o SCP, acrônimo para *Sequential Cutting Plane*, não seja confundido, com outro SCP, que seria *Sequential Convex Programming*. No *Sequential Convex Programming*, o problema é resolvido por meio de uma sequência de resoluções de subproblemas convexos não lineares separáveis. Já no *Sequential Cutting Plane*, é usada uma sequência de resoluções de subproblemas lineares.

2.4.1 Ferramentas usadas pelo SCP

O SCP (STILL; WESTERLUND, 2010) pode ser usado para resolver problemas não lineares, com restrições tanto de igualdade, quanto de desigualdade, o que, por exemplo, o LANCELOT (2.4.2.1) não pode resolver, já que requer que as restrições de desigualdade sejam convertidas em igualdades. Outro ponto importante é a convergência global a um ótimo KKT (2.4.1.1), mesmo em problemas não convexos, desde que sejam continuamente diferenciáveis.

2.4.1.1 Condições de Karush-Kuhn-Tucker

As condições de Karush-Kuhn-Tucker (KKT) são um conjunto de condições necessárias para que uma solução seja considerada um ótimo local para um dado problema. Esse conjunto de condições são usadas como peça fundamental para a construção de diversos métodos mais recentes.

Achava-se que as condições KKT foram propostas apenas em 1951 por Harold W. Kuhn e Albert W. Tucker (KUHN; TUCKER, 1951), mas mais tarde descobriu-se que já foram apresentadas em 1939 por William Karush (KARUSH, 1939), em sua tese de mestrado.

Um ponto KKT é aquele que passa por todas as condições de KKT, com testes de primeira e segunda ordem. É usada a forma Lagrangiana do seguinte problema:

$$\min_x f(x) \quad (29a)$$

$$\text{s.a } h_i(x) = 0, \quad i = 1, \dots, m; \quad (29b)$$

$$g_j(x) \leq 0, \quad j = 1, \dots, r. \quad (29c)$$

Onde a Lagrangiana é dada por:

$$L(x, \mu, \lambda) = f(x) + \lambda^T g(x) + \mu^T h(x) \quad (30)$$

As condições de primeira ordem para um x^* ser solução são as seguintes:

1. Estacionário:

A derivada de primeira ordem (ou gradiente) deve ser nula, o que caracteriza um ponto extremo da função. Mas também deve ser levada em consideração informações de primeira ordem das restrições, quando não cumpridas.

$$\nabla f(x^*) + \sum_{j=1}^r \lambda_j \nabla g_j(x^*) + \sum_{i=1}^m \mu_i \nabla h_i(x^*) = \vec{0} \quad (31)$$

2. Viabilidade primária:

Simplesmente considera que as restrições são satisfeitas.

$$\begin{aligned} g_j(x) &\leq 0, \quad j = 1, \dots, r; \\ h_i(x) &= 0, \quad i = 1, \dots, m. \end{aligned} \quad (32)$$

3. Negligência complementar:

Quando buscado um ótimo para a Lagrangiana, queremos desconsiderar o máximo possível das restrições a fim de otimizar a função. Então, em um ponto ótimo, anulamos as restrições e otimizamos apenas f . Como a solução é um ponto viável $h_i(x)$ é nulo, então não é necessário se preocupar nesta condição.

$$\lambda_j g_j(x) = 0, \quad j = 1, \dots, r; \quad (33)$$

4. Dupla Viabilidade:

Esta condição garante a existência da solução do problema dual (2.6).

$$\lambda_j \geq 0, \quad j = 1, \dots, r. \quad (34)$$

2.4.1.2 Qualificações de Mangasarian e Fromovitz

As qualificações de restrições de Mangasarian e Fromovitz são um trabalho em cima de um conjunto de qualificações "rival" das de Karush-Kuhn-Tucker, as qualificações de Fritz John.

As qualificações de restrições de Fritz John, (JOHN, 2014) originalmente foram construídas pensando apenas nas restrições de desigualdades. Existindo restrições de igualdades, elas devem ser convertidas em restrições de desigualdades, mas isso acaba tornando as qualificações inúteis, uma vez que qualquer ponto viável como solução satisfaz as restrições. Considerando o seguinte problema:

$$\min_x f(x) \quad (35a)$$

$$\text{s.a} \quad g_i(x) \leq 0, \quad i \in M := \{1, 2, \dots, m\}; \quad (35b)$$

$$h_j(x) = 0, \quad j \in K := \{1, 2, \dots, k\}. \quad (35c)$$

Considerando o conjunto K vazio, as condições de otimalidade de Fritz John são as seguintes:

Se $\bar{x}$ é solução do problema, então existe um vetor $u = (u_0, u_1, \dots, u_m) \in R^{m+1}$ de forma que:

$$u_0 \nabla f(x) + \sum_{i=1}^m u_i \nabla g_i(x) = 0; \quad (36)$$

$$\sum_{i=1}^m u_i \nabla g_i(x) = 0; \quad (37)$$

$$u_i > 0, i = 0, \dots, m. \quad (38)$$

No entanto, o conjunto de restrições de igualdade nem sempre é vazio. A saída mais fácil seria substituir cada restrição de igualdade por duas de desigualdade, mas isso tornaria as condições de otimalidade possíveis para qualquer valor viável. Então usar como uma qualificação de restrição pode não ser a melhor opção.

Mesmo existindo outras condições de otimalidade e qualificações para restrições, normalmente as restrições de desigualdades e igualdades eram analisadas separadamente. Com isso, Mangasarian e Fromovitz ([MANGASARIAN; FROMOVITZ, 1967](#)) melhoraram as condições de Fritz John para a análise conjunta das restrições. Tomando como componente principal o Teorema da Transposição de Motzkin ([NEMIROVSKI; ROOS, 2001](#)), generalizaram as condições para o seguinte:

Seja $\bar{x}$ solução do problema, e por isso, cumpre todas as restrições, então podemos dizer que $\bar{x}$ pertence a um conjunto solução.

$$\bar{x} \in S := \{x \mid g_i(x) \leq 0, i \in M, h_j(x) = 0, j \in K\} \quad (39)$$

Para que as condições sejam satisfeitas, $\nabla h_j(\bar{x}), j \in K$ são linearmente dependentes entre si, ou o sistema

$$\begin{aligned} \theta(x) - \theta(\bar{x}) &< 0, \quad \bar{x} \in S; \\ g_i(x) &< 0, \quad i \in M; \\ h_j(x) &= 0, \quad j \in K. \end{aligned} \quad (40)$$

não tem solução para $x \in \mathbb{R}^n$.

A partir daí, construíram também qualificações para as restrições, em que $\nabla h_j(\bar{x}), j \in K$ são linearmente independentes e existe um vetor $\bar{y}$ tal que:

$$\bar{y}' \nabla g_i(\bar{x}) < 0, i \in \bar{M} := \{i \mid i \in M, g_i(\bar{x}) = 0\} \neq \emptyset \quad (41)$$

$$\bar{y}' \nabla h_j(\bar{x}) = 0, j \in K \quad (42)$$

onde $\bar{y}'$ é $\bar{y}$ transposta. Caso $\bar{M}$ seja vazio, basta que os gradientes das restrições de igualdade sejam linearmente independentes.

2.4.2 Sobre a escolha do uso de problemas lineares

Como mostrado em um dos métodos anteriores em [2.3](#), a resolução de subproblemas lineares podem ser usadas tanto isoladamente, quanto em conjunto com a resolução de subproblemas quadráticos para resolver problemas de otimização não linear. Os métodos que usam os subproblemas quadráticos ganharam mais atenção, principalmente, depois que foi mostrada a convergência a um ótimo global em 1977 ([HAN, 1977](#)), e com o tempo, as técnicas que usam resolução de subproblemas lineares começaram a ser esquecidas. Nesse contexto, os autores, C. Still e T. Westerlund desejam mostrar que o método montado por eles é capaz de resolver problemas não lineares eficientemente apenas usando técnicas lineares.

Segundo seus experimentos, mostraram que o seu método é capaz de resolver problemas em um dado conjunto de problema tão eficientemente quanto outros métodos amplamente

utilizados, métodos estes que usam resolução de subproblemas quadráticos. Os resolvidores que usaram foi o LANCELOT (CONN; GOULD; TOINT, 1991) e o DONLP2 (SPELLUCCI, 1999). A escolha desses resolvidores pelos autores se deu por terem conceitos diferentes, mas ainda assim comparáveis ao conceito do SCP. O conjunto de problemas que todos os 3 resolveram foram os testes de Hock e Schittkowski (HOCK; SCHITTKOWSKI, 1981), e também os teste de Schittkowski (SCHITTKOWSKI, 1987).

Os dois conjuntos de testes usados podem ser considerados clássicos uma vez que são usados em diversas áreas da otimização e em artigos e livros relevantes. Dentre os exemplos de aplicação podemos citar testes em algoritmos genéticos (DEB, 2000; JOINES; HOUCK, 1994), ambientes de teste (BONGARTZ et al., 1995) e meta-heurísticas (GANDOMI et al., 2012).

2.4.2.1 LANCELOT

O LANCELOT (CONN; GOULD; TOINT, 1991) é na verdade um conjunto de técnicas compiladas em um único pacote da linguagem Fortran. Dentre as técnicas utilizadas, a técnica escolhida para fazer o comparativo com o SCP foi um método capaz de convergir para um ótimo global fazendo o uso da função Lagrangiana melhorada, assumindo restrições genéricas em um espaço simples. O problema que esse método se propõem a resolver é dado como:

$$\min_x f(x) \quad (43a)$$

$$\text{s.a} \quad c_i(x) = 0, \quad i \in \{1, \dots, m\} \quad (43b)$$

$$l \leq x \leq u \quad (43c)$$

É necessário que todas as funções sejam diferenciáveis, e que suas derivadas também sejam, ou então, sejam duas vezes continuamente diferenciáveis. Considera-se também que as restrições de desigualdades foram convertidas em restrições de igualdades. O ponto chave desse método é minimizar a função Lagrangiana melhorada:

$$\Phi(x, \lambda, S, \mu) = f(x) + \sum_{i=1}^m \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i=1}^m S_{ii} c_i(x)^2 \quad (44)$$

As duas primeiras parcelas são da função Lagrangiana original. A última parcela é um termo de penalidade. O μ define o peso dessa penalidade, a matriz S é diagonal e todas as entradas positivas (o que a torna definida positiva e simétrica), onde cada entrada da diagonal principal escala o quadrado da restrição correspondente. Portanto, a Lagrangiana melhorada considera como penalidade o quadrado das restrições escalados de acordo com a necessidade.

Fazendo o uso dos avanços em otimização desenvolvidos na época, criaram esse método a partir de outros estudos feitos anteriormente, mas agora com o suporte da Lagrangiana melhorada.

A Lagrangiana melhorada foi proposta de forma independente por (HESTENES, 1969) e (POWELL, 1969). O interesse nela foi sendo perdido à medida que os métodos de otimização

sucessiva quadrática (SQP) foi se tornando popular. Alguns autores ousaram e usaram o SQP juntamente com a Lagrangiana melhorada, como em (SCHITTKOWSKI, 1982) e em (GILL et al., 1986). Esses métodos não são "puros", mas acabam fazendo uso da Lagrangiana melhorada apenas para obter algumas informações.

Uma grande desvantagem dos métodos SQP para problemas de grande proporção é o custo computacional envolvido na resolução de um problema quadrático no final de cada iteração, da mesma forma que o método de Newton é custoso para problemas de grande proporção irrestrito. Essa foi uma das motivações para o desenvolvimento do LANCELOT.

Foram apresentados por Schittkowski (SCHITTKOWSKI, 1982) resultados de um estudo comparativo sobre métodos não lineares. A conclusão chegada por ele é que, mesmo com os problemas já apresentados, os métodos SQP pertencem à classe métodos mais eficientes e eficazes que se tinham acesso na época. Um dos primeiros resultados apresentados por Schittkowski é a substituição de uma função de penalidade posta por Han e Powell pela função Lagrangiana melhorada para fazer a busca em linha. Outro resultado de seus estudos mostrados foi a substituição de subproblemas quadráticos por subproblemas de quadrados mínimos, que são menos custosos de computar.

O foco do trabalho de Gill (GILL et al., 1986) consiste na análise do uso da Lagrangiana melhorada como uma função de mérito para um método SQP, seguindo a sugestão de Schittkowski. Com isso foi possível provar que a convergência se dá para um ótimo global e que é do tipo superlinear.

2.4.2.2 DONLP2

O DONLP2 é a implementação de uma variante de um método SQP capaz de resolver problemas abrangentes, da seguinte forma:

$$\min_x f(x) \quad (45a)$$

$$\text{s.a } h(x) = 0 \quad (45b)$$

$$g(x) \geq 0 \quad (45c)$$

Onde todas as funções são pelo menos duas vezes continuamente diferenciáveis, assim como no LANCELOT. A função de mérito determinada por esse método especifica o tamanho do passo a ser tomado, e esta função é uma função de penalidade exata, a l_1 (HAN; MANGASARIAN, 1979), usada nos estudos anteriormente citados de Schittkowski, no qual ele substitui a l_1 pela Lagrangiana melhorada.

2.5 Método do gradiente conjugado

Dentre os métodos mais simples podemos citar o método do gradiente descendente ou o próprio método de Newton, mas que possuem seus problemas, e por isso, a necessidade de métodos

um pouco mais elaborados, mas ainda assim não tão distantes da simplicidade empregadas nos outros já citados.

O método do gradiente conjugado se enquadra nessa configuração. Embora um pouco difícil de construir o método, o seu uso não é tão mais difícil que o do gradiente descendente.

Deve-se iniciar compreendendo o que é e os problemas do gradiente descendente afim para que se possa considerar as resoluções trazidas pelo método do gradiente conjugado.

2.5.1 Método do gradiente descendente

O método do gradiente descendente é um dos métodos iterativos mais simples que podemos encontrar. Ele consiste em mover o ponto da iteração atual de acordo com a direção de declive mais acentuada, ou seja, a direção do gradiente negativo. Considerando que queremos minimizar uma função $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, o passo de atualização de um ponto $x_k \in \mathbb{R}^n$ é dada por:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad (46)$$

α_k nada mais é que um escalar que controla o tamanho da atualização do ponto atual x_k na direção de descida. Existem várias possibilidades para α_k , podemos escolher um valor constante, ou fazer uma estimativa de um bom valor, ou ainda minimizar também esse valor afim de obter o tamanho ótimo do passo:

$$\min_{\alpha_k} f(x_k - \alpha_k \nabla f(x_k)) \quad (47)$$

O gradiente descendente, por mais simples e prático que seja, tem seu preço. Se pararmos para observar seu comportamento, percebemos um esforço gasto à toa. Analisando o movimento do ponto atualizado vemos um "zig-zag", isso quer dizer que a cada iteração, estamos desfazendo um pouco do progresso feito em uma iteração anterior. Além disso, o gradiente negativo não aponta numa direção para o mínimo, mas sim nos dá uma direção de minimização imediata em relação ao ponto atual. Se considerarmos minimizar uma função quadrática convexa com seções elípticas com alta excentricidade usando o gradiente descendente, depois de muitas iterações teremos feito muito pouco progresso, visto que o gradiente negativo vai apenas "rebater" o ponto entre os limites da elipse que são mais "estreitos".

Uma forma de resolver ambos os problemas, é, no lugar de considerar a direção apenas o gradiente negativo, fazendo com que o gradiente seja ajustado a cada iteração para ser ortogonal às direções previamente encontradas, o que impede a perda de progresso pelo comportamento de "zig-zag". Por serem ortogonais, eles formam um span (base) no $\mathbb{R}^n$ possibilitando que a solução seja escrita como uma combinação linear desse span, permitindo a cada iteração o progresso em apenas uma direção.

2.5.2 Construindo o método do gradiente conjugado

Uma das motivações do método do gradiente conjugado (BONNANS et al., 2006) (POLYAK, 1987) é resolver um sistema linear de n equações na forma $Ax = b$. Os métodos convencionais para esse tipo de problema falham no sentido de que, para um n muito grande, é inviável ter memória para ser guardada uma segunda matriz de mesmo tamanho, que, caso esparsa, gera um gasto de memória grande demais para pouca informação.

Considerando A uma matriz simétrica e definida positiva, podemos definir um produto interno com respeito à ela:

$$\langle u, v \rangle_A = \langle u, Av \rangle = \langle Au, v \rangle \quad (48)$$

Onde dizemos que os vetores u e v são conjugados em respeito a A (ou A-ortogonais) se:

$$\langle u, v \rangle_A = 0 \quad (49)$$

A partir daí podemos escolher um conjunto de n vetores que formam um span do $\mathbb{R}^n$:

$$U := \{d_1, d_2, \dots, d_n\} \quad (50)$$

$$\langle d_i, d_j \rangle_A = 0; i \neq j \quad (51)$$

Podemos escrever qualquer vetor $v \in \mathbb{R}^n$ como uma combinação linear dos vetores que formar U :

$$v = \sum_{i=1}^n \alpha_i d_i \quad (52)$$

Ou seja, podemos escrever a solução de $Ax = b$ dessa forma. Chamemos a solução de x^* . Temos então:

$$x^* = \sum_{i=1}^n \alpha_i d_i \quad (53)$$

Multiplicando A pela esquerda nos dois lados e depois multiplicando por algum d_k :

$$Ax^* = \sum_{i=1}^n \alpha_i Ad_i \Rightarrow d_k Ax^* = \sum_{i=1}^n \alpha_i d_k Ad_i \quad (54)$$

Substituindo Ax^* por um vetor b e explicitando o produto interno no lado direito:

$$\langle d_k, b \rangle = \sum_{i=1}^n \alpha_i d_k Ad_i \Rightarrow \langle d_k, b \rangle = \sum_{i=1}^n \alpha_i \langle d_k, d_i \rangle_A \quad (55)$$

Considerando (51), apenas a parcela $\langle d_k, d_k \rangle_A$ não é nula, e rearranjando para α_k :

$$\langle d_k, b \rangle = \alpha_k \langle d_k, d_k \rangle_A \Rightarrow \frac{\langle d_k, b \rangle}{\langle d_k, d_k \rangle_A} = \alpha_k \quad (56)$$

Logo, para calcular a solução diretamente, basta encontrar o conjunto U .

Antes de podermos encontrar o tal conjunto, primeiro vamos observar uma outra forma de interpretar o problema. Então, podemos escrever um problema do tipo $Ax = b$ como um problema de otimização de uma função quadrática convexa:

$$\min_x f(x) = \frac{1}{2} \langle x, x \rangle_A - \langle b, x \rangle \quad (57)$$

Como a função é quadrática e convexa, sabemos que possui apenas um único ponto de extremo x^* e que é o mínimo. A partir dessas informações podemos definir as seguintes expressões:

$$\nabla f(x^*) = 0 \quad (58)$$

Para termos $\nabla f(x)$, podemos simplificar o processo de derivação de $f(x)$ ao imaginar que $\langle x, x \rangle_A$ é de alguma forma equivalente à x^2 , e $\langle x, b \rangle$ ao produto de uma constante e a variável real. Portanto:

$$\nabla f(x) = Ax - b \quad (59)$$

E da mesma forma:

$$\nabla^2 f(x) = A \quad (60)$$

Com isso podemos usar o mesmo método para encontrar soluções para $Ax = b$ para minimizar funções cuja matriz hessiana ($\nabla^2 f(x)$), que chamaremos sempre de A , seja definida positiva.

Podemos gerar o conjunto U a partir dos gradientes, ajustando-os para que sejam mutuamente conjugados. Escolhendo dessa forma, pode não ser necessário ter-se todos os vetores que formam U , já que como a combinação linear de alguns poucos gradientes encontrados durante as iterações do gradiente descendente nos dão boas aproximações da solução, é esperado que esses vetores de descida corrigidos deem ainda uma melhor solução.

Começemos fazendo com que $d_1 = -\nabla f(x_1) = -g_1$, já que isso pode nos por na direção correta do mínimo. O algoritmo, sendo iterativo, tem uma forma semelhante ao gradiente descendente, mudando apenas a atualização:

$$x_{k+1} = x_k + \alpha_k d_k \quad (61)$$

Onde $d_k \in U$ e α_k é o tamanho do passo, que pode ser exato minimizando f na direção d_k .

Como queremos direções conjugadas, que por sua vez são geradas a partir dos gradientes, então o gradiente no próximo ponto já pode ser conjugado em relação à direção atual. Considerando (61), podemos dizer que:

$$\nabla f(x_{k+1}) = g_{k+1} = g_k + \alpha_k Ad_k \quad (62)$$

Com isso, queremos que:

$$\langle g_{k+1}, d_k \rangle = 0 \Rightarrow \langle g_k + \alpha_k Ad_k, d_k \rangle = 0 \quad (63)$$

Considerando as propriedades inerentes a um produto interno:

$$\langle g_k, d_k \rangle + \alpha_k \langle Ad_k, d_k \rangle = 0 \Rightarrow \alpha_k \langle Ad_k, d_k \rangle = -\langle g_k, d_k \rangle \quad (64)$$

Por fim, resolvendo para α_k e mudando a notação:

$$\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle Ad_k, d_k \rangle} \Rightarrow \alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A} \quad (65)$$

O que nos falta agora é a atualização de d_k . Como dito anteriormente, queremos construir essas direções a partir do gradiente negativo corrigido para que seja A-ortogonal em relação às direções já encontradas.

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (66)$$

Portando, β_k deve ser escolhido de forma que $\langle d_{k+1}, d_k \rangle_A = 0$:

$$\langle -g_{k+1} + \beta_k d_k, d_k \rangle_A = 0 \Rightarrow \langle -g_{k+1}, d_k \rangle_A + \beta_k \langle d_k, d_k \rangle_A = 0 \quad (67)$$

Rearranjando e resolvendo para β_k :

$$\beta_k \langle d_k, d_k \rangle_A = -\langle -g_{k+1}, d_k \rangle_A \Rightarrow \beta_k = -\frac{\langle -g_{k+1}, d_k \rangle_A}{\langle d_k, d_k \rangle_A} \quad (68)$$

Já temos o algoritmo pronto, mas ainda é custoso. Com mais um pouco de trabalho, podemos remover duas multiplicações por A de β_k :

$$\beta_k = -\frac{\langle -g_{k+1}, Ad_k \rangle}{\langle d_k, Ad_k \rangle} \quad (69)$$

Queremos substituir Ad_k por algo mais simples, para isso podemos usar (62):

$$g_{k+1} = g_k + \alpha_k Ad_k \Rightarrow g_{k+1} - g_k = \alpha_k Ad_k \Rightarrow Ad_k = \frac{g_{k+1} - g_k}{\alpha_k} \quad (70)$$

Como β_k é definido por uma razão, podemos trabalhar o numerador e o denominador separadamente. Então comecemos trabalhando apenas o numerador de β_k :

$$\langle -g_{k+1}, Ad_k \rangle \quad (71)$$

Substituindo Ad_k pela definição em (70), e extraíndo os coeficientes:

$$\langle -g_{k+1}, \frac{g_{k+1} - g_k}{\alpha_k} \rangle \Rightarrow -\frac{1}{\alpha_k} \langle g_{k+1}, g_{k+1} - g_k \rangle \quad (72)$$

E pela linearidade do produto interno:

$$-\frac{1}{\alpha_k} (\langle g_{k+1}, g_{k+1} \rangle + \langle -g_{k+1}, -g_k \rangle) \quad (73)$$

Sabemos, porque construímos assim, que $\langle g_{k+1}, g_k \rangle = 0$. Por fim, temos que o numerador de β_k é dado por:

$$-\frac{1}{\alpha_k} \langle g_{k+1}, g_{k+1} \rangle \quad (74)$$

Agora trabalhando o denominador de β_k :

$$\langle d_k, Ad_k \rangle \quad (75)$$

Novamente, usando (70), e extraindo o coeficiente:

$$\langle d_k, \frac{g_{k+1} - g_k}{\alpha_k} \rangle \Rightarrow \frac{1}{\alpha_k} \langle d_k, g_{k+1} - g_k \rangle \quad (76)$$

Pela linearidade do produto interno e usando (63):

$$\frac{1}{\alpha_k} (\langle d_k, g_{k+1} \rangle + \langle d_k, -g_k \rangle) \Rightarrow \frac{1}{\alpha_k} \langle d_k, -g_k \rangle$$

Considerando (66) para substituir d_k :

$$\frac{1}{\alpha_k} \langle -g_k + \beta_{k-1} d_{k-1}, -g_k \rangle \quad (77)$$

Aplicando a linearidade e extraindo coeficientes:

$$\frac{1}{\alpha_k} (\langle -g_k, -g_k \rangle + \langle \beta_{k-1} d_{k-1}, -g_k \rangle) \Rightarrow \frac{1}{\alpha_k} (\langle -g_k, -g_k \rangle - \beta_{k-1} (\langle d_{k-1}, g_k \rangle)) \quad (78)$$

Por fim, considerando (63), temos o denominador dado por:

$$\frac{1}{\alpha_k} \langle -g_k, -g_k \rangle \quad (79)$$

Finalmente, podemos reescrever β_k . Considerando (69), (74) e (79), e substituindo:

$$\beta_k = -\frac{\langle -g_{k+1}, Ad_k \rangle}{\langle d_k, Ad_k \rangle} \Rightarrow \beta_k = -\frac{-\frac{1}{\alpha_k} \langle g_{k+1}, g_{k+1} \rangle}{\frac{1}{\alpha_k} \langle -g_k, -g_k \rangle} \quad (80)$$

Isolando e resolvendo o coeficiente:

$$\beta_k = -\frac{-\frac{1}{\alpha_k}}{\frac{1}{\alpha_k}} \cdot \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle -g_k, -g_k \rangle} \Rightarrow \beta_k = -(-1) \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle -g_k, -g_k \rangle} \quad (81)$$

Por fim, temos a última peça para a definição do algoritmo do gradiente conjugado:

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle -g_k, -g_k \rangle} \quad (82)$$

Em resumo, iniciando com x_1 sendo uma estimativa do ótimo e a direção inicial sendo $d_1 = -\nabla f(x_1) = -g_1$, temos que o algoritmo é:

$$x_{k+1} = x_k + \alpha_k d_k \quad (83)$$

$$\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A} \quad (84)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (85)$$

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle -g_k, -g_k \rangle} \quad (86)$$

Podemos ver que o método não é complexo de ser usado, nem tão custoso computacionalmente em casos simples. Em problemas de maior dimensão, funções custosas ou ainda problemas difíceis que não sejam quadráticos, o método pode não ser o recomendado pelo fato de considerar várias avaliações da função para uma única iteração.

2.6 Problemas lineares duais

Qualquer problema linear possui um outro problema atrelado a ele, o problema dual (FISCHETTI, 2019). Considerando a ideia de que queremos minimizar uma quantidade na reta real e, por simplicidade, assumimos que este mínimo existe. Podemos dizer então que este valor mínimo também é um valor máximo para todas as quantidades menores que aquele mínimo. Com isso, se tivermos a capacidade de gerar um problema de maximização onde o máximo desse problema seja o mínimo do que queremos minimizar inicialmente, e se o que for importante de fato seja o valor, estamos feitos. A efeito de exemplo, podemos usar um exemplo clássico de um fazendeiro e o comprador de seus produtos, onde o primeiro quer maximizar seus lucros, e o segundo quer minimizar seus custos. É certo que ambas as partes querem fechar um acordo, e, para isso, o mínimo do comprador deve ser o máximo do fazendeiro.

Considerando o seguinte problema linear de minimização, que chamaremos de problema primal:

$$\min_x c^T x \quad (87a)$$

$$\text{s.a } x \in \mathbb{R}^n \quad (87b)$$

$$x \in P := \{Ax = b, x \geq \vec{0}\} \quad (87c)$$

$$b \in \mathbb{R}^m \quad (87d)$$

$$A \in \mathbb{R}^{m \times n} \quad (87e)$$

Podemos escrever o problema de achar um máximo de outro problema, que chamaremos de dual, como:

$$\min_x \{c^T x \mid x \in P\} = \max_{c_0} \{c_0 \mid c^T x \geq c_0 \forall x \in P\} \quad (88)$$

Com isso, já podemos perceber alguns fatos sobre ambos os problemas. Se considerarmos que o problema primal é irrestrito, ou seja, se $\min_x \{c^T x \mid x \in P\} = -\infty$, então não seremos capazes de encontrar um c_0 que satisfaça. Portanto, se o problema primal for irrestrito, então o problema dual é inalcançável. O caminho oposto também é verdade, se considerarmos o problema dual irrestrito, então $c_0 = \infty$, logo não existe um mínimo para o problema primal.

Como queremos encontrar c_0 de forma que $c^T x \geq c_0$, a partir dos sistemas de equações formados pelas restrições podemos construir expressões limitantes para $c^T x$. Novamente, considerando o mesmo problema (87), podemos multiplicar todo o sistema de equações por escalares $u \in \mathbb{R}^m$:

$$u^T Ax = u^T b \quad (89)$$

Com essa nova expressão, podemos modificar algumas características sem a tornar inválida. Podemos substituir $=$ por $\geq$.

$$u^T Ax \geq u^T b \quad (90)$$

Feito isso, podemos reduzir o valor de $u^T b$ o quanto quisermos, ou seja, sempre existe um $c_0 \leq u^T b$. Como $x \geq \vec{0}$, podemos aumentar os valores dos coeficientes de x , fazendo com que o valor aumente apenas na esquerda, mas a expressão continue válida.

Sabendo que $u^T A$ são coeficientes para x , podemos limita-los superiormente pelos coeficientes do problema original: $c^T \geq u^T A$. Por fim, temos as seguintes três expressões, todas válidas:

$$u^T Ax = u^T b \quad (91)$$

$$c^T x \geq u^T Ax \quad (92)$$

$$u^T b \geq c_0 \quad (93)$$

Que quando combinadas:

$$c^T x \geq u^T Ax = u^T b \geq c_0 \quad (94)$$

Com isso podemos montar um outro problema, encontrar o valor máximo de u de forma que $u^T b$ nos sirva como o limite inferior c_0 que buscamos, mas para isso temos que $c^T \geq u^T A$.

$$\max_u \{u^T b \mid u \in \{c^T \geq u^T A\}\} \quad (95)$$

Assim temos nosso problema dual, que satisfaz a condição que o mínimo do primal é o máximo do dual pois, por construção, no momento em que multiplicamos o sistema $Ax = b$ por u^T , vemos que existem soluções para $u^T A = c^T$. Portanto podemos dizer que:

$$c^T x = u^T Ax \Rightarrow c^T x = u^T b \geq c_0 \quad (96)$$

Mas como estamos considerando $u^T b = c_0$:

$$c^T x = u^T b = c_0 \Rightarrow c^T x = c_0 \quad (97)$$

Então mostramos que:

$$\min_x \{c^T x \mid x \in P\} = \max_{c_0} \{c_0 \mid c^T x \geq c_0 \forall x \in P\} = \max_u \{u^T b \mid u \in \{c^T \geq u^T A\}\} \quad (98)$$

Que é o mesmo que dizer que:

$$\min_x = \max_{c_0} = \max_u \Rightarrow c_0 = u^T b \quad (99)$$

Então temos mostrado que a solução do problema primal e dual são iguais (para o caso linear). Esse fato se torna útil verificação a resolução dos problemas em aplicações praticas. Outra utilidade é que como a solução do problema dual relaciona todas as restrições à solução do problema primal, isso nos diz algo sobre os multiplicadores de Lagrange, em alguns casos são usados como estimativas dos multiplicadores.

2.7 Cálculos numéricos da Hessiana

O cálculo das informações de segunda ordem de uma função, a matriz Hessiana, faz parte do conjunto de problemas computacionais encontrados durante o processo de otimização. Diversos métodos são formulados com a necessidade do uso da Hessiana.

O método de Newton, assim como toda a família de métodos usados como alternativas a ele, denominados de métodos Quasi-Newton, utilizam a matriz Hessiana durante seu processo de otimização de forma conjunta ou não com o gradiente da função. Como a família de métodos newtonianos é extensa e aplicada em diversas áreas, diversos métodos para o cálculo da Hessiana foram desenvolvidos pensando na eficiência de sua aplicação. Dentre os diversos métodos, alguns se tornaram famosos e amplamente utilizados, como o BFGS e o DFP, e outros mais específicos como métodos adjuntos, métodos complexos e o método hiper-dual.

É necessário ter o conhecimento que nem todos os métodos de otimização, que usam a Hessiana de alguma forma, requerem a computação exata da Hessiana, então existe uma divisão entre métodos de computação exata da Hessiana, e métodos que computam apenas aproximações da Hessiana.

2.7.1 Diferença finita

Os métodos que usam diferença finita podem ser vistos como traduções das definições matemáticas das derivadas. No entanto, quando esses métodos são implementados praticamente, problemas com a precisão dos números obtidos se tornam um problema, assim como o custo de avaliar diversas vezes a função objetivo (CAPLAN, 2011).

Existem duas formas normalmente apresentadas para o uso da diferença finita, a diferença progressiva e a diferença central. Que para o cálculo de derivadas de primeira ordem, temos:

$$\frac{\partial f}{\partial x_i} = \frac{f(x + he_i) - f(x)}{h} \quad (100)$$

$$\frac{\partial f}{\partial x_i} = \frac{f(x + he_i) - f(x - he_i)}{2h} \quad (101)$$

onde e_k é a k -ésima base canônica do $\mathbb{R}^n$, (100) é o cálculo da derivada por diferença progressiva e (101) por diferença central.

É preferível escolher a diferença central principalmente por causa da impossibilidade de cálculo causada pela incapacidade de operar com um valor de h infinitesimalmente pequeno, já quando se calcula a derivada considerando a diferença $f(x + he_i) - f(x)$, acabamos calculando a derivada em um ponto um pouco mais à frente do que o desejado, gerando um erro significativo. Já usando a diferença central, contorna-se isso equilibrando a posição onde a derivada será calculada pela mesma proporção.

Mas ainda é necessário tomar um certo cuidado no uso da diferença central no que se diz respeito a escolha de h . Como estamos tomando uma diferença maior entre os valores observados, devemos compensar esse aumento na diferença escolhendo um valor menor para h do que escolheríamos se usássemos a diferença progressiva.

O cálculo da derivada de segunda ordem também é possível por esses mesmos métodos. Para o uso da diferença progressiva, (DENNIS; SCHNABEL, 1996) mostram e provam a seguinte expressão:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{f(x + he_i + he_j) - f(x + he_i) - f(x + he_j) + f(x)}{h^2} \quad (102)$$

Percebemos logo seu custo de 4 avaliações da função por entrada da matriz Hessiana, o que pode se tornar um problema considerando um problema de alta dimensionalidade ou funções difíceis de se avaliar, mesmo sabendo que a matriz Hessiana é simétrica, ou seja, basta computar apenas um triângulo da matriz.

Já para o uso da diferença central, descrita em (ABRAMOWITZ; STEGUN, 1972), temos a seguinte expressão:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{f(x + he_i + he_j) - f(x + he_i - he_j) - f(x - he_i + he_j) + f(x - he_i - he_j)}{4h^2} \quad (103)$$

Da mesma forma, temos 4 avaliações da função, porém com mais operações sobre o valor de entrada, e considerando o mesmo problema de ter que escolher um valor menor para h . E também o uso da diferença central tem uma precisão melhor que a diferença progressiva.

Como se pode perceber, o uso de diferença finita é a forma mais direta e mais fácil de se computar derivações, no entanto, como já foi considerado, o erro é algo que pode tornar impraticável. No caso da otimização, a precisão da solução é diretamente relacionada à precisão oferecida, ou ainda, condições de parada de métodos (2.4.1.1).

2.7.2 DFP e BFGS

Quando o cálculo da matriz Hessiana se torna custoso demais para se calcular o valor exato, métodos de atualização da Hessiana tomam seus lugares. Esses métodos normalmente constroem essa atualização baseando-se na característica iterativa dos algoritmos que estão envolvidos.

Podemos iniciar tomando uma aproximação da expansão local do gradiente da próxima iteração:

$$\nabla f(x_{k+1}) \approx \nabla f(x) + H(x)(x_{k+1} - x) \quad (104)$$

Que reorganizando e tomando a Hessiana como constante (o que não é, mas é esperado que o gradiente não mude bruscamente) temos a seguinte expressão, gerando uma aproximação para a Hessiana em x_k e x_{k+1} :

$$H \approx \frac{\nabla f(x_{k+1}) - \nabla f(x)}{(x_{k+1} - x)} \quad (105)$$

Ou ainda:

$$H\sigma \approx y \quad (106)$$

$$y = (\nabla f(x_{k+1}) - \nabla f(x)), \sigma = (x_{k+1} - x) \quad (107)$$

Que nada mais é que a equação da secante. O problema que temos agora, é que a Hessiana deve ser sempre simétrica e normalmente também definida positiva, o que não está garantido desta forma.

Em 1970, Davidon (DAVIDON, 1991), usou em seu método de minimização uma matriz que especificava um mapeamento linear das mudanças do gradiente em relação as mudanças da posição observada. Essa matriz é atualizada a cada iteração de seu método pelas relações do que foi mudado em relação à posição e no que foi mudado em relação ao gradiente da posição anterior e o atual. No entanto, seu método nem sempre era usável, e por isso em 1963, Fletcher e Powell (FLETCHER; POWELL, 1963) introduziram novas ferramentas, garantindo o funcionamento em diversos casos e a convergência dessa tal matriz à matriz Hessiana inversa (o que já acontecia antes, mas não em todos os casos), tornando o método da atualização

da matriz o primeiro a receber um grande prestígio. A atualização da aproximação da matriz inversa à Hessiana de Davidon-Fletcher-Powell (DFP) é dada por:

$$H_{k+1}^{-1} = H_k^{-1} + A_k + B_k \quad (108)$$

$$A_k = \frac{\sigma_k^T \sigma_k}{\sigma_k^T y_k} \quad (109)$$

$$B_k = -\frac{H_k^{-1} y_k y_k^T H_k^{-1}}{y_k^T H_k^{-1} y_k} \quad (110)$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad (111)$$

$$\sigma_k = (x_{k+1} - x_k) \quad (112)$$

σ_k também pode ser definido como o vetor que quando somado à x_k , minimiza a função na direção $-H_k^{-1} \nabla f(x_k)$.

Algumas informações podem ser verificadas a partir das expressões dadas, como por exemplo, escolhendo uma matriz inicial definida positiva, as seguintes iterações serão positivas definidas, e com as novas ferramentas trazidas por Fletcher e Powell, o método DFP também se torna um método conjugado, por algumas relações ortogonalizantes no processo.

Um dos passos tomados na definição do método DFP consiste em restringir a próxima aproximação da matriz inversa à Hessiana a ser definida positiva. Mas podemos restringir a própria aproximação da Hessiana a ser positiva definida e já calculá-la diretamente. Broyden (BROYDEN, 1970), Fletcher (FLETCHER, 1970), Goldfarb (GOLDFARB, 1970) e Shanno (SHANNO, 1970) chegaram a essa conclusão e demonstraram o método que ofuscaria o DFP, o BFGS, definido da seguinte forma:

$$H_{k+1} = H_k + A_k + B_k \quad (113)$$

$$A_k = \frac{y_k^T y_k}{y_k^T \sigma_k} \quad (114)$$

$$B_k = -\frac{H_k \sigma_k \sigma_k^T H_k}{\sigma_k^T H_k \sigma_k} \quad (115)$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad (116)$$

$$\sigma_k = x_{k+1} - x_k \quad (117)$$

O BFGS basicamente inverte as posições de variáveis em relação ao DFP. Na literatura encontra-se constantemente exemplos de que o BFGS é um dos melhores métodos disponíveis, tanto em eficiência quanto em facilidade de implementação.

2.7.3 Passo complexo

O método do passo complexo, mostrado por Lai ([LAI et al., 2005](#)), mas conhecido desde 1967 ([LYNESS; MOLER, 1967](#)), toma como base a expansão de Taylor da função perturbada por um termo complexo:

$$f(x + ih) = f(x) + \frac{i^1 h^1}{1!} f'(x) + \frac{i^2 h^2}{2!} f''(x) + \dots \quad (118)$$

Para a derivada de primeira ordem, podemos truncar as expressões para não considerar derivadas de maiores ordens. O que nos deixa com:

$$f(x + ih) = f(x) + ihf'(x) \quad (119)$$

Sabendo que $f(x + ih)$ é um número complexo, podemos pegar apenas a parte imaginária:

$$\text{Imag}(f(x + ih)) = hf'(x) \quad (120)$$

Por fim:

$$f'(x) = \frac{\text{Imag}(f(x + ih))}{h} \quad (121)$$

Da mesma forma, pode ser construído um método para se obter as segundas derivadas:

$$f''(x) = \frac{2}{h^2} (f(x) - \text{Re}(f(x + ih))) \quad (122)$$

Esse método nos dá uma grande vantagem em relação ao erro criado pelas operações de ponto flutuantes executadas em computadores. A desvantagem é a necessidade que todas as ferramentas usadas em conjunto com esse método sejam preparadas para trabalhar com funções de imagem complexa.

O erro em relação a qualquer método de diferença finita (comparáveis por serem simples de implementar) é consideravelmente menor para um h suficientemente pequeno ([CAPLAN, 2011](#)).

2.7.4 Números hiper-duais

O método do uso de números hiper-duais ([FIKE; ALONSO, 2011](#)) é baseado na ideia de generalização dos números complexos. Um número hiper-dual é definido como:

$$a = a_0 + a_i \epsilon_i + a_j \epsilon_j + x_{ij} \epsilon_i \epsilon_j \quad (123)$$

onde $\epsilon_i^2 = \epsilon_j^2 = (\epsilon_i \epsilon_j)^2 = 0$, mas $\epsilon_i, \epsilon_j, \epsilon_i \epsilon_j \neq 0$.

Quando fazemos a expansão de Taylor de uma função perturbada sobre tais números, temos apenas:

$$f(x+h) = f(x) + h_i \frac{df}{dx} \epsilon_i + h_j \frac{df}{dx} \epsilon_j + h_i h_j \frac{d^2 f}{dx^2} \epsilon_{ij} \quad (124)$$

Reorganizando, temos:

$$\frac{df}{dx} = \frac{1}{h_i} \epsilon_{if} = \frac{1}{h_j} \epsilon_{jf} \quad (125)$$

Onde ϵ_{kf} é a ϵ_k parte de $f(x + h_i \epsilon_i + h_j \epsilon_j)$

Para segundas derivadas:

$$\frac{d^2 f}{dx^2} = \frac{1}{h_{ij}} \epsilon_{ijf} \quad (126)$$

onde ϵ_{ijf} é a ϵ_{ij} parte de $f(x + h_i \epsilon_i + h_j \epsilon_j)$

Encontramos o mesmo problema que o método imaginário, a aplicação que se quer usar deve ter suporte para os números hiper-duais. No entanto, essa é uma das formas mais precisas e eficientes de se ter primeiras e segundas derivadas (FIKE; ALONSO, 2011).

3 Percurso Metodológico

Para a produção desse trabalho foi considerado o método analítico, inaugurado por Descartes. O método, de forma geral, considera a criação de conjecturas acerca do tema, e por meio de passos bem elaborados, chega à construção de uma tese.

A introdução ao tema foi feita no ano de 2019, junto ao início de um projeto de pesquisa realizado por um grupo de alunos do curso de Bacharelado da Ciência da Computação da UFAPE. No início do projeto foi feita uma exposição inicial sobre o que deveria ser feito, quais temas buscar e onde buscar. Após essa orientação inicial, eram feitas reuniões semanais onde eram apresentados os resultados de uma pesquisa superficial sobre as aplicações de métodos de otimização em situações práticas, como em controladores de máquinas de precisão e otimização de classificadores. Essas pesquisas semanais foram feitas a partir da leitura de artigos encontrados em buscas em bases científicas, as quais são acessíveis por intermédio do portal de periódicos da CAPES. Depois de feita uma quantidade que foi considerada suficiente dessas reuniões, cada aluno do grupo de pesquisa foi incumbido de fazer a escolha de um método dos que foram apresentados durante as reuniões. O método escolhido foi esse que resultou na produção desse trabalho.

Terminado o projeto de pesquisa e já escolhido o método, os esforços foram voltados para a confecção desse trabalho, iniciando uma nova fase de estudos sobre o SCP. Inicialmente foi feita uma leitura um pouco mais aprofundada, buscando conhecer, de forma simples, os fundamentos usados pelos autores para a construção do método, bem como compreender os problemas para os quais o método foi aplicado, a fim de ter o entendimento do meio em que ele foi concebido e também a identificação de métodos concorrentes em relação à aplicação, e métodos semelhantes em relação às ferramentas utilizadas para seu funcionamento. Em paralelo a isso, enquanto se buscava identificar os fundamentos do SCP foi sendo construído um conhecimento básico sobre o funcionamento do método, culminando em um seminário sobre o mesmo.

Quando houve a decisão de que o tema escolhido para este trabalho seria o estudo do SCP, foi feito um delineamento do que seria feito para a pesquisa. Além disso, também houve a decisão da implementação de um código para o algoritmo. Um dos motivos para essa implementação, além da criação de uma versão pública e acessível do algoritmo, é auxiliar no entendimento do funcionamento do método e de suas bases, uma vez que experimentos empíricos trazidos por essa implementação ajudam na compreensão e indicação de comportamentos do método.

Para a implementação do algoritmo foi escolhida uma linguagem de programação que cumpria alguns requisitos para que se pudesse ter ao final o que se esperava do código. Dentre os requisitos, temos que a linguagem deve ser capaz de produzir uma biblioteca dinâmica para que pudesse ser usada por qualquer outra linguagem com capacidade de utilizar uma biblioteca dinâmica, que a linguagem deve permitir o porte do código do algoritmo para qualquer outra plataforma suportada pelo compilador, que seja uma linguagem auto-explicativa no melhor que ela puder ser e que tenha recursos que facilitem a seu uso e a fácil produção de código de máquina seguro e eficiente. A linguagem escolhida para ser usada foi Rust, uma linguagem moderna, mas voltada para a construção de sistemas críticos, como bibliotecas para programas, sistemas operacionais e aplicações em embarcados. A linguagem cumpre todos os requisitos já citados, além de que já se existia um conhecimento prévio dela.

A construção do código foi feita em paralelo ao estudo das ferramentas usadas para elaborar o método. Uma vez terminado o estudo de uma das bases elencadas pelos autores do SCP, era implementado o código equivalente ao que foi estudado, apelando quando necessário para um estudo mais aprofundado. Desta forma, existia um vínculo entre o artigo onde foi apresentado o SCP, o estudo feito a respeito do que se foi empregado no artigo e a construção do código. Enquanto construído, o código passava por testes, os quais eram feitos para verificar a correção de cada parte. Para cada parte do código, eram usados problemas específicos para que fosse testado, tentando ao máximo garantir o funcionamento correto de cada parte.

3.1 Delineamento da Pesquisa

A definição do escopo do projeto desse trabalho foi feita ainda durante o projeto de pesquisa, bem como o tema a ser estudado. A escolha foi feita a partir do interesse criado sobre o método e sobre o que levou os autores a elaborarem o mesmo. Dada a complexidade do método é natural que se levantem discussões sobre o funcionamento e o que levou a sua construção, e por isso um interesse maior.

O objetivo desse trabalho é, em primeiro lugar, a compreensão do método SCP, seus fundamentos, utilidade e implicações de sua existência. Em segundo lugar, é ter como um código acessível, público e útil para os problemas que o SCP se propõe em resolver. Ao considerarmos que o método é de fato competitivo em relação aos outros métodos que temos hoje em dia, um código público e acessível do mesmo é de grande importância para a evolução da comunidade científica no que diz respeito aos estudos de métodos de otimização, uma vez que, por exemplo, testes numéricos podem ser realizados entre a implementação do SCP e outro método que venha a existir ou que já exista, produzindo um melhor entendimento a eficiência e capacidade de aplicação dos métodos. O projeto se torna viável quando é percebido que as ferramentas utilizadas para a construção do método, no que diz respeito às suas implementações, não são complexas a ponto de gerar algum impedimento. O que poderia gerar uma possível inviabilidade para a execução do projeto seria a tentativa de compreensão total dessas ferramentas. Mas ao considerar que o estudo dessas ferramentas é necessário até

um certo ponto, e não em sua completude, descartamos essa inviabilidade.

O cronograma considerado para a execução desse trabalho teve início após o término do projeto de pesquisa já mencionado. Iniciado em Janeiro de 2021, o primeiro passo do cronograma foi fazer uma releitura no artigo onde foi descrito o SCP, tentando listar o máximo de pontos citados para um estudo posterior. Essa releitura ditou a ordem em que seriam feitas as pesquisas sobre os pontos elencados, uma vez que essa foi a escolha dos autores para a apresentação do método. Feita essa releitura, iniciou-se apenas os estudos dos primeiros pontos inicialmente elencados, resultando em apresentações semanais sobre o que foi estudado. Ao chegar ao ponto em que o assunto coincidia com o primeiro passo a ser tomado para a construção do código, foi de fato iniciado a construção, a fim de manter um paralelismo entre o código que estava sendo construído e o assunto que estava sendo estudado, garantindo uma melhor extração de conhecimento por ambos os meios. E assim foi feito até o término dos estudos e a finalização do código em uma primeira versão.

Como o objetivo primário desse trabalho é gerar conhecimento sobre um método moderno e tudo o que isso representa, é possível considerar metas não alcançadas. Dentre elas, a garantia que ao final do projeto se tenha o código proposto completamente funcional, ou ainda, a contemplação de todos os assuntos elencados durante a releitura do artigo do SCP em igual profundidade, ou deixando algum de fora quando se viu algum outro com maior prioridade.

3.2 Coleta e Tratamento de Dados

A busca pelo material de estudo, como já dito, iniciou-se por uma releitura do artigo, onde foram extraídas referências pertinentes à outros artigos e livros. Com esse material em mãos, foi sendo feita uma busca em profundidade nas referências de cada tema estudado, afim de se chegar a uma origem concreta sobre o que se era estudado, bem como o percurso histórico. Todos os textos foram acessados por meio do portal de periódicos da CAPES.

Para cada texto visto, foi feita uma pequena avaliação qualitativa e quantitativa acerca do que se era expressado, bem como seu valor para a pesquisa que estava sendo feita. Essa avaliação garantiu a existência de uma base de conhecimento sólida para a execução do projeto. Dessa forma, podemos concluir que mesmo que seja um processo custoso, o método aplicado é robusto e gera um resultado que pode ir além das expectativas vistas em seu planejamento, o que por si só pode ser visto como um indicativo de uma garantia de sucesso em algum aspecto do trabalho.

4 Análise e discussão dos resultados

4.1 O modernismo sob o ponto de vista da programação

Ao observarmos o percurso histórico dos métodos de otimização, podemos ir tão atrás no tempo quanto a matemática é usada, mas para que possamos considerar passagens de eras é necessário buscar acontecimentos com uma certa relevância, e que tais acontecimentos estejam relacionados a contextos de relevância na época.

Um grande acontecimento que podemos tomar como algo que marque a passagem de uma era para outra é o desenvolvimento do cálculo por Newton (sabe-se que existem outros, mas Newton é o mais creditado). Ao observarmos o que seria um processo de otimização matemática prévio ao cálculo, vemos que parece ser algo de certa forma não tão eficiente ou prático. Mesmo considerando um problema simples, tomando que sua interpretação por meio de gráficos bastaria para uma resolução do problema, não seria algo tão trivial como percebemos hoje, uma vez que até a própria estruturação de problemas gráficos era algo recente no momento de Newton, já que a estruturação de coordenadas do plano acabava de ser feita por Descartes.

Logo após o desenvolvimento do cálculo, toda uma nova forma de se considerar problemas que já existiam ou ainda derivar novos problemas foi conhecida. Nomes como Bernoulli, Euler, Cauchy (2.3) Weierstrass (1) comprovam a rápida aceitação das ideias de cálculo e análise matemática, como também formam bases sólidas para o que temos sobre otimização atualmente. Métodos já tinham sido propostos, bem como algumas de suas propriedades, como o próprio método de Newton (em colaboração com Raphson), ou ainda a primeira aparição do método do gradiente descendente, por Cauchy.

Desde essa época, o estudo da otimização foi tomando ramos e se especializando cada vez mais. As especializações normalmente surgem de acordo com modelos de problemas reais que deseja-se otimizar, por isso hoje somos capazes de encontrar toda uma vasta bibliografia a respeito de otimização de problemas lineares, não lineares, considerando variáveis inteiras, variáveis reais, ou ambas.

Ao considerarmos os interesses de cada época em aprimorar o conhecimento matemático, podemos perceber acontecimentos que estão relacionados a esses interesses. Ao tomarmos como um ponto de partida o movimento iluminista no século XVIII, vemos e sabemos a relação com os estudos de Newton e os mesmos de sua época, e também compreendemos que o ideal iluminista contribuiu bastante para a disseminação desses estudos. Um segundo acontecimento

que podemos levar em consideração é a revolução industrial. Conceitos físicos de máquinas pensadas durante a revolução industrial são construídos sobre a nova fundação do cálculo, permitindo, dentre outras coisas, melhor eficiência das máquinas. Podemos dizer que um dos pontos altos desse período, mesmo que tardio, é o trabalho de Maxwell sobre a termodinâmica, o qual, de forma geral, é baseado em conceitos de estabilidade de sistemas.

O movimento de industrialização sempre crescente começou a demonstrar seus pontos fracos, principalmente em relação às eficiências das indústrias, ainda no começo do século XX, quando as demandas cresceram, logísticas sobre recursos de produção foram dificultadas, e tensões internacionais estavam se formando. Passada a Primeira Guerra Mundial, diversos países se encontram com recursos escassos e buscando mais atenção no gerenciamento desses recursos. Essa atenção nos leva à dois nomes que desenvolveram, de certa forma, as mesmas ferramentas para as mesmas finalidades, Kantorovich(2.2.1) e Dantzig (2.2.2), ambos buscando otimizar o processo de alocação de pessoas, recursos, ou algo em relação com o potencial de produção. Mesmo havendo um interesse maior em otimização linear nesse período, outras áreas da otimização vinham sendo desenvolvidas também, um exemplo disso são as condições KKT (2.4.1.1).

Agora após a Segunda Guerra Mundial, os computadores digitais começam a ganhar mais espaço. Na década de 50, como uma das primeiras linguagens de programação, temos o Fortran, do inglês Formula Translation, como sendo uma linguagem pensada para os cálculos matemáticos precisos e custosos de se fazer manualmente. Junto a isso, o recém desenvolvimento da otimização linear e as novas bases formadas para outras áreas da otimização expandiu o interesse que se tinha na época.

Quando chegamos na década de 60, encontramos os nomes e acontecimentos já citados na seção 2.3, os quais agora podem ser compreendidos como em uma era moderna. E seguindo daí, não é mais tão claro um acontecimento com influência na otimização para podermos separar uma nova era.

Olhando agora para o lado dos computadores, enquanto a otimização já estava em seu período moderno, os computadores estavam apenas começando, sendo usados mainframes para cálculos já computacionalmente desenvolvidos, o que acabava restringindo o acesso ao estudo nesse meio, mesmo considerando o interesse crescente no assunto. Considerando os avanços rápidos, Fortran nunca deixou de ser usada e atualizada, um exemplo é que ainda hoje diversas bibliotecas matemáticas em fortran são usadas. Avanços, como invenção e difusão de linguagens estruturadas (C e Ada) contribuíram na difusão e acessibilidade dos estudos sobre otimização, tanto que o DONLP2 (2.4.2.2) que foi escrito em C, e Ada é uma linguagem usada até hoje como linguagem para sistemas críticos, como sistemas operacionais da aviação.

Agora considerando como um todo a análise histórica apresentada no capítulo 2, o SCP se encaixa em um contexto com um ideal semelhante ao renascentismo. É dito na apresentação do SCP que um de seus objetivos é recuperar o interesse no uso de técnicas de resolução de subproblemas lineares que haviam sido deixados de lado em favor dos métodos quadráticos.

Considerando os resultados obtidos pelos autores do SCP, foi mostrado por eles que houve uma mudança de foco, sendo voltado para outras técnicas de otimização, isso sem antes esgotar as possibilidades que os problemas lineares tinham a oferecer. Podemos ver isso ao considerar que as técnicas empregadas no SCP já existiam, ou pelo menos tinham suas bases firmes, no momento que os métodos quadráticos ganharam sua fama. Dito isso, podemos colocar o SCP como sendo um método pós-moderno/renascentista, também considerando a década de sua concepção, de 2000 à 2010.

4.2 Introdução ao SCP

O *Sequential Cutting Plane* (SCP) (STILL; WESTERLUND, 2010) pode ser visto como uma demonstração da capacidade das técnicas de otimização por problemas lineares que, como já dito, haviam sido deixadas de lado. O método apresentado por Claus Still e Tapio Westerlund em 2010 pode ser visto como a conclusão do estudo deles sobre abordagens de subproblemas lineares em problemas não lineares. Um estudo de pelo menos 5 anos, com o primeiro artigo sendo uma apresentação do método em um estágio inicial aplicado a problemas de otimização não linear inteira mista (MINLP) (STILL; WESTERLUND, 2005).

O método foi primariamente aplicado em problemas MINLP como parte de um algoritmo, demonstrando melhor performance em problemas MINLP convexos. Outra possibilidade de aplicação é o uso para resolução de problemas não lineares de grande escala, de grande dimensionalidade, visto a problemática enfrentada por métodos quadráticos para tais problemas.

4.2.1 Aplicação SCP em MINLP

Uma das aplicações do SCP foi como parte de um otimizador de outra classe de problemas, na otimização não linear inteira mista (MINLP). Existem problemas que buscam o ótimo usando variáveis tanto discretas como contínuas, com relações entre elas que são não lineares e que também contém funções não convexas. Tudo isso dificulta muito a computação da solução. Otimizadores capazes de resolverem esses tipos de problemas são ditos como os mais flexíveis, já que seu escopo é tão abrangente. Isso o torna o centro das atenções por pesquisadores de todas as áreas.

Problemas MINLP, na sua forma mais abstrata se dão da seguinte forma:

$$\min_x f(x) \quad (127a)$$

$$\text{s.a } x \in \mathcal{F} \subseteq \mathbb{R}^n \quad (127b)$$

onde o conjunto $\mathcal{F}$ pode ser tanto não linear quanto discreto, e $f : \mathbb{R}^n \mapsto \mathbb{R}$. As diferentes escolhas de f e $\mathcal{F}$ levam as diferentes subclasses de MINLP, das quais temos algumas denominações mais comuns, como:

- Otimização Convexa, onde $\mathcal{F}$ é convexo, e o ótimo é inteiro;

- Otimização Disjuntiva, onde algumas variáveis são contínuas e outras são booleanas;
- Otimização Não Linear, onde restrições não lineares constroem $\mathcal{F}$ e/ou f é não linear. Normalmente usados métodos *Sequential Quadratic Programming* (SQP);
- Otimização por meio de grafos de expressões;
- Otimização por meio de subespaços convexos e/ou linearizações de $\mathcal{F}$.

Podemos encontrar que as formas mais gerais de MINLP são incomputáveis, visto que pode não existir informações ou ferramentas para serem usadas na resolução do problema. Para problemas MINLP convexos, e com restrições de desigualdades, a resolução de subproblemas lineares oferecem limites inferiores para o ótimo, o que para o SCP, saber desses limites ajuda em questões de eficiência. Segundo os autores do SCP, ao ser usado para problemas MINLP, foram encontradas soluções melhores em apenas um minuto do que usando outros métodos que encontraram boas soluções em 12 horas. Essa implementação foi de uma versão prévia do SCP, apresentada em 2006 (STILL; WESTERLUND, 2005), mas também com um código de difícil acessível.

A resolução de problemas MINLP convexos é importante no contexto global de problemas MINLP, já que a maioria dos problemas são resolvidos por uma sequência de problemas que foram convertidos em problemas convexos. Experiências numéricas mostram que o SCP pode ser aplicado para problemas MINLP não convexos como solucionador de subproblemas. A convergência ao ponto estacionário, quando aplicado a estes tipos de problemas, aparenta ser rápida o suficiente mesmo que o estado do algoritmo esteja longe do ponto estacionário.

4.3 Análise Qualitativa do algoritmo - SCP

O SCP resolve problemas no seguinte formato:

$$\min_x f(x) \quad (128a)$$

$$\text{s.a } g_j(x) \leq 0, \quad j = 1, \dots, m_i \quad (128b)$$

$$h_r(x) = 0, \quad r = 1, \dots, m_e \quad (128c)$$

Onde todas as funções devem ser continuamente diferenciáveis em $\mathbb{R}^n$ assim como devem ser funções $\mathbb{R}^n \mapsto \mathbb{R}$. Um avanço em relação a forma anterior do método é que nenhuma função é assumida ser convexa, o que se deve ao refinamento das ferramentas utilizadas, mais especificamente, a não necessidade de nenhuma propriedade de funções convexas para o funcionamento do método. A única coisa assumida é que alguma das funções $g_j(x)$ é linear. Com isso podemos perceber que o SCP é capaz de trabalhar com uma grande quantidade de problemas, visto que essa forma de problema pode ser modelada de aplicações em diversas áreas.

Outra coisa que é assumida, é que as qualificações de restrições de Mangasarian-Fromovitz estendidas (*Extended Mangasarian-Fromovitz Constraint Qualification*, EMFCQ) (PILLO, 1994) são válidas para qualquer $x \in \mathbb{R}^n$, como visto na seção 2.4.1.2.

Estas qualificações garantem que, para qualquer ponto não viável, exista uma direção de descida em direção a uma região viável. Existe também uma relação entre as EMFCQ e funções de penalidade exatas.

4.3.1 O Algoritmo

O algoritmo é iterativo, assim como no clássico método de Newton se é gerada uma sequência de pontos convergentes para um ponto ótimo. Cada uma de suas iterações consiste em um conjunto de subiterações de resolução de problemas lineares. A cada problema linear resolvido, é obtida uma direção de descida, e com essa direção, é feita uma busca em linha, buscando um tamanho ótimo para o passo. Após essas iterações de problemas lineares, é esperado que um novo ponto capaz de reduzir suficientemente uma função de mérito, o que garante a convergência. Uma sumarização do método pode ser visto no Algoritmo 1.

Algoritmo 1: SCP - Sequential Cutting Plane

```

1  $k \leftarrow 1$ ;
2  $H_k \leftarrow I$ ;
3  $x_k \leftarrow$  ponto inicial;
4 while  $k \leq$  iterações máximas do
5    $(d_k, x_{k+1}, H_{k+1}, \lambda_{k+1}, \mu_{k+1}, est) \leftarrow SPL(x_k, H_k)$  // Problemas lineares
6   if  $est = true$  then
7     break ; // Para se foi encontrado um ponto KKT estacionário
8   end
9    $x_{k+1} \leftarrow VFM(d_k, x_k, \lambda_k, \mu_k, x_{k+1}, \lambda_{k+1}, \mu_{k+1})$  ; // Verificação de
    descida
10   $ARC(x_k, x_{k+1})$ ; // Atualização das regiões de confiança
11  if  $KKTE(x_{k+1}, \lambda_{k+1}, \mu_{k+1}) = true$  then
12    break ; // Para se foi encontrado um ponto KKT estacionário
13  end
14   $k \leftarrow k + 1$ 
15 end

```

A função SPL , executada na linha 5, é responsável por resolver a sequência de problemas lineares buscando direções ótimas de descida enquanto já desloca o ponto atual. A função VFM , chamada na linha 9, é responsável por fazer a verificação da qualidade do novo ponto encontrado por SPL , se o ponto satisfizer as condições impostas, ele é mantido, caso contrário é buscado um que satisfaça por outros meios. A função ARC , na linha 10, verifica o tamanho da região de confiança para as soluções d encontradas em SPL . Por fim, $KKTE$, linha 11, verifica se o ponto é um KKT estacionário do problema. Não é necessário apresentar a implementação da função $KKTE$, visto sua simplicidade, uma vez que é constituída a partir de verificações feitas por cálculos simples. Quando o algoritmo para, o ponto mais recente

adicionado à sequência é a solução, x_k se foi atingido o número máximo de iterações, ou x_{k+1} em uma parada por ponto estacionário. O número máximo de interações verificados na linha 4, a priori, deve ser escolhido empiricamente pelas conclusões tiradas a partir do funcionamento do método. Uma outra possibilidade é usar um mecanismo de limitação dinâmica, com base em análises simples do problema observado.

4.3.1.1 Subiterações lineares

Cada subiteração linear é a resolução de um problema linear fixado no ponto atual x_i , i sendo a iteração linear atual. O problema linear que será resolvido é dado por:

$$\min_x \quad \nabla f(x_i)^T d + C \left(\sum_{j=1}^{m_i} t_j^g + \sum_{r=1}^{m_e} t_r^{h^+} + \sum_{r=1}^{m_e} t_r^{h^-} \right) \quad (129a)$$

$$\text{s.a} \quad g_j(x_i) + \nabla g_j(x_i)^T d \leq t_j^g, \quad j = 1, \dots, m_i \quad (129b)$$

$$h_r(x_i) + \nabla h_r(x_i)^T d = t_r^{h^+} - t_r^{h^-}, \quad r = 1, \dots, m_e \quad (129c)$$

$$(d_r)^T (H_i d) = 0, \quad r = 1, \dots, i-1, i > 1 \quad (129d)$$

$$d^L \leq d \leq d^U \quad (129e)$$

$$0 \leq t_j^g \leq \max(g_j(x_i), 0), \quad j = 1, \dots, m_i \quad (129f)$$

$$0 \leq t_r^{h^+} \leq |h_r(x_i)|, \quad r = 1, \dots, m_e \quad (129g)$$

$$0 \leq t_r^{h^-} \leq |h_r(x_i)|, \quad r = 1, \dots, m_e \quad (129h)$$

com a solução sendo d, t^g, t^{h^+}, t^{h^-} .

As restrições (129b) e (129c) são versões lineares das restrições originais, onde a busca por um d_i que resolva o problema garante que na próxima iteração, o ponto já não viole restrições. As variáveis t^g, t^{h^+}, t^{h^-} , são relaxamentos das restrições, o que facilita encontrar um d_i dentro de (129e).

A restrição (129d) faz com que a direção de descida d_i seja H-conjugada (49) à todas as outras direções encontradas anteriormente por outras subiterações lineares desta interação não linear. H_i é a matriz hessiana da função Lagrangiana (30) do problema original, também avaliada no ponto x_i .

Não é necessária mais que uma iteração de resolução de problema linear para que o algoritmo venha a convergir, mas ainda assim pode ser feito para melhorar a velocidade de convergência. Como é necessário que a cada iteração linear a solução d_i seja conjugada à todos os outros d s de iterações lineares anteriores, o algoritmo busca uma direção de descida que "aponte" mais certamente para o ótimo. Por esse motivo esse método lembra o gradiente conjugado, quando essa restrição é considerada.

A restrição (129e) nada mais são que um paralelepípedo que limita o espaço de busca de uma direção descida. Onde $d^L < \vec{0}$ e $d^U > \vec{0}$.

As linearizações vistas em (129b) e (129c) e o requerimento de ortogonalidade entre as direções de descida encontradas em respeito à Lagrangiana da função em (129e) são

(hiper)planos que cortam o espaço.

As restrições (129b) são linearizações que geram uma região no espaço que aproxima a região viável das restrições $g_j(x)$, $j = 1, \dots, m_i$. Da mesma forma (129c) são (hiper)planos que aproximam também a região limitada de $h_r(x)$, $r = 1, \dots, m_e$.

Por fim as restrições (129f), (129g) e (129h) controlam o quanto a função objetivo (129a) é relaxada em relação às restrições que não estão sendo satisfeitas. Quando todas as restrições são satisfeitas em x_i , vemos que t^g , t^{h^+} e t^{h^-} são todos nulos.

Para a implementação da função SPL , como apresentado no Algoritmo 2, precisamos inicialmente de uma forma de gerar o problema linear (129), que uma vez gerado, na linha 6, deve ser resolvido (linha 7). O próximo passo a ser tomado é resolver o problema dual de (129), e feito isso a solução do problema dual pode ser usada como estimativa dos multiplicadores de Lagrange, o que é feito na linha 8. Após isso, já temos a possibilidade de verificar se o ponto atual é um ponto KKT estacionário (visto a necessidade dos multiplicadores de Lagrange para isso). Caso seja KKT estacionário já podemos retornar as informações necessárias, como é feito na linha 10, do contrário, é feita uma busca em linha buscando um tamanho para o passo na direção encontrada na solução do problema linear. Essa busca em linha é feita em uma função Lagrangiana penalizada de acordo com as restrições cumpridas. Uma vez encontrado o tamanho do passo, o ponto da interação atual pode ser atualizado como no método de gradiente descendente. Agora, a depender de como foi escolhido, é computada a matriz Hessiana da função Lagrangiana para a próxima iteração. Por fim, caso algum dos critérios de parada de subiterações lineares forem satisfeitos, retornamos as informações obtidas até agora, caso contrário, repetimos o processo tomando o ponto atualizado como o novo ponto de partida.

Algoritmo 2: SPL - Sequência de Problemas Lineares

```

1 Function SPL( $x_k, H_k$ ):
2    $i \leftarrow 1$ ;
3    $x_i \leftarrow x_k$ ;
4    $H_i \leftarrow H_k$ ;
5   while Condições de parada não satisfeitas do
6      $(A, b, c) \leftarrow \text{gerar\_problema}(x_i, H_i)$ ;
7      $(d_i, t_i^g, t_i^{h^+}, t_i^{h^-}) \leftarrow \text{primal}(A, b, c)$ ;
8      $(\lambda_i, \mu_i) \leftarrow \text{dual}(A, b, c)$ ;
9     if  $\text{KKT E}(x_i, \lambda_i, \mu_i) = \text{true}$  then
10      | return  $(d_i, x_i, H_i, \lambda_i, \mu_i, \text{true})$ ;
11    end
12     $\alpha_i \leftarrow \text{busca\_em\_linha\_lagrangiana\_penalizada}(x_i, d_i, \lambda_i, \mu_i)$ ;
13     $x_{i+1} \leftarrow x_i + \alpha_i d_i$ ;
14     $H_{i+1} \leftarrow \text{hessiana}(x_i, x_{i+1})$ ;
15     $i \leftarrow i + 1$ ;
16  end
17  return  $(d_i, x_i, H_i, \lambda_i, \mu_i, \text{false})$ ;

```

Para a geração e resolução do problema é necessário compreender o formato dele e ajusta-lo para a melhor forma possível de resolver. Para isso, vamos considerar o seguinte problema de minimização linear e seu dual:

$$\min_x \quad c^T x \quad (130a)$$

$$\text{s.a} \quad Ax \geq b \quad (130b)$$

$$\max_y \quad b^T y \quad (131a)$$

$$\text{s.a} \quad A^T y = c \quad (131b)$$

Para que possamos resolver (129) com facilidade pelos métodos convencionais, temos que transforma-lo na tripla A, b, c dos problemas acima, e com isso, também considerar apenas um único vetor solução x . Tomemos x como sendo $(d, t^g, t^{h^+}, t^{h^-})$, logo, um vetor do $\mathbb{R}^{n+m_i+m_e+m_e}$. Podemos perceber rapidamente o que é o vetor c a partir de (129a), que as primeiras entradas são $\nabla f(x_i)$ e todo o resto é o escalar de relaxamento C . Para a matriz A e o vetor b precisamos primeiro reescrever as restrições de forma que fique clara as entradas. Podemos reescrever (129) da seguinte forma:

$$\min_x \quad \nabla f(x_i)^T d + \sum_{j=1}^{m_i} C t_j^g + \sum_{r=1}^{m_e} C t_r^{h^+} + \sum_{r=1}^{m_e} C t_r^{h^-} \quad (132a)$$

$$\text{s.a} \quad -\nabla g_j(x_i)^T d + e_j^T t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq g_j(x_i), \quad j = 1, \dots, m_i \quad (132b)$$

$$-\nabla h_r(x_i)^T d + 0 \cdot t^g + e_r^T t^{h^+} - e_r^T t^{h^-} \geq h_r(x_i), \quad r = 1, \dots, m_e \quad (132c)$$

$$\nabla h_r(x_i)^T d + 0 \cdot t^g - e_r^T t^{h^+} + e_r^T t^{h^-} \geq -h_r(x_i), \quad r = 1, \dots, m_e \quad (132d)$$

$$-(Hd_r)^T d + 0 \cdot t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq 0, \quad r = 1, \dots, i-1 \quad (132e)$$

$$(Hd_r)^T d + 0 \cdot t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq 0, \quad r = 1, \dots, i-1 \quad (132f)$$

$$1 \cdot d + 0 \cdot t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq d^L \quad (132g)$$

$$-1 \cdot d + 0 \cdot t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq -d^U \quad (132h)$$

$$0 \cdot d + e_j^T t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq 0, \quad j = 1, \dots, m_i \quad (132i)$$

$$0 \cdot d - e_j^T t^g + 0 \cdot t^{h^+} + 0 \cdot t^{h^-} \geq -\max(0, g_j(x_i)), \quad j = 1, \dots, m_i \quad (132j)$$

$$0 \cdot d + 0 \cdot t^g + e_r^T t^{h^+} + 0 \cdot t^{h^-} \geq 0, \quad r = 1, \dots, m_e \quad (132k)$$

$$0 \cdot d + 0 \cdot t^g - e_r^T t^{h^+} + 0 \cdot t^{h^-} \geq -|h_r(x_i)|, \quad r = 1, \dots, m_e \quad (132l)$$

$$0 \cdot d + 0 \cdot t^g + 0 \cdot t^{h^+} + e_r^T t^{h^-} \geq 0, \quad r = 1, \dots, m_e \quad (132m)$$

$$0 \cdot d + 0 \cdot t^g + 0 \cdot t^{h^+} - e_r^T t^{h^-} \geq -|h_r(x_i)|, \quad r = 1, \dots, m_e \quad (132n)$$

em que e_k é a k -ésima base canônica e $(\cdot)$ é o produto de um vetor por um escalar.

Posto assim, extrair A e b não é tão difícil. Basta considerar funções para cada família de restrições retornando blocos de A e entradas de b . Durante esse processo é necessário tomar cuidado com o custo tanto em memória consumida quanto em processamento. Todas as funções e restrições são avaliadas em x_i , então é correto que se faça essa avaliação antes de começar a gerar a matriz e os vetores. A respeito da memória, não há muito o que se possa fazer, a não ser considerar variáveis temporárias ou algo semelhante.

Depois de gerar a matriz e os vetores, chega a vez de resolver o problema. A forma como o problema é resolvido não importa, desde que se obtenha o resultado correto de forma eficiente. Uma vez que a linguagem escolhida para a implementação foi Rust, foi usada uma biblioteca própria para problemas lineares, chamada de minilp, mantida por Alexey Zatelepin ([ZATELEPIN, 2020](#)), que é capaz de resolver os problemas rapidamente.

Resolvido ambos os problemas, primal e dual, precisamos agora extrair as estimativas dos multiplicadores de Lagrange. Para isso foi considerada a prova do Lema 3 em ([STILL; WESTERLUND, 2010](#)) onde essa versão do SCP foi proposta. Pela prova podemos perceber que as estimativas dos multiplicadores são as $m_i + m_e + m_e$ entradas da solução do problema dual, referentes às restrições (132b), (132c) e (132d). Podemos extrair λ das m_i primeiras entradas da solução dual. Já para μ , precisamos ter μ^+ como as m_e entradas logo após m_i entradas, e μ^- como as m_e entradas logo após $m_i + m_e$ entradas, e por fim, $\mu = \mu^+ - \mu^-$.

Agora com as estimativas dos multiplicadores podemos verificar se o ponto atual é um KKT estacionário. Caso seja, já podemos retornar as informações pertinentes, $d_i, x_i, H_i, \lambda_i, \mu_i$

e um sinal informando que as iterações podem parar. Esse sinal é mais necessário por questões de eficiência, se não fosse enviado um sinal o algoritmo pararia na próxima verificação de KKT, no entanto, seriam feitas uma imensa quantidade de operações à toa.

O próximo passo é encontrar um tamanho de passo na direção d_i que minimize uma função Lagrangiana penalizada, dada por:

$$\tilde{L}(x, \lambda, \mu) = f(x) + \sum_{j=1}^{m_i} \lambda_j g_j(x)^+ + \sum_{r=1}^{m_e} |\mu_r h_r(x)| + \rho \sum_{j=1}^{m_i} \lambda_j (g_j(x)^+)^2 + \rho \sum_{r=1}^{m_e} |\mu_r| (h_r(x))^2 \quad (133)$$

Podemos perceber essa função como a Lagrangiana penalizada já que ela considera apenas as restrições que não foram cumpridas, uma vez que $g_j(x)^+ = \max(0, g_j(x))$. Como se deseja minimizar essa função na direção d_i , qualquer restrição que venha a ser descumprida irá fazer a função crescer. O objetivo da minimização é encontrar um único valor real α_i entre 0 e 1.

$$\min_{0 \leq \alpha_i \leq 1} \tilde{L}(x_i + \alpha_i d_i, \lambda_i, \mu_i) \quad (134)$$

A forma como é feita essa busca em linha deve ser escolhida com cuidado. Se for possível e de interesse, uma busca em linha exata deve ser feita, se não pelo menos deve-se considerar uma boa aproximação. Se o tamanho do passo for impreciso demais, significando que é fácil encontrar um ponto ótimo melhor que o que viria a ser o próximo ponto quando considerada esse tamanho de passo, podemos ter problemas quando o algoritmo estiver próximo de um ótimo, podendo, por exemplo, nunca se aproximar o suficiente para que seja considerado solução.

Um próximo passo, se foi escolhido fazer dessa forma, é a computação da matriz Hessiana da função (30) Lagrangiana. Os autores do SCP propuseram o uso do BFGS (2.7.2) por ser um método iterativo, e por isso, poder ser evitado cálculos desnecessários. A outra opção considerada para a implementação foi calcular a Hessiana de forma exata, usando o método já apresentado em (103).

Agora, no final da iteração, é incrementado o contador de iterações e são verificadas as condições de parada. Temos ao todo cinco condições a serem verificadas, bastando que alguma delas seja descumprida para parar as iterações. A primeira verifica se $i > n$, aplicável apenas quando a restrição (129d) é considerada, já que sendo um método conjugado, não se tem mais como haver outras direções de descida para a busca da solução. A segunda é quando a norma de d_i é próxima de 0, indicando nenhum movimento. A terceira condição é quando alguma variável de relaxamento não é reduzida, ou seja, não existe nenhum j de forma que $t_j^g \leq g_j(x)$ e também não existe nenhum r de forma que $t_r^{h^+} \leq h_r(x)$ e $t_r^{h^-} \leq h_r(x)$. A quarta condição é se já estamos pelo menos na segunda iteração e qualquer variável de relaxamento é positiva. A quinta é se α_i é próximo de 0, pelos mesmos motivos que a segunda condição. Uma vez parada as iterações de resoluções de problemas lineares, deve ser retornada as informações necessárias, $(d_i, x_i, H_i, \lambda_i, \mu_i)$.

4.3.1.2 Função Mérito

É usada uma função de mérito para verificações da velocidade de descida. O mérito é dado pela minimização da função objetivo (128a) e a anulação de um termo de penalização:

$$M(x) = f(x) + p(x) \quad (135)$$

O termo de penalização é semelhante aos dois primeiros termos que penalizam a função objetivo na Lagrangiana penalizada (133). No entanto, ao invés de ser usado as estimativas dos multiplicadores de Lagrange, são usados números maiores que elas, garantindo que a penalização seja ditada apenas por uma proporção dos valores das restrições não cumpridas, o que vem a penalizar de acordo com o quanto o problema está distante de ser resolvido.

$$p(x) = \sum_{j=1}^{m_i} \bar{\lambda}_j g_j(x)^+ + \sum_{r=1}^{m_r} \bar{\mu}_r |h_r(x)| \quad (136)$$

Não é dito o quanto os valores para $\bar{\lambda}$ e $\bar{\mu}$ devem ser maiores que as estimativas, então devemos contar com a experiência trazida pela aplicação do método. Já que $\bar{\lambda} > \lambda$ e $\bar{\mu} > \mu$ basta considerar um incremento para os valores, $\bar{\lambda} = \lambda + \lambda_{inc}$ e $\bar{\mu} = \mu + \mu_{inc}$.

A verificação da qualidade da descida é dada por três condições, o decréscimo da função de mérito, o decréscimo suficiente da função de mérito, e se o passo tomado é suficientemente grande. A primeira condição é apenas verificar se $M(x_{k+1}) \leq M(x_k)$. A segunda condição é feita por querer descobrir se o passo tomado desacelera a descida do algoritmo, verificando se a função está sendo reduzida suficientemente. Essa verificação é feita pela análise da diferença das avaliações da função de mérito em x_k e x_{k+1} em relação com a derivada direcional da função.

$$M(x_{k+1}) - M(x_k) \leq \sigma \alpha_k D_{d_k} M(x_k), \quad \sigma \in (0, 1) \quad (137)$$

Uma interpretação rápida para a derivada direcional é medir o quanto a descida pelo gradiente vai de encontro com a direção esperada. Nesse caso, a derivada mede a velocidade de descida da função de mérito na direção d . A derivada direcional de $M(x)$ é dada por:

$$D_d M(x) = \nabla f(x)^T d + \sum_{j=1}^{m_i} \bar{\lambda}_j D_d g_j(x)^+ + \sum_{r=1}^{m_e} \bar{\mu}_r D_d |h_r(x)| \quad (138)$$

E as derivadas direcionais das restrições consideram apenas quando as restrições não estão sendo cumpridas:

$$D_d g_j(x)^+ = \begin{cases} \nabla g_j(x)^T d, & \text{se } g_j(x) > 0; \\ \max(\nabla g_j(x)^T d, 0), & \text{se } g_j(x) = 0; \\ 0, & \text{se } g_j(x) < 0. \end{cases} \quad (139)$$

$$D_d|h_r(x)| = \begin{cases} \nabla h_r(x)^T d, & \text{se } h_r(x) > 0; \\ |\nabla h_r(x)^T d|, & \text{se } h_r(x) = 0; \\ -\nabla h_r(x)^T d, & \text{se } h_r(x) = 0. \end{cases} \quad (140)$$

O terceiro teste com a função de mérito é a verificação de que o passo tomado foi grande o suficiente. Essa condição é feita por verificar se a derivada direcional em x_{k+1} é maior que em x_k , mesmo que reduzida:

$$D_{d_k}M(x_{k+1}) \geq \eta D_{d_k}M(x_k), \quad \eta \in (\sigma, 1) \quad (141)$$

Como a derivada direcional é maior quanto mais próximo de d , visto que o movimento descrito por $x_{k+1} - x_k$ caracteriza o denominador da derivada, essa condição ser satisfeita significa que as iterações estão cada vez melhor "apontadas" para o ótimo e fazendo isso com uma velocidade significativa.

A derivada direcional da função de mérito, além de considerar a descida da função objetivo, também considera as direções de descida para que as restrições que não estão sendo cumpridas sejam. Então, de forma geral, a derivada direcional da função de mérito quantifica o quão próximo da melhor descida possível o algoritmo está fazendo.

Caso não seja possível cumprir todas as três condições, é necessário buscar um novo ponto que satisfaça. Dentre as várias formas de se fazer isso, a mais simples é buscar o ponto na direção d_k que minimize a função de mérito:

$$\min_{\alpha_n} M(x_k + \alpha_n d_k) \quad (142)$$

Uma vez encontrado α_n , basta salvar o novo ponto em x_{k+1} e retornar:

$$x_{k+1} = x_k + \alpha_n d_k \quad (143)$$

Um pseudo-código para o que foi dito está apresentado no Algoritmo 3.

Algoritmo 3: VFM - Verificação da Função de Mérito

```

1 Function VFM( $d_k, x_k, \lambda_l, \mu_k, x_{k+1}, \lambda_{k+1}, \mu_{k+1}$ ):
2    $\sigma \leftarrow 0.5$ ;
3    $\eta \leftarrow 0.75$ ;
4    $\alpha_k \leftarrow \frac{x_{k+1} - x_k}{d_k}$ ;
5    $c1 \leftarrow (M(x_{k+1}) \leq M(x_k))$ ;
6    $c2 \leftarrow (M(x_{k+1}) - M(x_k) \leq \sigma \alpha_k D_{d_k} M(x_k))$ ;
7    $c3 \leftarrow (D_{d_k} M(x_{k+1}) \geq \eta D_{d_k} M(x_k))$ ;
8   if  $c1 = \text{true}$  e  $c2 = \text{true}$  e  $c3 = \text{true}$  then
9     return  $x_{k+1}$ ;
10  end
11   $\alpha_k \leftarrow \text{busca\_em\_linha\_funcao\_merito}(x_k, d_k, \lambda_k, \mu_k)$ ;
12   $x_{k+1} \leftarrow x_k + \alpha_k d_k$ ;
13  return  $x_{k+1}$ ;

```

4.3.1.3 Regiões de confiança

As regiões de confiança são usadas como delimitações para o espaço das direções de descidas, garantindo que estas sejam de um tamanho adequado, e atualizadas de acordo com os movimentos do ponto em cada iteração, ou seja, se a região é muito grande e o ponto em relação a última iteração não se deslocou tanto, a região pode ser diminuída, e o mesmo para o contrário, a região é aumentada caso a diferença de posições entre iterações seja muito grande. O parâmetro é um número real δ_{max} dado por:

$$\delta_{max} = \max(\delta_l) \quad (144)$$

Onde δ_l é um conjunto de valores reais das maiores diferenças em relação ao tamanho da região de confiança:

$$\delta_l = \max \left(\left| \frac{x_{k+1,l} - x_{k,l}}{d_l^L} \right|, \left| \frac{x_{k+1,l} - x_{k,l}}{d_l^U} \right| \right), \quad l = 1, \dots, n \quad (145)$$

Existe uma tolerância para o tamanho do movimento sem que as regiões sejam atualizadas. Um parâmetro é escolhido como limite antes de crescer a região e outro para diminuir a região. Por experiência numérica dos autores (STILL; WESTERLUND, 2010), foram identificados valores para esses parâmetros, para o crescimento da região $\delta_{inc} = 0.8$ e para a diminuição $\delta_{dec} = 0.2$. Podem ser escolhidos quaisquer valores dentro dos intervalos, que são $0.5 \leq \delta_{inc} \leq 1$ e $0 \leq \delta_{dec} \leq 0.5$. Uma vez considerado esses parâmetros, a atualização da região pode ser feita como exemplificada no Algoritmo 4.

Algoritmo 4: ARC - Atualização da Região de Confiança

```

1 Function ARC( $x_k, x_{k+1}$ ):
2    $\delta_l \leftarrow \{\}$ ;
3    $\delta_{inc} \leftarrow 0.8$ ;
4    $\delta_{dec} \leftarrow 0.2$ ;
5   for  $l \leftarrow 1$  to  $n$  do
6      $\delta \leftarrow x_{k+1,l} - x_{k,l}$ ;
7      $\delta^L \leftarrow \frac{\delta}{d_l^L}$ ;
8      $\delta^U \leftarrow \frac{\delta}{d_l^U}$ ;
9      $\delta_l \leftarrow \delta_l \cup \{\max(\delta^L, \delta^U)\}$ ;
10  end
11   $\delta_{max} = \max(\delta_l)$ ;
12  if  $\delta_{max} < \delta_{dec}$  then
13     $\gamma \leftarrow \frac{\delta_{max}}{\delta_{dec}}$ ;
14    if  $\gamma \neq 0$  then
15       $d^L \leftarrow \gamma d^L$ ;
16       $d^U \leftarrow \gamma d^U$ ;
17    end
18  end
19  if  $\delta_{max} > \delta_{inc}$  then
20     $\gamma \leftarrow 2\delta_{max}$ ;
21    if  $\gamma \neq 0$  then
22       $d^L \leftarrow \gamma d^L$ ;
23       $d^U \leftarrow \gamma d^U$ ;
24    end
25  end

```

Algumas coisas precisam ser consideradas a respeito dessa atualização. Embora seja uma função de atualização simples e não muito custosa, não dar atenção suficiente pode resultar em problemas. A verificação $\gamma \neq 0$ é necessária uma vez que se a atualização por algum motivo não parou anteriormente, a distância entre os pontos de cada iteração pode ser praticamente nula, o que leva para o anulamento da região de confiança para d , forçando que $d = \vec{0}$, o que de novo leva para uma nova iteração sem movimento e iterando no mesmo ponto até o limite de iterações.

4.4 Análise Quantitativa do algoritmo - SCP

Para que pudesse ser feita uma análise quantitativa completa, seria necessário ter uma proposta de código finalizada, o que não é o caso. Para que possa se ter essa proposta, é preciso fazer

uma avaliação fina tanto do funcionamento do método como um todo, quanto com cada parte isolada, com um foco especial na busca da direção de descida e no ajuste das constantes de penalidade presentes em todo o processo. No entanto, será apresentado resultados que consideramos parciais acerca do custo de processamento. O código de nossa autoria que foi construído como uma proposta para o SCP é o que usamos nos testes, e este está disponível publicamente em um repositório no Github ¹.

Para os testes, foram usados 10 problemas de 3 bases de dados diferentes. Duas dessas bases de dados foram as mesmas usadas pelos desenvolvedores do SCP, uma compilada por Hock e Schittkowski (HOCK; SCHITTKOWSKI, 1981) e Schittkowski (SCHITTKOWSKI, 1987). Enquanto a elaboração do teste, apenas são considerados problemas cujo domínio é o $\mathbb{R}^2$, e problemas com ambas as restrições, tanto de igualdade quanto de desigualdade, por simplicidade e verificação da consistência entre problemas. Dado o estado atual da implementação do método, não foi vista a necessidade de apresentação do ponto supostamente ótimo encontrado pelo algoritmo para cada um dos problemas apresentados. No entanto, será falado a respeito das implicações acerca do que foi encontrado.

Todos os testes foram feitos em uma CPU AMD FX-6300, no sistema operacional ArchLinux-2021.06.01. O compilador usado foi o rustc 1.51.0-nightly, edição 2018, compilação não otimizada e informações de debug. A não otimização e a preservação das informações de debug do programa são feitas para que se possa usar um debugger mais facilmente e evitar o rearranjo de código por parte do compilador, uma vez que durante a compilação as otimizações feitas podem alterar drasticamente a ordem das operações executadas a fim de garantir a maior eficiência possível para as operações de baixo nível.

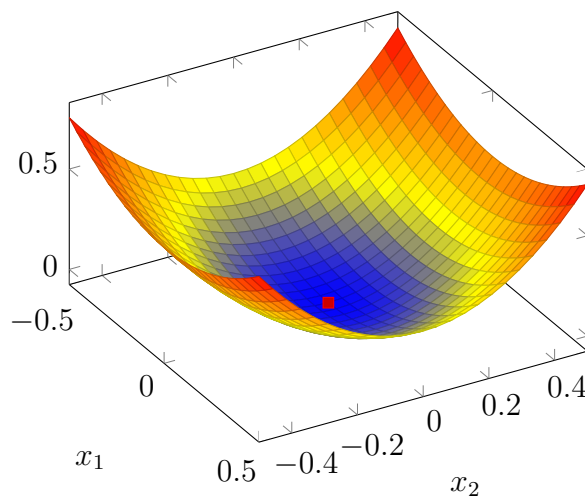
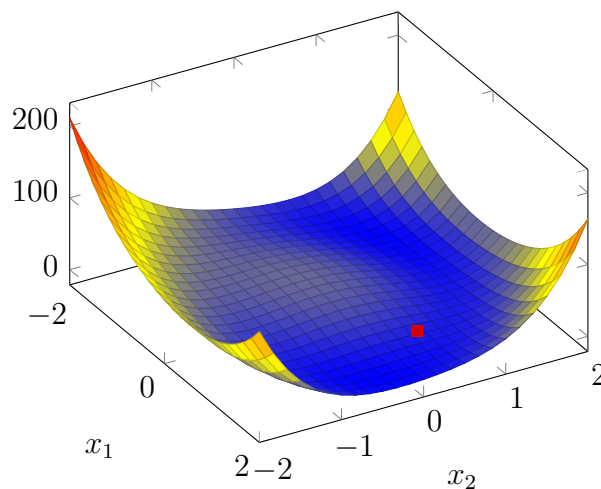
4.4.1 Problemas de Surjanovic e Bingham

A primeira base de problemas considerada (SURJANOVIC; BINGHAM, 2013) é parte de um projeto colaborativo para se ter uma biblioteca de funções e conjuntos de dados para testes em diversas áreas de otimização, simulação e modelagens em geral. Foram escolhidos quatro problemas da base por serem problemas simples, os quais estão listados na Tabela 1. No entanto, são todos irrestritos, isto é, consistem apenas em minimizar a função sem nenhuma restrição.

¹Disponível em: <https://github.com/Sekva/tcc_prog/tree/56fa674>

Tabela 1 – Funções de teste irrestritas.

Nome	Expressão	Ótimo (x^*)
Bohachevsky	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$	$(0, 0)$
Perm	$f(x) = \sum_{i=1}^2 \left(\sum_{j=1}^2 (j + \beta) x_j^i - \frac{1}{j^i} \right)^2$	$(1, \frac{1}{2})$
Trid	$f(x) = \sum_{i=1}^2 (x_i - 1)^2 - \sum_{i=2}^2 x_i x_{i-1}$	$(2, 2)$
Sum Squares	$f(x) = \sum_{i=1}^2 i x_i^2$	$(0, 0)$

Figura 1 – Função de Bohachevsky com ótimo em $(0, 0)$.Figura 2 – Função Perm com ótimo em $(1, 0.5)$.Figura 3 – Função Trid com ótimo em $(2, 2)$.

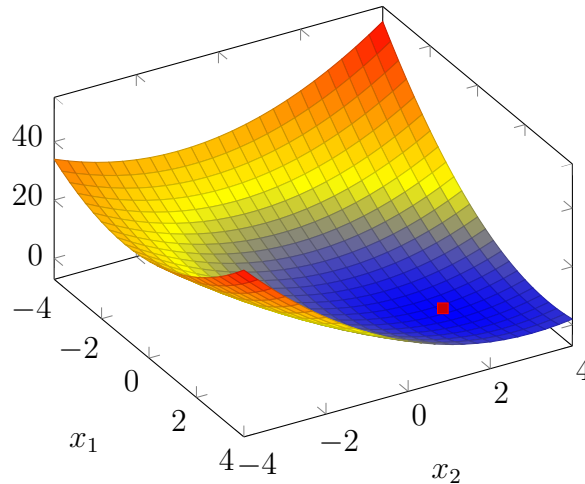
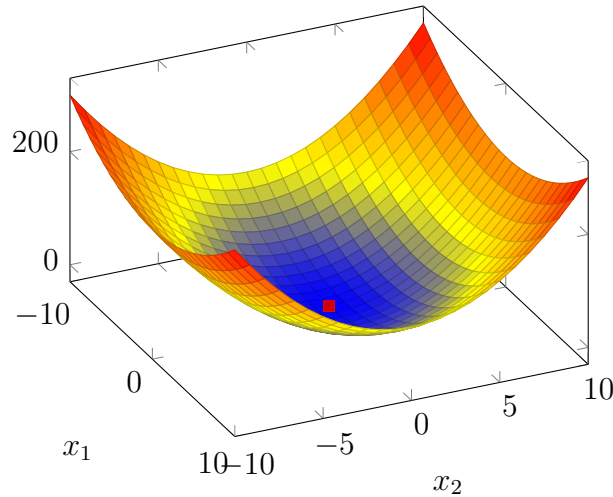


Figura 4 – Função Sum Squares com ótimo em (0, 0).



Como todos os problemas são irrestritos, foram consideradas restrições simples. Para as restrições de desigualdade, foram consideradas quatro funções limitando o ponto a um quadrado em uma distância razoável do ótimo. Para as restrições de igualdade, basta considerar uma função que passe no ótimo, a mais simples é apenas uma reta. Devemos lembrar que estamos trabalhando apenas no $\mathbb{R}^2$.

Cada uma das funções buscam testar o funcionamento da descida do método em superfícies com características diferentes. A função de Bohachevsky (Figura 1) tem uma superfície com lombadas, causadas pela função trigonométrica em sua definição, o que faz verificar se o processo não fica preso em um ótimo local. A função de Perm (Figura 2) tem uma espécie de anel em que o mínimo está localizando, verificando se o método é capaz de contornar ou atravessar esse anel. A função Trid (Figura 3) tem uma inclinação diferente em cada direção, o que serve de apoio para a análise das direções de descidas encontradas durante os testes. Por fim, a função Sum Squares (Figura 4) é apenas um parabolóide elíptico de rápida descida.

4.4.2 Problemas de Hock e Schittkowski

As outras duas bases de testes são compilações de funções usadas ao longo de anos de estudos sobre métodos de resolução de problemas não lineares dos autores. Segundo eles, o livro onde é apresentada a base de dados deve ser considerado uma assistência para o design de testes, uma vez que a performance só pode ser medida de fato através de experiência numérica, a qual se faz necessário ter uma coleção de problemas para que exista uma experiência.

Da primeira base, foram escolhidos os seguintes problemas:

$$\min_x f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (146a)$$

$$\text{s.a } 1.5 \leq x_2 \quad (146b)$$

$$x^* = (1, 1) \quad (146c)$$

$$\min_x f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (147a)$$

$$\text{s.a } -0.25x_1^2 - x_2^2 + 1 \geq 0 \quad (147b)$$

$$x_1 - 2x_2 + 1 = 0 \quad (147c)$$

$$x^* = (0.5(\sqrt{7} - 1), 0.25(\sqrt{7} + 1)) \quad (147d)$$

O problema (146) é o Problema N°1 e (147) é o Problema N°14. Já da segunda base:

$$\min_x f(x) = -x_2 \quad (148a)$$

$$\text{s.a } 1 + x_1 - 2x_2 \geq 0 \quad (148b)$$

$$x_1^2 + x_2^2 - 1 = 0 \quad (148c)$$

$$x^* = (0.6, 0.8) \quad (148d)$$

$$\min_x f(x) = -x_1 \quad (149a)$$

$$\text{s.a } (1 - x_1)^3 - x_2 \geq 0 \quad (149b)$$

$$x^* = (0.25, 0.25) \quad (149c)$$

$$\min_x f(x) = (x_1 - 20)^2 + (x_2 + 20)^2 \quad (150a)$$

$$\text{s.a } \frac{x_1^2}{100} + \frac{x_2^2}{36} = 0 \quad (150b)$$

$$x^* = (7.809, -3.748) \quad (150c)$$

$$\min_x f(x) = x_1^2 + x_2 \quad (151a)$$

$$\text{s.a } -(x_1 + x_2) + 1 \geq 0 \quad (151b)$$

$$-(x_1 + x_2^2) + 1 \geq 0 \quad (151c)$$

$$x_1^2 + x_2^2 - 9 = 0 \quad (151d)$$

$$x^* = (-2.372, -1.336) \quad (151e)$$

Temos que (148) é o Problema N°217, (149) é o Problema N°221, (150) é o Problema N°313 e (151) é o Problema N°325.

4.4.3 Avaliação numérica do SCP

Para avaliar o custo do SCP no estado atual temos que verificar um custo geral e também custos em partes específicas. Para isso observaremos o tempo de processamento médio de cada iteração, o tempo total das subiterações lineares, o tempo total entre as subiterações lineares e o fim da iteração, e os tempos de resolução do problema primal e dual, bem como a geração das informações para tal. Todos os tempos são em segundos.

Deve ser lembrado que não existe um código acessível para o SCP, uma vez que os autores não disponibilizaram o que foi usado para os resultados numéricos. Portanto não existe um código de referência para a nova construção. Comparativos entre os testes numéricos da implementação aqui proposta e a implementação feita pelos autores também é algo dificultado pela falta de acessibilidade do código. A escrita do método, considerando todas as suas vantagens, pode ser considerado algo inovador, já que existe a contribuição de ser exposto um algoritmo funcional e público para um método que é apresentado para a comunidade como sendo de difícil acesso.

Tabela 2 – Tempo de execução de diversas etapas.

Problema	Iteração	SLP	Pós SLP	Primal	Dual	Geração
Bohachevsky	0.013982	0.012956	0.000069	0.007525	0.002942	0.000273
Perm	0.036630	0.034369	0.002100	0.007365	0.006032	0.000263
Trid	0.013688	0.012513	0.001262	0.004786	0.004890	0.000228
Sum Squares	0.038192	0.035703	0.002019	0.007337	0.006878	0.000284
Prob. n°1	0.012007	0.010918	0.000044	0.002089	0.001545	0.000134
Prob. n°14	0.006658	0.006107	0.000526	0.001957	0.001557	0.000125
Prob. n°217	0.006057	0.005336	0.000698	0.001936	0.001501	0.000127
Prob. n°221	0.005204	0.004598	0.000568	0.000884	0.000519	0.000078
Prob. n°313	0.014293	0.013699	0.000459	0.004419	0.004174	0.000192
Prob. n°325	0.013886	0.013006	0.000989	0.002889	0.001671	0.000144

Como o código ainda não está finalizado, não foram apresentadas as soluções. No entanto, dos 10 problemas, 2 são resolvidos e os demais levantam questões a respeito da implementação feita. Dois ótimos encontrados pelo código (no estado atual) são os ótimos reais, a solução do problema de Bohachevsky e o Problema N°217. Dentre as questões que podem ser percebidas das soluções dos outros problemas temos, por exemplo, o fato de ter encontrado a solução apenas em uma coordenada e na outra ter encontrado o simétrico do que deveria. Ou ainda, encontrar soluções não tão distantes da solução real, mas com uma proximidade não suficiente para que sejam consideradas um erro de precisão. Desses detalhes podemos concluir que o código no estado atual está bem próximo de ser terminado. Dentre os problemas que podem existir, podemos pensar sobre a geração da matriz A (131b) dos problemas lineares, a qual passa por diversas transformações para ser construída a partir da definição das restrições do problema (129) e chegar em sua forma matricial.

4.5 As vantagens e desvantagens computacionais do SCP

Agora pensando exclusivamente na implementação do método, primeiro foi necessária a construção de uma estrutura para o armazenamento do problema. Essa estrutura guarda a função objetivo do problema, uma lista de funções das restrições de desigualdades e outra lista de funções para as restrições de igualdades. Para o armazenamento das funções basta considerar ponteiros para elas. Outras informações que podem ser armazenadas nessa estrutura do problema são as regiões de confiança, como os pontos que de fato são. Para essa estrutura também foi criada uma função para avaliar todos os aspectos básicos do problema em um certo ponto. Nessas funções são avaliada a função objetivo e seu gradiente e todas as restrições e seus gradientes. Essa função é usada em quase todas as vezes que se é requerida uma avaliação do problema, não sendo usada em casos onde se usam certas funções isoladamente. A possibilidade de se ter essa estrutura onde todas as funções podem ser avaliadas de uma vez nos dá uma vantagem em questões de performance, uma vez que podemos otimizar o tempo de realizar todas essas avaliações diminuindo consideravelmente o tempo de execução do algoritmo.

Um outro efeito benéfico sobre o método ter esses momentos onde todas as funções são avaliadas é um custo reduzido em problemas com funções muito difíceis. No entanto, ainda é feita uma grande quantidade de avaliações, por isso, mesmo considerando esse efeito benéfico, os autores não recomendam para problemas com funções difíceis.

O uso de subproblemas lineares é o ponto principal do método, e com ele vem todos os efeitos. O problema linear, mesmo que grande, é de rápida computação e, a depender do método de resolução, pode ser resolvido com operações simples da CPU, o que reafirma a velocidade do método. No entanto, complexidade de processo não é a única preocupação que se deve ter com um método capaz de resolver problemas de grande escala. Para que possamos usar métodos já conhecidos para otimização linear, foi necessária a transformação do problema no formato proposto originalmente (129) em um equivalente de formato mais simples (132)

de ser percebido e a partir dele extraído a matriz e os vetores (130), e aí que encontramos o problema.

Mesmo sendo apenas números, o armazenamento da matriz e dos vetores pode vir a se tornar um problema. Primeiro temos que considerar os tamanhos. Para o vetor c , temos que seu número de elementos é $n + m_i + m_e + m_e$, o que de cara não parece ser um problema, principalmente pelo fato de que se foi possível armazenar as funções e restrições na estrutura do problema, deve-se ter capacidade para armazenar as entradas de c . No entanto, isso vale quando consideramos que $c \in \mathbb{R}^n$, mas se por exemplo, quisermos adaptar o método para funcionar com números hiper-duais (2.7.4), que equivalem a quatro números reais, já existe um aumento considerável nos requerimentos de armazenamento.

Sabendo quantos elementos tem o vetor c , sabemos também quantas colunas tem a matriz A , visto que devem ser iguais para que possa ter Ax . O que nos falta agora é saber o número de linhas de A , que também é o número de elemento de b , também para que se possa ter $Ax = b$. Ao observarmos o problema ajustado (132), podemos ver que o número de linhas é de pelo menos $m_i + m_e + m_e + n + n + m_i + m_i + m_e + m_e + m_e + m_e = 2n + 3m_i + 6m_e$. É no mínimo isso pelo fato das restrições (132e) e (132f) adicionarem $2n$ linhas a cada iterações de resolução de problemas lineares, e se chegar a ter n iterações, são $2n^2$ linhas adicionais. Portanto, a matriz A tem no mínimo $(2n + 3m_i + 6m_e) \cdot (n + m_i + 2m_e)$ elementos e no máximo $(2n^2 + 2n + 3m_i + 6m_e) \cdot (n + m_i + 2m_e)$, o que pode facilmente se tornar uma dificuldade de uso em casos onde seria interessante se ter os benefícios, um exemplo seria um dispositivo embarcado que resolva problemas ideais para o SCP, porém tem pouca memória.

Outro ponto que podemos observar é a construção do ferramental para o funcionamento do SCP. Mesmo que isso não seja relacionado ao funcionamento do método propriamente dito, a existência do ferramental por si só já é uma vantagem. Isso talvez seja resultado da posição histórica do método, uma vez que os autores foram capazes de estudar e compreender as ferramentas que utilizaram em um estado muito mais refinado do que se tinha, por exemplo, na década de 70.

Contando só com as ferramentas mais básicas, como as operações entre vetores, números, matrizes e cálculos de derivadas, podemos facilmente implementar métodos básicos, como o método de Newton, e o gradiente descendente. Para o método de Newton, que requer o uso das primeiras e segundas derivadas, temos duas opções. A primeira sendo calcular a matriz Hessiana diretamente, o que é custoso, mas existindo a segunda opção para cobrir o caso que é necessário mais desempenho, não é um problema. A segunda opção é caracterizada pelo uso do BFGS (método de Broyden–Fletcher–Goldfarb–Shanno) (2.7.2), podendo ser adaptado para calcular o inverso da Hessiana, já que é o inverso que é usado no método de Newton. Além disso, já existindo o cálculo pelo BFGS, podemos usar o próprio método de minimização do BFGS. Para o método do gradiente descendente, pequenas mudanças devem ser feitas ao método de busca em linha para que se possa escolher os limites para o tamanho do passo, uma vez que para o SCP, na função Lagrangiana penalizada, o tamanho do passo é limitado

entre 0 e 1.

Também existem métodos mais complexos que podemos montar a partir do que foi construído para o SCP. Foram construídas funções Lagrangianas, com e sem penalidades, e uma função de mérito, que quando analisamos a penalidade dela vemos que também tem uma forma semelhante à Lagrangiana. Dessa forma temos a habilidade de usar essas funções de maneira similar ao uso nos métodos empregados no LANCELOT (2.4.2.1) e DONLP2 (2.4.2.2).

5 Conclusão

Como efeito do estudo epistemológico da literatura acerca do SCP, partindo do que se sugere ser a origem, que podemos considerar com ancestralidade à Newton, os primeiros movimentos pós-guerra que introduzem a programação linear e suas metalinguagens. Temos uma análise tanto qualitativa quanto quantitativa, arquitetada sobre pilares de estrutura lógico-computacional, o que permite dissertar e descrever o algoritmo que é apresentado no trabalho (STILL; WESTERLUND, 2010). Também nos permitiu, uma vez acessado o ideal do método, construir em uma configuração própria, o que configura o significado de inovação, um código que traduz o algoritmo. E podemos também conjecturar que há um quê de inovador, em alguma dimensão, do algoritmo primal quando construído o código que sugerimos.

Num outro edifício fundamentado na funcionalidade e eficácia do método temos uma análise quantitativa que permeia as escalas de profundidade que pudemos alcançar com as ferramentas que tínhamos à disposição, a saber, a proposta de código ainda não finalizada.

5.1 Trabalhos Futuros

Tendo em vista o amplo potencial destacado do SCP no capítulo 4, sabemos que problemas em que a função objetivo seja convexa estão todos dentro do escopo de trabalhos futuros. Além do mais, pelo que conferimos, desde que a função objetivo seja continuamente diferenciável, e ademais, com restrições dadas por funções suaves, problemas de modelagem matemática com expressões ao menos algébricas podem ser atingidos pelo SCP. Dentre esses problemas temos o problema de N-Corpos. Especificamente o problema de encontrar configurações centrais para N suficientemente grande, com o intuito de determinar sobre a finitude das classes de configurações centrais abordando o fator simétrico.

No que se refere à questão computacional, é pretendido não só finalizar o código que propomos, como também expandir a abordagem das estimativas para a matriz hessiana. O que pode vir a simplificar o algoritmo, e como consequência quiçá amplificar, ainda mais, o leque de aplicações de problemas solúveis pelo SCP.

Para a finalização, deve ser feita a verificação de corretude para o subproblema linear (129), afim de existir uma garantia à respeito da direção de descida. Uma outra questão pendente que pode ser apresentada para uma versão completa do método é o ajusta fino do incremento das estimativas dos multiplicadores de Lagrange na função de mérito (136).

5.2 Considerações Finais

Uma plausível convergência das concatenações do que foi exposto nos capítulos 2 e 4, pode ser uma alusão à imersão das contendas tanto teóricas quanto práticas de uma área de pesquisa matemática-computacional que é a otimização. Nesses termos qualquer avanço que se possa, teórico ou prático, nessa área aponta rapidamente para uma evolução em todas as áreas que hoje fazem uso da otimização como sub-rotina de sua metodologia, o que confere aos avanços que aqui apresentamos como sugestivos a serem aplicados em curto prazo em áreas díspares, visto, por exemplo, que modelos de crescimento populacional atingem da sociologia à biomedicina com equivalente potencial de valor. E os modelos mais modernos possuem intrinsecamente como função objetivo, uma função que compete à programação não linear. Isso nos mostra que cedo ou tarde os métodos de otimização terão que ocupar o espaço de técnicas viáveis para resolver problemas de fenômenos contumaz modernos.

Como uma conclusão óbvia, o trabalho que foi realizado tanto é moderno como necessário. Ultrapassando o viés primário de evidenciar as competências de um discente em sua conclusão de graduação e afluindo para um desfiladeiro de novos horizontes e uma certeza, de que dentre todas as possibilidades, esse se apresenta em tempo como o trabalho que me preparei para realizar ao longo de minha formação acadêmica.

Referências

- ABRAMOWITZ, M.; STEGUN, I. **Handbook of Mathematical Functions**. Dover, New York. 1972. Citado na página 44.
- BONGARTZ, I. et al. Cute. **ACM Transactions on Mathematical Software**, Association for Computing Machinery (ACM), v. 21, n. 1, p. 123–160, 1995. ISSN 1557-7295. Disponível em: <<http://dx.doi.org/10.1145/200979.201043>>. Citado na página 34.
- BONNANS, J. F. et al. **Numerical optimization: theoretical and practical aspects**. [S.l.]: Springer Science & Business Media, 2006. ISBN 9783540354475. Citado na página 37.
- BROYDEN, C. G. The convergence of a class of double-rank minimization algorithms 1. general considerations. **IMA Journal of Applied Mathematics**, Oxford University Press (OUP), v. 6, n. 1, p. 76–90, 1970. ISSN 1464-3634. Disponível em: <<http://dx.doi.org/10.1093/imamat/6.1.76>>. Citado na página 46.
- BYRD, R. H. et al. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. **Mathematical Programming**, Springer, v. 100, n. 1, p. 27–48, 2003. Disponível em: <<https://doi.org/10.1007/s10107-003-0485-4>>. Citado na página 30.
- CAPLAN, P. Numerical computation of second derivatives with applications to optimization problems. **Unpublished academic report, MIT**, 2011. Citado 2 vezes nas páginas 44 e 47.
- CHENEY, E. W.; GOLDSTEIN, A. A. Newton's method for convex programming and tchebycheff approximation. **Numerische Mathematik**, Springer, v. 1, n. 1, p. 253–268, 1959. Disponível em: <<https://doi.org/10.1007/bf01386389>>. Citado na página 29.
- CONN, A. R.; GOULD, N.; TOINT, P. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. **SIAM Journal on Numerical Analysis**, SIAM, v. 28, n. 2, p. 545–572, 1991. Disponível em: <<https://doi.org/10.1137/0728030>>. Citado na página 34.
- DANTZIG, G. Linear programming and extensions. Princeton University Press, Dec 1963. Disponível em: <<http://dx.doi.org/10.1515/9781400884179>>. Citado na página 12.
- DANTZIG, G. B.; THAPA, M. N. **Linear Programming 1**. [S.l.: s.n.], 1997. ISBN 9780387948331. Citado na página 26.
- DAVIDON, W. C. Variable metric method for minimization. **SIAM Journal on Optimization**, SIAM, v. 1, n. 1, p. 1–17, 1991. Disponível em: <<https://doi.org/10.1137/0801001>>. Citado na página 45.
- DEB, K. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, Elsevier BV, v. 186, n. 2-4, p. 311–338, 2000. ISSN 0045-7825. Disponível em: <[http://dx.doi.org/10.1016/S0045-7825\(99\)00389-8](http://dx.doi.org/10.1016/S0045-7825(99)00389-8)>. Citado na página 34.

DENNIS, J.; SCHNABEL, R. B. **Numerical Methods for Unconstrained Optimization and Nonlinear Equations**. [S.l.: s.n.], 1996. ISBN 9780898713640. Citado na página 44.

FIKE, J.; ALONSO, J. The development of hyper-dual numbers for exact second-derivative calculations. **49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition**, American Institute of Aeronautics and Astronautics, 2011. Disponível em: <<http://dx.doi.org/10.2514/6.2011-886>>. Citado 2 vezes nas páginas 47 e 48.

FISCHETTI, M. **Introduction to Mathematical Optimization**. [S.l.: s.n.], 2019. ISBN 9781692792022. Citado na página 41.

FLETCHER, R. A new approach to variable metric algorithms. **The Computer Journal**, Oxford University Press (OUP), v. 13, n. 3, p. 317–322, 1970. ISSN 1460-2067. Disponível em: <<http://dx.doi.org/10.1093/comjnl/13.3.317>>. Citado na página 46.

FLETCHER, R.; POWELL, M. J. A rapidly convergent descent method for minimization. **The computer journal**, Oxford University Press, v. 6, n. 2, p. 163–168, 1963. Disponível em: <<https://doi.org/10.1093/comjnl/6.2.163>>. Citado na página 45.

GANDOMI, A. H. et al. Bat algorithm for constrained optimization tasks. **Neural Computing and Applications**, Springer Science and Business Media LLC, v. 22, n. 6, p. 1239–1255, 2012. ISSN 1433-3058. Disponível em: <<http://dx.doi.org/10.1007/s00521-012-1028-9>>. Citado na página 34.

GILL, P. et al. **Some Theoretical Properties of an Augmented Lagrangian Merit Function**. [S.l.], 1986. Citado na página 35.

GOLDFARB, D. A family of variable-metric methods derived by variational means. **Mathematics of Computation**, American Mathematical Society (AMS), v. 24, n. 109, p. 23–23, 1970. ISSN 0025-5718. Disponível em: <<http://dx.doi.org/10.1090/S0025-5718-1970-0258249-6>>. Citado na página 46.

GRIFFITH, R.; STEWART, R. A nonlinear programming technique for the optimization of continuous processing systems. **Management science**, INFORMS, v. 7, n. 4, p. 379–392, 1961. Disponível em: <<https://doi.org/10.1287/mnsc.7.4.379>>. Citado na página 29.

HALMOS, P. R. **Espaços Vetoriais de Dimensão Finita**. [S.l.]: Editora Campus, 1978. Citado na página 24.

HAN, S.; MANGASARIAN, O. L. Exact penalty functions in nonlinear programming. **Mathematical programming**, Springer, v. 17, n. 1, p. 251–269, 1979. Disponível em: <<https://doi.org/10.1007/bf01588250>>. Citado na página 35.

HAN, S. P. A globally convergent method for nonlinear programming. **Journal of optimization theory and applications**, Springer, v. 22, n. 3, p. 297–309, 1977. Disponível em: <<https://doi.org/10.1007/bf00932858>>. Citado 2 vezes nas páginas 29 e 33.

HESTENES, M. R. Multiplier and gradient methods. **Journal of optimization theory and applications**, Springer, v. 4, n. 5, p. 303–320, 1969. Disponível em: <<https://doi.org/10.1007/bf00927673>>. Citado na página 34.

HOCK, W.; SCHITTKOWSKI, K. **Test Examples for Nonlinear Programming Codes**. Springer Berlin Heidelberg, 1981. Disponível em: <<https://doi.org/10.1007/978-3-642-48320-2>>. Citado 2 vezes nas páginas 34 e 66.

JOHN, F. Extremum problems with inequalities as subsidiary conditions. In: **Traces and emergence of nonlinear programming**. Springer, 2014. p. 197–215. Disponível em: <https://doi.org/10.1007/978-3-0348-0439-4_9>. Citado na página 32.

JOINES, J.; HOUCK, C. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's. **Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence**, IEEE, 1994. Disponível em: <<http://dx.doi.org/10.1109/ICEC.1994.349995>>. Citado na página 34.

KANTOROVICH, L. V. Mathematical methods of organizing and planning production. **Management science**, INFORMS, v. 6, n. 4, p. 366–422, 1960. Disponível em: <<http://dx.doi.org/10.1287/mnsc.6.4.366>>. Citado na página 24.

KARUSH, W. Minima of functions of several variables with inequalities as side constraints. **M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago**, 1939. Disponível em: <https://doi.org/10.1007/978-3-0348-0439-4_10>. Citado na página 31.

KELLEY, J. E. An application of linear programming to curve fitting. **Journal of the Society for Industrial and Applied Mathematics**, SIAM, v. 6, n. 1, p. 15–22, 1958. Disponível em: <<https://doi.org/10.1137/0106002>>. Citado na página 28.

KELLEY, J. E. A computational approach to convex-programming. In: BLACKWELL PUBL LTD 108 COWLEY RD, OXFORD OX4 1JF, OXON, ENGLAND. **Econometrica**. [S.l.], 1959. v. 27, n. 2, p. 276–276. Citado na página 28.

KELLEY, J. E. The cutting-plane method for solving convex programs. **Journal of the society for Industrial and Applied Mathematics**, SIAM, v. 8, n. 4, p. 703–712, 1960. Disponível em: <<https://doi.org/10.1137/0108053>>. Citado na página 28.

KUHN, H. W.; TUCKER, A. W. **Nonlinear programming**. Berkeley. Univ. of California Press, 1951. Disponível em: <https://doi.org/10.1007/978-3-0348-0439-4_11>. Citado na página 31.

LAI, K. L. et al. New complex-step derivative approximations with application to second-order kalman filtering. **AIAA Guidance, Navigation, and Control Conference and Exhibit**, American Institute of Aeronautics and Astronautics, 2005. Disponível em: <<http://dx.doi.org/10.2514/6.2005-5944>>. Citado na página 47.

LANG, S. **Real Analysis**. [S.l.: s.n.], 1969. ISBN 9780201041798. Citado na página 24.

LEMARÉCHAL, C. Cauchy and the gradient method. **Doc Math Extra**, v. 251, n. 254, p. 10, 2012. Citado na página 28.

LIMA, E. **Espaços métricos**. [S.l.]: Instituto de Matemática Pura e Aplicada, CNPq, 1977. (Projeto Euclides). Citado na página 24.

LIMA, E. L. **Análise real Vol. 1**. [S.l.: s.n.], 1989. ISBN 9788524400483. Citado na página 24.

- LIMA, E. L. **Álgebra linear**. [S.l.: s.n.], 1995. ISBN 9788524400896. Citado na página 24.
- LIMA, E. L. **Análise real Vol. 2**. [S.l.: s.n.], 2021. ISBN 9788524402210. Citado na página 24.
- LYNESS, J. N.; MOLER, C. B. Numerical differentiation of analytic functions. **SIAM Journal on Numerical Analysis**, Society for Industrial & Applied Mathematics (SIAM), v. 4, n. 2, p. 202–210, 1967. ISSN 1095-7170. Disponível em: <<http://dx.doi.org/10.1137/0704019>>. Citado na página 47.
- MANGASARIAN, O. L.; FROMOVITZ, S. The fritz john necessary optimality conditions in the presence of equality and inequality constraints. **Journal of Mathematical Analysis and applications**, Elsevier, v. 17, n. 1, p. 37–47, 1967. Disponível em: <[https://doi.org/10.1016/0022-247x\(67\)90163-1](https://doi.org/10.1016/0022-247x(67)90163-1)>. Citado na página 33.
- NEMIROVSKI, A.; ROOS, K. Motzkin transposition theorem. Springer US, Boston, MA, p. 1537–1540, 2001. Disponível em: <https://doi.org/10.1007/0-306-48332-7_306>. Citado na página 33.
- PILLO, G. D. Exact penalty methods. In: **Algorithms for Continuous Optimization**. Springer, 1994. p. 209–253. Disponível em: <https://doi.org/10.1007/978-94-009-0369-2_8>. Citado na página 55.
- POLYAK, B. T. **Introduction to Optimization**. [S.l.: s.n.], 1987. ISBN 9780911575149. Citado na página 37.
- POWELL, M. J. A method for nonlinear constraints in minimization problems. **Optimization**, Academic Press, p. 283–298, 1969. Citado na página 34.
- REMEZ, E. Sur un procédé convergent d'approximations successives pour déterminer les polynômes d'approximation. **CR Acad. Sci. Paris**, v. 198, p. 2063–2065, 1934. Citado na página 29.
- SCHITTKOWSKI, K. The nonlinear programming method of wilson, han, and powell with an augmented lagrangian type line search function. **Numerische Mathematik**, Springer, v. 38, n. 1, p. 83–114, 1982. Disponível em: <<https://doi.org/10.1007/bf01395810>>. Citado na página 35.
- SCHITTKOWSKI, K. **More Test Examples for Nonlinear Programming Codes**. Springer Berlin Heidelberg, 1987. Disponível em: <<https://doi.org/10.1007/978-3-642-61582-5>>. Citado 2 vezes nas páginas 34 e 66.
- SHANNO, D. Conditioning of quasi-newton methods for function minimization. **Mathematics of Computation**, American Mathematical Society (AMS), v. 24, n. 111, p. 647–647, 1970. ISSN 0025-5718. Disponível em: <<http://dx.doi.org/10.1090/S0025-5718-1970-0274029-X>>. Citado na página 46.
- SPELLUCCI, P. Donlp2 short users guide. **Technische Universitt Darmstadt**, Citeseer, 1999. Citado na página 34.
- STILL, C.; WESTERLUND, T. Solving convex minlp optimization problems using a sequential cutting plane algorithm. **Computational Optimization and Applications**, Springer Science and Business Media LLC, v. 34, n. 1, p. 63–83, 2005. ISSN 1573-2894. Disponível em: <<http://dx.doi.org/10.1007/s10589-005-3076-x>>. Citado 2 vezes nas páginas 54 e 55.

STILL, C.; WESTERLUND, T. A linear programming-based optimization algorithm for solving nonlinear programming problems. **European Journal of Operational Research**, Elsevier BV, v. 200, n. 3, p. 658–670, 2010. Disponível em: <<https://doi.org/10.1016/j.ejor.2009.01.033>>. Citado 6 vezes nas páginas 30, 31, 54, 60, 64 e 74.

SURJANOVIC, S.; BINGHAM, D. **Virtual Library of Simulation Experiments: Test Functions and Datasets**. 2013. Disponível em: <<https://www.sfu.ca/~ssurjano/index.html>>. Citado na página 66.

TAO, T. **Analysis I**. [S.l.: s.n.], 2016. ISBN 9789811017896. Citado na página 24.

WOLFE, P. **THE RAND SYMPOSIUM ON MATHEMATICAL PROGRAMMING. LINEAR PROGRAMMING AND RECENT EXTENSIONS. PROCEEDINGS OF A CONFERENCE, MARCH 16-20, 1959**. [S.l.], 1960. Citado na página 29.

ZATELEPIN, A. **minilp - A fast linear programming solver library**. 2020. Disponível em: <<https://crates.io/crates/minilp>>. Citado na página 60.