

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Programa de Pós-Graduação em Matemática

Camila De Freitas Fernandes

NFDNA - um algoritmo para otimização não convexa e não diferenciável.

Juiz de Fora

2016

Camila De Freitas Fernandes

NFDNA - um algoritmo para otimização não convexa e não diferenciável.

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal de Juiz de Fora, na área de concentração em Matemática Aplicada, como requisito parcial para obtenção do título de Mestre em Matemática.

Orientador: Wilhelm Passarella Freire

Juiz de Fora

2016

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

Fernandes, Camila.

NFDNA - um algoritmo para otimização não convexa e não diferenciável. / Camila De Freitas Fernandes. – 2016.

63 f. : il.

Orientador: Wilhelm Passarella Freire

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Matemática, 2016.

1. Otimização Não Diferenciável. 2. Otimização Não convexa. 3. Dualidade Lagrangeana. 4. Algoritmos de Pontos Interiores e Direções Viáveis em Otimização. I. Freire, Wilhelm Passarella. NFDNA - um algoritmo para otimização não convexa e não diferenciável.

Camila De Freitas Fernandes

NFDNA - um algoritmo para otimização não convexa e não diferenciável.

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal de Juiz de Fora, na área de concentração em Matemática Aplicada, como requisito parcial para obtenção do título de Mestre em Matemática.

Aprovada em: Juiz de Fora, 08 de abril de 2016.

BANCA EXAMINADORA

Prof. Dr. Wilhelm Passarella Freire - Orientador
Universidade Federal de Juiz de Fora

Professor Dr. Sandro Rodrigues Mazorche
Universidade Federal de Juiz de Fora

Professor Dr. Alexandre Rocha Durte
IFSEMG

AGRADECIMENTOS

Hoje percebo que acabou mais uma etapa. Esse momento se iniciou há dois anos, quando ingressei nesta nova fase da minha vida acadêmica e me vi em uma nova atmosfera, encarando esse caminho como um projeto de vida. Ao finalizar esse trabalho, me reporto nesse espaço às pessoas essenciais, sem as quais esta dissertação não seria possível e dedico a elas meus sinceros agradecimentos.

Quero agradecer, em primeiro lugar, a Deus, pela força e coragem durante toda esta longa caminhada. Seu fôlego de vida em mim foi sustento e me deu coragem para questionar realidades e propor sempre um novo mundo de possibilidades.

À minha família, por acreditar e investir em mim, Pai, sua presença significou segurança e certeza de que não estou sozinha nesta jornada. Mãe, seu cuidado e sua dedicação foi que me deu, em alguns momentos, a esperança para seguir em frente. Ao meu irmão que mesmo longe foi carinhoso e presente neste momento e a minha tia Maria Inês por estar ao meu lado me apoiando. Não esquecendo do meu namorado Thiago por compartilhar comigo esses 7 anos de convivência e por sempre segurar a barra em meio aos meus nervosismos.

Ao Professor Wilhelm Passarella Freire, obrigada pela orientação segura e competente, seu estímulo constante e testemunho de seriedade, permitiram-me concretizar este estudo. Agradeço também pela compreensão a respeito dos meus limites, auxiliando-me com sua imensa sabedoria de forma imprescindível para a elaboração deste trabalho, foram valiosas suas contribuições para o meu crescimento intelectual e pessoal.

Aos professores Alexandre Duarte e Sandro Rodrigues Mazorche foi um prazer tê-los na banca examinadora. E a todos os outros professores do curso pelos conhecimentos compartilhados.

Não posso esquecer-me de meus mais fiéis amigos: Ana Paula, Ingrid, Luan Lokkerbol. É com vocês que compartilho angústias, alegrias, felicidades e tantas outras coisas das quais constroem uma amizade. Só vocês entendem o meu objetivo, o sumiço, a falta de tempo, o cansaço, a necessidade de isolamento que a escrita exige.

E por fim, meu agradecimento aos companheiros que fizeram parte da minha formação e que, com certeza, vão continuar presentes em minha vida. Principalmente ao Hernando J. Rocha, Natália E. Alves e Talita C. Santos um muito obrigada.

Dedico este trabalho aos meus pais Aluizio e Maria Virginia, ao meu irmão Guilherme e ao meu namorado Thiago pois sem eles a realização desse estudo não seria possível, um muito obrigada.

RESUMO

Neste trabalho estudamos um algoritmo para solução de problemas de otimização irrestrita com funções não necessariamente convexas ou diferenciáveis, denominado Nonsmooth Feasible Direction Nonconvex Algorithm - NFDNA, e fazemos uma aplicação deste algoritmo que consistiu em utilizá-lo como subrotina de um outro algoritmo chamado Interior Epigraph Direction (IED) method. O IED, desenvolvido para resolver problemas de otimização não convexa, não diferenciável mas com restrições, utiliza Dualidade Lagrangeana que requer a minimização da função Lagrangeana. A eficiência do IED depende fortemente de tal minimização. Como aplicação, substituímos a rotina `fminsearch` do Matlab, utilizada originalmente pelo IED, pelo NFDNA. Mostramos através da solução de problemas teste que a performance do IED foi mais eficiente com a utilização do NFDNA.

Palavras-chave: Otimização Não Diferenciável. Otimização Não Convexa. Dualidade Lagrangeana. Algoritmos de Pontos Interiores e Direções Viáveis em Otimização.

ABSTRACT

In this work we study an algorithm for solving unconstrained, not necessarily convex or differentiable optimization problems called Nonsmooth Feasible Direction Nonconvex Algorithm - NFDNA. We also employ this algorithm as a subroutine of the Interior Epigraph Directions (IED) method. The IED method, devised for solving constrained, nonconvex and nonsmooth optimization problems uses Lagrangean Duality which requires the minimization of the Lagrangean function. The effectiveness of the IED depends strongly on the Lagrangean function minimization. As an application, we replace the Matlab routine `fminsearch`, originally used by IED, with NFDNA. We show through the solution of test problems that the IED performance is more efficient by employing NFDNA.

Key-words: Nonsmooth Optimization. Nonconvex Optimization. Lagrangean Duality. Feasible Directions Interior Points Algorithms.

LISTA DE ILUSTRAÇÕES

Figura 3.1 – Direção de busca do FDIPA.	18
Figura 4.1 – Conjunto Convexo e Conjunto Não Convexo.	22
Figura 5.1 – Cálculo de um novo plano de corte viável.	31

SUMÁRIO

1	INTRODUÇÃO	9
2	CONCEITOS PRELIMINARES	12
2.1	Notações	12
2.2	Definições Gerais	13
3	FDIPA	15
3.1	Método do FDIPA	16
3.2	Algoritmo	19
4	OTIMIZAÇÃO CONVEXA NÃO DIFERENCIÁVEL E O NFDA	21
4.1	Método	25
4.2	Algoritmo	27
5	NFDNA	29
5.1	Método - NFDNA	29
5.2	Algoritmo - NFDNA	32
5.3	Análise de Convergência	34
5.4	Resultados Numéricos	42
6	APLICAÇÃO	45
6.1	Problemas Testes Utilizados	47
6.2	Experimentos Numéricos	52
7	Conclusão	60
	REFERÊNCIAS	61

1 INTRODUÇÃO

Tal como em muitos outros ramos da Matemática, a otimização teve a sua origem nas aplicações. Embora, no seu caso, não seja preciso recuar muitos anos para identificar as aplicações que impulsionaram o seu desenvolvimento. De fato, a história dos principais resultados de otimização é surpreendentemente curta. Surpreendente porque a otimização aparenta ser uma questão tão natural num contexto real como num contexto abstrato, como é o da Matemática.

Mas, enquanto que a resolução de equações, nas mais variadas formas, permaneceu central para a Matemática, pouca relevância foi dada à resolução de inequações.

No final da década de 30 e início da década de 40 do século *XX* apareceram resultados que se podem considerar hoje como inspiradores do franco desenvolvimento da otimização. Pode-se até perceber porque é que esses desenvolvimentos ocorreram nessa época. Aparentemente, a situação de guerra e competição que se vivia, criou condições para que se procurasse desempenhar as tarefas melhor e mais rápido.

No final da década de 50, este ramo da Matemática ficou mais conhecido por Programação Matemática. Do ponto de vista da Matemática, este movimento causou o aparecimento de uma grande variedade de teoremas e até mesmo teorias. Surgiram estudos mais aprofundados de conceitos como os de sistemas de inequações, poliedros e dualidade. E, ao mesmo tempo, muitos outros resultados de teoria de convexidade foram desenvolvidos ou especializados de forma tal que a convexidade é, hoje, uma parte fundamental da otimização.

A Programação Matemática estimulou o estudo de fenômenos econômicos e, claramente, também havia a interação com as ciências de computação. Os computadores possibilitaram a execução de alguns métodos em grandes problemas. Por outro lado, outros métodos revelaram fragilidades numéricas na computação e originaram questões de complexidade computacional.

O termo otimização refere-se ao estudo de problemas em que se busca minimizar ou maximizar uma função por meio da escolha dos valores de variáveis dentro de um conjunto viável.

Hoje existem diversos métodos de otimização e desta forma a utilização de cada método depende do tipo de problema em estudo, ou seja, depende se o problema será

- (i) Linear ou não linear;
- (ii) Restrito ou irrestrito;
- (iii) Diferenciável ou não diferenciável;

(iv) Convexo ou não convexo.

A teoria clássica de otimização presume certa diferenciabilidade e fortes hipóteses de regularidade como pode ser visto em Fletcher [14]. Contudo, estas hipóteses estão longe de acontecer na prática, onde em muitos casos a própria física do problema impõe um modelo não diferenciável. Problemas de otimização não diferenciáveis aparecem em muitos campos de aplicação, como por exemplo, em Mecânica [28], Economia [30], Controle Ótimo [12].

Para problemas onde não se tem a diferenciabilidade da função, existe uma área da Programação Matemática denominada Otimização Não Diferenciável que utiliza técnicas da Análise Convexa que substituem o Cálculo Diferencial clássico.

Existem grandes dificuldades quando se lida com funções não diferenciáveis e, em muitos casos, essas funções tem mínimo exatamente onde o gradiente não está definido. Ratificando tal ideia, não é preciso ir muito longe para entender melhor as dificuldades causadas pela não diferenciabilidade. Para tanto, basta considerar a função valor absoluto $f(x) = |x|$, com $x \in \mathbb{R}$, nota-se que f não é diferenciável na origem, justamente o ponto onde ocorre o mínimo. Então, se para esta função, for aplicado qualquer método de otimização diferenciável, este nem se quer reconhecerá o ponto de mínimo $\bar{x} = 0$, pois nestes métodos o mínimo precisa satisfazer o famoso resultado de Fermat

$$\nabla f(\bar{x}) = 0$$

que é, na verdade, uma condição necessária de otimalidade.

Diante do exposto, vem sendo desenvolvidos uma série de métodos para lidar com este tipo de problema como pode ser visto, por exemplo, nos trabalhos de Kelley, Kiwiel, Lemaréchal e Mäkelä, entre outros.

Os métodos considerados mais eficientes e confiáveis quando a função objetivo é convexa, são os Métodos de Feixes (veja [3] ou [20]), os quais são baseados na teoria de subdiferencial desenvolvida por Rockafellar [25] e Clarke [10]. O que estes métodos tem em comum é que, em cada iteração, requerem a utilização de um subgradiente além do valor da função objetivo. A ideia básica é aproximar o subdiferencial (que é o conjunto dos subgradientes), usando informações armazenadas em iterações anteriores no chamado feixe.

Problemas envolvendo funções não convexas e não diferenciáveis são mais difíceis de lidar. Contudo, um passo crucial realizado nessa área foi a tese de doutorado de Clarke em 1973. Além disso, sabe-se que um cálculo eficiente, bem como condições de otimalidade aplicáveis, em termos de construções convexas locais, pode ser desenvolvido para a classe de funções localmente Lipschitz contínuas.

Neste trabalho são apresentadas duas alternativas de algoritmos para Otimização Não Diferenciável: o primeiro algoritmo chamado NFDA - Nonsmooth Feasible Direction Algorithm inicialmente proposto por Freire [15] em sua tese de doutorado, considera o caso convexo sem restrições. Em seguida, estudaremos um algoritmo, denominado NFDNA - Nonsmooth Feasible Direction Nonconvex Algorithm, proposto por Tanaka [35], que é uma extensão do NFDA, para a solução de problemas irrestritos cujas funções objetivo são não convexas.

Os métodos são híbridos, neles são utilizadas algumas ideias do método clássico de planos de corte de Kelley [21], para realizar aproximações das funções não diferenciáveis. Em cada iteração, utiliza-se o FDIPA, um método de pontos interiores e direções viáveis, desenvolvido por Herskovits em [19], para gerar direções de descida viáveis.

Outra contribuição deste trabalho é a aplicação do NFDNA ao IED - Interior Epigraph Directions Algorithm que foi apresentado por Burachik, Freire e Kaya [5] para a solução de problemas de otimização não convexa, não diferenciável, com restrições.

Este trabalho está organizado da seguinte forma: No Capítulo 2, reservados as preliminares, são definidas as notações e alguns resultados básicos. No Capítulo 3 apresentamos o Feasible Direction Interior Point Algorithm (FDIPA) proposto por Herskovits [19]. No capítulo 4 apresentamos o Nonsmooth Feasible Directions Algorithm (NFDA) para resolução de problemas não diferenciáveis e convexas. No Capítulo 5 apresentamos o Nonsmooth Feasible Directions Nonconvex Algorithm (NFDNA) para a solução de problemas de otimização não diferenciáveis e não convexas. No Capítulo 6 fazemos uma aplicação do NFDNA e, finalmente, no capítulo 7 são apresentadas as conclusões e os possíveis trabalhos futuros.

2 CONCEITOS PRELIMINARES

Nos próximos capítulos desta tese serão apresentados alguns métodos de otimização, nos quais usaremos as notações, os conceitos e os resultados básicos da otimização apresentados a seguir. Deixaremos para cada capítulo as notações específicas utilizadas.

2.1 Notações

1. \mathbb{R} é o conjunto dos números reais.
2. $\mathbb{N} = \{1, 2, 3, \dots\}$ é o conjunto dos números naturais.
3. $\mathbb{R}_{++} = \{x \in \mathbb{R} \mid x > 0\}$.
4. $\mathbb{R}^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in \mathbb{R}, i = 1, 2, \dots, n\}$ é o espaço euclidiano n-dimensional.
5. $\mathbb{R}_{++}^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in \mathbb{R}, x_i > 0, i = 1, 2, \dots, n\}$.
6. $x \in \mathbb{R}^n$ será

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

7. $x^t = (x_1, x_2, \dots, x_n)$ é denominado vetor transposto.
8. O produto interno de dois vetores $x, y \in \mathbb{R}^n$ será dado por

$$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \dots + x_n y_n = x^t y$$

9. A norma euclidiana é $\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$, em particular, se $n = 1$, escreveremos $\|x\| = |x|$.
10. Se $X \subset \mathbb{R}^n$, X° é o interior de X , ou seja,

$$X^\circ = \{a \in X \mid \exists \varepsilon > 0; B(a, \varepsilon) \subset X\}$$

onde $B(a, \varepsilon) = \{y \in \mathbb{R}^n \mid \|y - a\| < \varepsilon\}$ é a bola aberta de centro em a e raio ε .

11. $X^c = \mathbb{R}^n - X$ é denominado complementar de X .
12. O vetor gradiente da função $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ em $a \in X$ será

$$\nabla f(a) = \left(\frac{\partial f}{\partial x_1}(a), \frac{\partial f}{\partial x_2}(a), \dots, \frac{\partial f}{\partial x_n}(a) \right)^t$$

13. Dada a função $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ a matriz hessiana de f será

$$Hf(a) = \nabla^2 f(a) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(a) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(a) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(a) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(a) & \frac{\partial^2 f}{\partial x_2^2}(a) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(a) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(a) & \frac{\partial^2 f}{\partial x_n \partial x_2}(a) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(a) \end{pmatrix}$$

14. A matriz jacobiana da função $g : X \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ no ponto $a \in X$ é indicada por

$$Jg(a) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1}(a) & \frac{\partial g_1}{\partial x_2}(a) & \cdots & \frac{\partial g_1}{\partial x_n}(a) \\ \frac{\partial g_2}{\partial x_1}(a) & \frac{\partial g_2}{\partial x_2}(a) & \cdots & \frac{\partial g_2}{\partial x_n}(a) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(a) & \frac{\partial g_m}{\partial x_2}(a) & \cdots & \frac{\partial g_m}{\partial x_n}(a) \end{pmatrix}$$

15. Nos algoritmos que apresentaremos d indicará a direção de busca, e t denotará o passo dado na direção d .

2.2 Definições Gerais

Definição 2.1. Diremos que a função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é **diferenciável** em $a \in \mathbb{R}^n$ quando existirem as derivadas parciais $\frac{\partial f}{\partial x_1}(a), \dots, \frac{\partial f}{\partial x_n}(a)$ e para todo vetor $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ tivermos

$$f(a + v) = f(a) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(a) v_i + r(v) \text{ com } \lim_{v \rightarrow 0} \frac{r(v)}{\|v\|} = 0$$

Definição 2.2. Um ponto $x \in \mathbb{R}^n$ é um **mínimo local** de f , se existir $\varepsilon > 0$ tal que $f(x) < f(y)$ para todo $y \in B(x, \varepsilon)$.

Definição 2.3. Um ponto $x \in \mathbb{R}^n$ é um **mínimo global** de f , se satisfaz $f(x) < f(y)$ para todo $y \in \mathbb{R}^n$.

Definição 2.4. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Uma direção $d \in \mathbb{R}^n$ é dita uma **direção de descida** para f em $a \in \mathbb{R}^n$ se existe $\varepsilon > 0$ tal que $f(a + td) < f(a)$, para todo $t \in (0, \varepsilon)$.

Definição 2.5. Um vetor $d \in \mathbb{R}^n$ é uma **direção viável** em $x \in X$ se existe $\tau > 0$ tal que $x + td \in X$, para todo $t \in (0, \tau)$.

Definição 2.6. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$. A **derivada direcional** de f no ponto $a \in \mathbb{R}^n$ na direção do vetor $v \in \mathbb{R}^n$ é definida por

$$f'(a, v) = \lim_{t \rightarrow 0^+} \frac{f(a + tv) - f(a)}{t}$$

Definição 2.7. O **epígrafo** da função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é o conjunto

$$epif = \{(x, r) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq r\}.$$

Definição 2.8. *Seja o problema de minimização com restrições de desigualdade*

$$(P) = \begin{cases} \text{minimizar} & f(x) \\ \text{sujeito a} & g(x) \leq 0 \end{cases}$$

com $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ dada por $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^t$. As funções $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, 2, \dots, m$, são chamadas **restrições do problema**. Uma restrição $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ será dita **ativa** no ponto $a \in \mathbb{R}^n$ se $g_i(a) = 0$.

Definição 2.9. *A função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é **localmente Lipschitziana** com constante K em $a \in \mathbb{R}^n$ se existe $r > 0$ tal que*

$$|f(y) - f(x)| \leq K\|y - x\|$$

para todo $x, y \in B(a, r)$.

3 FDIPA

Em algoritmos empregados na resolução de otimização não linear, a função objetivo e/ou pelo menos uma das restrições é não linear. A otimização não linear diferenciável se apoia fortemente no conceito de diferenciabilidade. Neste capítulo, apresentaremos alguns resultados importantes para a teoria e um algoritmo para solução de problemas de otimização não linear diferenciável, cuja direção de busca é amplamente utilizada pelos métodos que serão apresentados neste trabalho.

Teorema 3.1. *Sejam $v \in \mathbb{R}^n$ e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável em $a \in \mathbb{R}^n$. Então*

$$f'(a, v) = v^t \nabla f(a).$$

Teorema 3.2. *Sejam $d \in \mathbb{R}^n$ e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável em $a \in \mathbb{R}^n$. Se $d^t \nabla f(a) < 0$ então d é uma direção de descida para f em a .*

Teorema 3.3. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável em $a \in \mathbb{R}^n$. Se a é um ponto de mínimo local de f então $\nabla f(a) = 0$.*

Consideremos agora o seguinte problema de otimização não linear diferenciável com restrições de desigualdade:

$$(P_{2.1}) \begin{cases} \text{minimizar} & f(x) \\ \text{sujeito a} & g(x) \leq 0 \end{cases}$$

onde as funções $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $g = (g_1, g_2, \dots, g_p) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ são funções continuamente diferenciáveis em \mathbb{R}^n e f ou pelo menos uma das g_i é não linear.

Denota-se por $\Omega = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$ o conjunto viável e introduzindo a variável auxiliar $\lambda \in \mathbb{R}^m$ chamada variável dual ou multiplicador de Lagrange, define-se a função Lagrangeana associada ao problema $(P_{2.1})$

$$L(x, \lambda) = f(x) + \lambda^t g(x)$$

cuja matriz hessiana é dada por

$$H(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 g_i(x).$$

Representa-se por $I(x) = \{i \mid g_i(x) = 0\}$ o conjunto de índices cujas restrições são ativas e diz-se que x é um **ponto regular** se os vetores $\nabla g_i(x)$ para $i \in I(x)$ forem linearmente independentes.

Teorema 3.4. *Seja x^* um ponto regular para o problema $(P_{2.1})$. Se x^* é um mínimo local de $(P_{2.1})$ então existe $\lambda^* \in \mathbb{R}^m$ tal que*

$$\begin{aligned} \nabla f(x^*) + Jg(x^*)^t \lambda^* &= 0 \\ G(x^*) \lambda^* &= 0 \\ \lambda^* &\geq 0 \\ g(x^*) &\leq 0 \end{aligned} \tag{3.1}$$

onde $G(x) = \text{diag}(g_1(x), g_2(x), \dots, g_m(x))$.

Agora apresentaremos um algoritmo que foi proposto por José Herskovits em [17, 19] para resolver problemas de otimização não linear diferenciável com restrições de desigualdade, denominado Feasible Direction Interior Point Algorithm (FDIPA).

De acordo com Herskovits [19], o FDIPA converge globalmente para pontos de Karush-Kuhn-Tucker, é de fácil implementação, robusto e eficiente. Não necessita da solução de subproblemas quadráticos e não se trata de um método de penalidades, barreira ou filtros.

O algoritmo, a partir de um ponto inicial no interior da região viável, gera uma sequência de pontos também interiores à região. Em cada iteração, é calculada uma direção de descida resolvendo-se um sistema de equações lineares nas variáveis primal e dual com a mesma matriz seguida de uma busca linear inexata.

Em seguida, o sistema linear é perturbado para produzir uma deflexão na direção de descida no sentido do interior da região viável de modo a obter uma direção de descida viável. Uma busca linear é então realizada para obter um novo ponto interior e garantir a convergência global do método.

3.1 Método do FDIPA

Considerando-se o sistema de equações lineares extraídas de (3.1)

$$\nabla f(x) + Jg(x)^t \lambda = 0 \tag{3.2}$$

$$G(x) \lambda = 0 \tag{3.3}$$

fazendo

$$y = \begin{pmatrix} x \\ \lambda \end{pmatrix} \quad \text{e} \quad \Phi(y) = \begin{pmatrix} \nabla f(x) + Jg(x)^t \lambda \\ G(x) \lambda \end{pmatrix}$$

obtemos

$$J\Phi(y) = \begin{pmatrix} H(x, \lambda) & Jg(x)^t \\ \Lambda \nabla g(x) & G(x) \end{pmatrix}$$

onde $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$.

Uma iteração de Newton para resolver o sistema de equações lineares $\Phi(y) = 0$, com ponto $y^k = (x^k, \lambda^k)$ na iteração k , define um ponto $y^{k+1} = (x^{k+1}, \lambda^{k+1})$, solução do sistema linear $J\Phi(y^k)(y - y^k) = -\Phi(y^k)$ que pode ser escrito como

$$\begin{pmatrix} B^k & Jg(x^k)^t \\ \Lambda^k Jg(x^k) & G(x^k) \end{pmatrix} \begin{pmatrix} x - x^k \\ \lambda - \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla f(x^k) + Jg(x^k)^t \lambda^k \\ G(x^k) \lambda^k \end{pmatrix} \quad (3.4)$$

onde B^k é uma matriz simétrica de ordem n que pode ser a própria hessiana $H(x^k, \lambda^k)$ ou uma aproximação dela obtida por alguma técnica quasi-Newton ou a matriz identidade. Contudo, B^k deve ser definida positiva para que se garanta a convergência global do algoritmo.

Definindo-se $d = x - x^k$ pode-se reescrever o sistema (3.4) da seguinte maneira

$$B^k d + Jg(x^k)^t \lambda = -\nabla f(x^k) \quad (3.5)$$

$$\Lambda^k Jg(x^k) d + G(x^k) \lambda = 0 \quad (3.6)$$

A solução (d_1^k, λ_1^k) deste sistema fornece uma direção d_1^k e uma nova estimativa λ_1^k para λ .

Herskovits provou em [19] que d_1^k é uma direção de descida para a função f , ou seja, $(d_1^k)^t \nabla f(x^k) < 0$. Entretanto, d_1^k pode não ser uma direção viável pois, quando alguma restrição se aproxima de zero, d_1^k tende a uma direção tangente ao conjunto viável. De fato, reescrevendo a equação (3.6) temos

$$\lambda_i^k \nabla g_i(x^k) d_1^k + g_i(x^k) \lambda_{1i}^k = 0, \quad \forall i = 1, 2, \dots, m$$

o que implica em $\nabla g_i(x^k) d_1^k = 0$ para todo i tal que $g_i(x^k) = 0$.

Para evitar esse problema, adiciona-se um vetor negativo no lado direito de (3.6) e define-se um novo sistema em d e $\bar{\lambda}$

$$B^k d + Jg(x^k)^t \bar{\lambda} = -\nabla f(x^k) \quad (3.7)$$

$$\Lambda Jg(x^k) d + G(x^k) \bar{\lambda} = -\rho_k \Lambda^k \omega^k \quad (3.8)$$

onde $\rho_k \in \mathbb{R}_{++}$, $\omega^k \in \mathbb{R}_{++}^n$, d^k é a nova direção e $\bar{\lambda}^k$ é a nova estimativa de λ .

Neste caso, (3.8) é equivalente a

$$\lambda_i^k \nabla g_i(x^k)^t d^k + g_i(x^k) \bar{\lambda}_i^k = -\rho_k \lambda_i^k \omega_i^k, \quad i = 1, 2, \dots, m$$

e conseqüentemente

$$\nabla g_i(x^k)^t d^k = -\rho_k \omega_i^k < 0$$

para as restrições ativas. Logo d^k é uma direção viável.

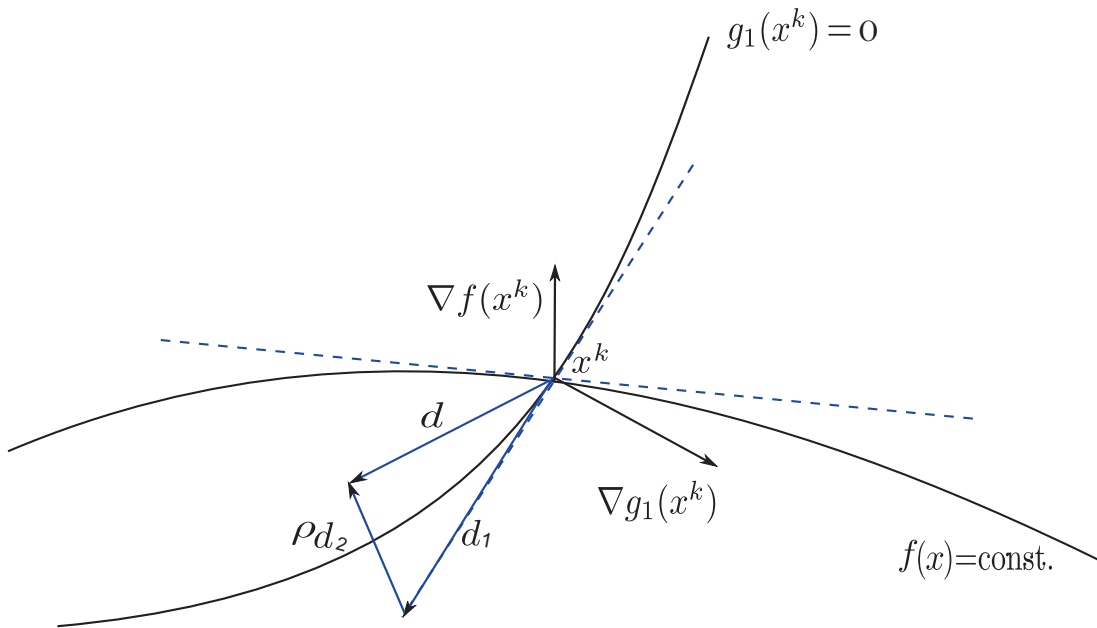


Figura 3.1 – Direção de busca do FDIPA.

O termo $-\rho_k \Lambda^k \omega^k$ em (3.8) produz uma deflexão em d_1^k na direção do interior da região viável. O problema que surge é que, ao contrário de d_1^k , d^k pode não ser uma direção de descida para f . Entretanto, essa propriedade pode ser mantida se ρ_k for escolhido convenientemente. Observa-se que a direção de descida d^k e $\bar{\lambda}^k$ podem ser obtidos resolvendo-se os sistemas

$$\begin{cases} B^k d_1 + Jg(x^k)^t \lambda_1 = -\nabla f(x^k) \\ \Lambda^k Jg(x^k) d_1 + G(x^k) \lambda_1 = 0 \end{cases}, \quad \begin{cases} B^k d_2 + Jg(x^k)^t \lambda_2 = 0 \\ \Lambda^k Jg(x^k) d_2 + G(x^k) \lambda_2 = -\Lambda^k \omega^k \end{cases}$$

e define-se $d^k = d_1^k + \rho_k d_2^k$, $\bar{\lambda}^k = \lambda_1^k + \rho_k \lambda_2^k$. Desta forma tem-se

$$(d^k)^t \nabla f(x^k) = (d_1^k)^t \nabla f(x^k) + \rho_k (d_2^k)^t \nabla f(x^k).$$

Mas, como já visto, $(d_1^k)^t \nabla f(x^k) < 0$. Logo, se $(d_2^k)^t \nabla f(x^k) \leq 0$ então

$$(d^k)^t \nabla f(x^k) < 0, \quad \forall \rho_k \in \mathbb{R}_{++}.$$

Portanto, ρ_k deve ser convenientemente escolhido no caso em que $(d_2^k)^t \nabla f(x^k) > 0$. Neste caso, impondo-se $(d^k)^t \nabla f(x^k) \leq \xi (d_1^k)^t \nabla f(x^k)$ com $\xi \in (0, 1)$ obtém-se

$$(d_1^k)^t \nabla f(x^k) + \rho_k (d_2^k)^t \nabla f(x^k) \leq \xi (d_1^k)^t \nabla f(x^k) \quad \text{ou} \quad \rho_k \leq (\xi - 1) \frac{(d_1^k)^t \nabla f(x^k)}{(d_2^k)^t \nabla f(x^k)}.$$

Herskovits provou em [19] que se ρ_k for escolhido de maneira a satisfazer a desigualdade acima, d^k será uma direção de descida para f . A decomposição do sistema formado

pelas equações (3.7)-(3.8) nos dois últimos sistemas permite o cálculo de ρ_k . Além disso, após o cálculo de d_1^k pelo primeiro sistema, obtém-se o teste de parada do algoritmo.

Apresentamos a seguir o FDIPA.

3.2 Algoritmo

Parâmetros: $\xi \in (0, 1)$, $\eta \in (0, 1)$, $\varphi > 0$ e $\nu \in (0, 1)$.

Dados iniciais: $x \in \Omega^\circ$, $\lambda \in \mathbb{R}_{++}^m$, $\omega \in \mathbb{R}_{++}^m$, $B \in \mathbb{R}^{n \times m}$ simétrica definida positiva.

Passo 1

1. Calcule d_1 e λ_1 resolvendo o sistema linear

$$\begin{cases} Bd_1 + Jg(x)^t \lambda_1 = -\nabla f(x) \\ \Lambda Jg(x)d_1 + G(x)\lambda_1 = 0 \end{cases}$$

se $d_1 = 0$ pare.

2. Calcule d_2 e λ_2 resolvendo o sistema linear

$$\begin{cases} Bd_2 + Jg(x)^t \lambda_2 = 0 \\ \Lambda Jg(x)d_2 + G(x)\lambda_2 = -\Lambda\omega \end{cases}$$

3. Se $d_2 \nabla f(x) > 0$ faça

$$\rho = \min \left\{ \varphi \|d_1\|^2, (\xi - 1) \frac{(d_1)^t \nabla f(x)}{(d_2)^t \nabla f(x)} \right\}$$

caso contrário, faça $\rho = \varphi \|d_1\|^2$.

4. Calcule a direção de descida $d = d_1 + \rho d_2$ e $\bar{\lambda} = \lambda_1 + \rho \lambda_2$.

Passo 2 Calcule t , o maior tamanho da sequência $\{1, \nu, \nu^2, \nu^3, \dots\}$ que satisfaz

$$\begin{cases} f(x + td) \leq f(x) + \eta t \nabla f(x)^t d & \text{e } g_i(x + td) < 0 & \text{se } \lambda_i \geq 0 \\ f(x + td) \leq f(x) + \eta t \nabla f(x)^t d & \text{e } g_i(x + td) \leq g_i(x) & \text{caso contrário.} \end{cases}$$

Passo 3

1. Faça $x = x + td$.
2. Atualize B simétrica positiva definida, $\omega \in \mathbb{R}_{++}^m$ e $\lambda \in \mathbb{R}_{++}^m$ que define Λ .
3. Vá para o Passo 1.

Esse algoritmo é a versão básica do FDIPA. Na verdade, obtém-se uma família de algoritmos estabelecendo-se diferentes regras para a atualização da matriz B e dos vetores λ e ω . Algumas dessas regras encontram-se com detalhes em [18, 19]. A prova de convergência global, a extensão para o caso em que aparecem restrições de igualdade, bem como testes numéricos que comprovam a eficiência desse algoritmo podem ser encontrados em [19].

4 OTIMIZAÇÃO CONVEXA NÃO DIFERENCIÁVEL E O NFDA

A otimização pode ser dividida em diferenciável e não diferenciável. A otimização diferenciável trabalha com problemas cujas funções possuem derivadas em todos os pontos do seu domínio, enquanto que a otimização não diferenciável trabalha com problemas cujas funções não possuem derivadas em alguns pontos do seu domínio.

Para resolver um problema de otimização convexa não diferenciável utilizam-se técnicas que substituem o cálculo diferencial clássico. Tais técnicas são derivadas de uma área da Matemática chamada Análise Convexa. Estes tipos de problemas tem recebido uma grande atenção e vários novos métodos têm sido propostos.

Os métodos de otimização, na maioria das vezes, utilizam a fórmula $x^{k+1} = x^k + t^k d^k$ para atualizar o ponto corrente x^k , onde t^k é o tamanho do passo e d^k a direção de busca.

Na otimização diferenciável clássica, geralmente, os métodos de minimização usados são baseados na aproximação de primeira e segunda ordem envolvendo, respectivamente, o gradiente ∇f e a hessiana Hf da função objetivo f .

Na otimização não diferenciável, existem pontos do domínio onde a função não possui gradiente e, obviamente, nem matriz hessiana. Desta forma, aplicar métodos clássicos para resolver problemas diferenciáveis em problemas não diferenciáveis fica inviável. O que se pode fazer é adaptar os métodos clássicos para funções não diferenciáveis. Por exemplo, os Métodos Gradientes podem ser modificados para funções não diferenciáveis utilizando a direção oposta a um subgradiente como a direção de busca. Esses métodos, denominados Métodos Subgradientes, elaborados por Shor [34] foram os primeiros métodos de otimização não diferenciável. Eles possuem uma estrutura extremamente simples, mas em geral, não apresentam bons resultados numéricos [22, 34].

Nos Métodos Subgradientes, a direção oposta ao subgradiente não é necessariamente uma direção de descida e, mesmo se for, não se tem garantia de convergência do método. Para contornar essa dificuldade, uma opção é escolher o tamanho do passo t^k convenientemente.

Um outro problema encontrado nos Métodos Subgradientes é a necessidade de se conhecer o subdiferencial $\partial f(x^k)$ da função f no ponto corrente x^k . Para vencer tal dificuldade, os Métodos ε -Subgradientes recorrem a, não apenas um subgradiente no ponto corrente, mas a subgradientes de pontos de uma vizinhança do ponto corrente.

Os Métodos de Feixes, primeiramente propostos por Mifflin [27] e Lemaréchal [24], são atualmente os mais utilizados, como sugere a enorme variedade de trabalhos sobre este tema, nestes métodos, a direção de busca é obtida através da combinação convexa de subgradientes gerados em iterações anteriores. Ao contrário dos Métodos Subgradientes, nos Métodos de Feixes a escolha do passo envolve uma busca linear inexata

que produz uma solução melhor (passo sério) ou uma solução que é rejeitada (passo nulo). Independentemente do casos, um novo subgradiente é calculado e acrescentado ao feixe para encontrar uma nova direção de busca.

Uma diferença importante entre os Métodos de Feixes e os Métodos Subgradientes é que, contrariamente a este último, os Métodos Tipo Feixes geram uma sequência de iterações para que os valores da função objetivo sejam monótonos decrescentes. Por esta razão, esses métodos são classificados como *Métodos de Descida*. No entanto, uma dificuldade encontrada nos Métodos de Feixes é que eles exigem a solução de um subproblema quadrático em cada iteração para encontrar a direção de busca, e isto pode tornar-se bastante caro, em particular para problemas maiores.

Na otimização existem muitos problemas nos quais não se tem a diferenciabilidade das funções envolvidas. Entretanto, se tais funções possuírem certas propriedades como, por exemplo, a convexidade, pode-se contar com resultados que, de certa forma, substituem a diferenciabilidade. Desta forma apresentaremos, a seguir, alguns conceitos básicos sobre convexidade de conjuntos e funções. Um estudo mais detalhado pode ser encontrado em [31, 25 e 30].

Definição 4.1. *Seja $C \subset \mathbb{R}^n$. Dizemos que C é um **conjunto convexo** quando*

$$\lambda x + (1 - \lambda)y \in C, \text{ para todo } x, y \in C \text{ e } \lambda \in [0, 1].$$

Geometricamente, significa que, todo segmento de reta que une os pontos x e y está inteiramente contido em C sempre que $x, y \in C$.

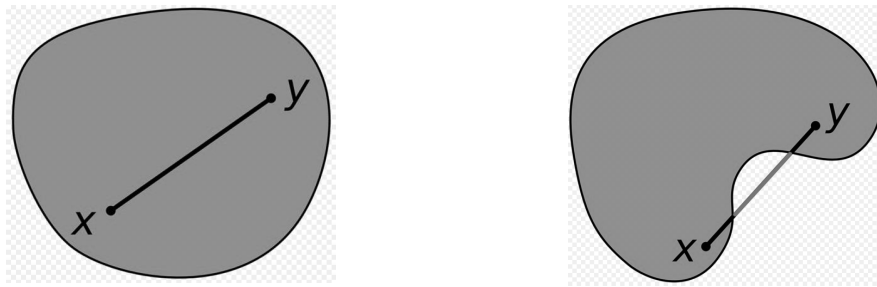


Figura 4.1 – Conjunto Convexo e Conjunto Não Convexo.

Abaixo, dois fatos básicos envolvendo conjuntos convexos:

- (i) Se $C_1, C_2 \in \mathbb{R}^n$ são conjuntos convexos e $\lambda_1, \lambda_2 \in \mathbb{R}$, então $\lambda_1 C_1 + \lambda_2 C_2$ também é convexo.
- (ii) A interseção de conjuntos convexos é convexo, isto é, $C_i \in \mathbb{R}^n$ são conjuntos convexos para $i = 1, 2, \dots, m$, então sua interseção $\bigcap_{i=1}^m C_i$ é um conjunto convexo.

Definição 4.2. Seja $f : C \rightarrow \mathbb{R}$ uma função definida no conjunto $C \in \mathbb{R}^n$

1. f é convexa se

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \text{ para todo } x, y \in C \text{ e } \lambda \in [0, 1].$$

2. f é estritamente convexa se

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \text{ para todo } x, y \in C \text{ e } \lambda \in [0, 1].$$

Teorema 4.3. Sejam $C \in \mathbb{R}^n$ convexo e $f : C \rightarrow \mathbb{R}$ uma função convexa. Todo mínimo local de f é também um mínimo global. Se f for estritamente convexa, então todo mínimo local é o único mínimo global de f em C .

Teorema 4.4. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa. Então, para todo $x \in \mathbb{R}^n$, f é localmente Lipschitziana em x .

Teorema 4.5. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convexa com constante K em $x \in \mathbb{R}^n$. Então

1. A derivada direcional em cada direção $v \in \mathbb{R}^n$ é

$$f'(x, v) = \inf_{t>0} \frac{f(x + tv) - f(x)}{t}.$$

2. A função $v \mapsto f'(x, v)$ é positivamente homogênea e subaditiva em \mathbb{R}^n com

$$|f'(x, v)| \leq K\|v\|.$$

3. $f'(x, v)$ é semicontínua superiormente enquanto função de (x, v) e Lipschitziana com constante K enquanto função de v .

4. $-f'(x, -v) \leq f'(x, v)$.

Definição 4.6. O subdiferencial da função convexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$ no ponto $a \in \mathbb{R}^n$ é o conjunto

$$\partial f(a) = \{s \in \mathbb{R}^n \mid f(x) \geq f(a) + \langle s, x - a \rangle, \forall x \in \mathbb{R}^n\}$$

e um vetor $s \in \partial f(a)$ é chamado um subgradiente de f em a .

Teorema 4.7. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convexa. Então, em todo $x \in \mathbb{R}^n$ tem-se

1. $f'(x, v) = \max \{\langle s, v \rangle \mid s \in \partial f(x), \forall x \in \mathbb{R}^n\}$.

2. $\partial f(x) = \{s \in \mathbb{R}^n \mid f'(x, v) \geq \langle s, v \rangle, \forall v \in \mathbb{R}^n\}$.

3. $\partial f(x)$ é um conjunto não vazio, convexo e compacto tal que $\partial f(x) \subset B(0, K)$, sendo K a constante de Lipschitz de f em x .

Teorema 4.8. *Se $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é convexa e diferenciável em $x \in \mathbb{R}^n$, então*

$$\partial f(x) = \{\nabla f(x)\}$$

Definição 4.9. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa. A ε -derivada direcional de f em $a \in \mathbb{R}^n$ na direção de $v \in \mathbb{R}^n$ é definida por*

$$f'_\varepsilon(a, v) = \inf_{t>0} \frac{f(a + tv) - f(a) + \varepsilon}{t}.$$

Definição 4.10. *Seja $\varepsilon > 0$. O ε -subdiferencial da função convexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$ em $a \in \mathbb{R}^n$ é o conjunto*

$$\partial_\varepsilon f(a) = \{s \in \mathbb{R}^n \mid f(x) \geq f(a) + \langle s, x - a \rangle - \varepsilon, \forall x \in \mathbb{R}^n\}$$

cada elemento $s \in \partial_\varepsilon f(a)$ é chamado ε -subgradiente de f em a .

Teorema 4.11. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa com constante de Lipschitz K .*

1. $f'_\varepsilon(x, v) = \max \{s^t v \mid s \in \partial_\varepsilon f(x), \forall v \in \mathbb{R}^n\}$.
2. $\partial_\varepsilon f(x) = \{s \in \mathbb{R}^n \mid f'_\varepsilon(x, v) \geq s^t v, \forall v \in \mathbb{R}^n\}$.
3. $\partial_\varepsilon f(x)$ é um conjunto não vazio, convexo e compacto tal que $\|s\| \leq K, \forall s \in \partial_\varepsilon f(x)$.

Teorema 4.12. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa. As seguintes condições são equivalentes:*

1. f atinge seu mínimo global em a .
2. $0 \in \partial f(a)$.

No capítulo anterior vimos um método para resolver problemas de otimização não linear diferenciável. Como já vimos, a não diferenciabilidade gera uma série de dificuldades que acarretam na inviabilidade da aplicação dos algoritmos para problemas diferenciáveis em problemas não diferenciáveis. Além das dificuldades já mencionadas, uma outra é estabelecer um critério de parada que, na maioria das vezes, não fica bem definido. Por exemplo, um critério de parada que utiliza $\|\nabla f(x^k)\| < \varepsilon$ para algum $\varepsilon > 0$ fixado, não pode ser usado em problemas não diferenciáveis. Para ver isto basta considerar $f : \mathbb{R} \rightarrow \mathbb{R}$ definida por $f(x) = |x|$. Tem-se $\|\nabla f(x^k)\| = 1, \forall x \neq 0$, não importando o quanto x^k esteja próximo da solução $x^* = 0$.

Desta forma, apresentaremos um algoritmo para resolução de problemas convexos não diferenciáveis, que foi apresentado inicialmente em 2005, na tese de doutorado de Freire [15], denominado Nonsmooth Feasible Direction Algorithm (NFDA).

O NFDA combina algumas ideias do método clássico de planos de corte de Kelley [21], os passos sério e nulo do método de feixe tradicional [22] com os sistemas internos do FDIPA, para criar um método que não utiliza qualquer tipo de subproblema de programação quadrática, funções de penalidade, barreira ou mesmo filtros.

Considere o problema

$$(PND) = \begin{cases} \text{minimizar} & F(x) \\ \text{sujeito a} & x \in \mathbb{R}^n \end{cases}$$

onde $F : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa e não necessariamente diferenciável.

Para evitar os problemas da não diferenciabilidade, o NFDA substitui o problema original (PND) pela minimização de uma função linear com uma restrição não diferenciável.

$$(PE) \begin{cases} \text{minimizar} & f(x, z) = z \\ \text{sujeito a} & F(x) \leq z \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R} \end{cases}$$

Em seguida, aproxima-se essa restrição por um conjunto de hiperplanos de apoio ao epígrafo da função F para obter uma direção de busca resolvendo dois sistemas de equações lineares análogos aos definidos no FDIPA. Constrói-se desta forma uma sequência de pontos interiores ao epígrafo de F cujos pontos de acumulação são soluções do problema original.

4.1 Método

O NFDA inicia com um ponto (x^1, z^1) no interior do epígrafo da função F e em seguida determina um hiperplano de apoio ao epígrafo de F no ponto $(x^1, F(x^1))$

$$H^1 := \{(x, z) \in \mathbb{R}^{n+1} \mid z = h^1(x)\}$$

onde $h^1(x) := F(x) + \langle s^1, x - x^1 \rangle$ com $s^1 \in \partial F(x^1)$.

Desse hiperplano define-se o seguinte problema auxiliar

$$(PA_1) \begin{cases} \text{minimizar} & f(x, z) = z \\ \text{sujeito a} & g^1(x, z) \leq 0 \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R} \end{cases}$$

onde $g^1 : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ é dada por $g^1(x, z) = g_1^1(x, z) = h^1(x) - z$.

Ao invés de resolver o problema (PA_1) que pode não ter solução, o algoritmo utilizará de maneira similar ao FDIPA os sistemas lineares

$$\begin{cases} B^1 d_1 + A^1 \lambda_1 = -\nabla f \\ \Lambda^1 (A^1)^t d_1 + G^1 \lambda_1 = 0 \end{cases} \quad \text{e} \quad \begin{cases} B^1 d_2 + A^1 \lambda_2 = 0 \\ \Lambda^1 (A^1)^t d_2 + G^1 \lambda_2 = -\Lambda^1 \omega^1 \end{cases}$$

sendo $A^1 = (Jg^1)^t = \nabla g_1^1$ para determinar uma direção $d^1 = (d_x^1, d_z^1) \in \mathbb{R}^{n+1}$.

Calcula-se então $d^1 = d_1 + \rho_1 d_2$ e $\bar{\lambda}^1 = \lambda_1 + \rho_1 \lambda_2$ onde ρ_1 deve satisfazer

$$0 < \rho_1 \leq (\xi - 1) \frac{d_1^t \nabla f}{d_2^t \nabla f}, \quad \xi \in (0, 1).$$

A partir do ponto (x^1, z^1) encontra-se um passo $t_1 > 0$ impondo-se a condição $g_1^1((x^1, z^1) + t_1 d^1) = 0$. Calculado t_1 , define-se $(y_1^1, \gamma_1^1) = (x^1, z^1) + t_1 d^1$ e também $(x_1^1, z_1^1) = (x^1, z^1) + \mu t_1 d^1$, sendo $0 < \mu < 1$ um parâmetro pré-definido.

Geometricamente, o ponto (y_1^1, γ_1^1) está na interseção da reta $(x, z) = (x^1, z^1) + t d^1$ com o hiperplano H^1 . Caso essa interseção seja vazia, o algoritmo toma $t = T$ onde $T > 0$ é também pré-estabelecido, de modo que $(y_1^1, \gamma_1^1) \in B((x^1, z^1), T \|d^1\|)$. Se $F(x_1^1) < z_1^1$ então $(x_1^1, z_1^1) \in (\text{epi}F)^\circ$ e define-se $x^2 = x_1^1$, $z^2 = z_1^1$ em um procedimento similar ao passo sério nos Métodos de Feixe. Caso contrário, se $F(x_1^1) \geq z_1^1$, isto é, se $(x_1^1, z_1^1) \notin (\text{epi}F)^\circ$ então continua-se no ponto (x^1, z^1) analogamente ao passo nulo nos Métodos de Feixe.

Em qualquer caso, determina-se o hiperplano de apoio ao epígrafo de F no ponto $(x^1, F(x^1))$ para então definir-se um novo problema auxiliar (PA_2) obtido de (PA_1) acrescentando a este, o hiperplano recém calculado, como nova restrição. Na iteração m , tem-se $(x^k, z^k) \in (\text{epi}F)^\circ$ e o problema auxiliar

$$(PA_m) \begin{cases} \text{minimizar} & f(x, z) = z \\ \text{sujeito a} & g^m(x, z) \leq 0 \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R} \end{cases}$$

onde $m = k + \sum_{j=1}^k i_j$ sendo i_j o número de sub-iterações necessárias para, dado o ponto $(x^j, z^j) \in (\text{epi}F)^\circ$ encontrar $(x^{j+1}, z^{j+1}) \in (\text{epi}F)^\circ$.

São admitidas no máximo M restrições, sendo M pré-estabelecido, de modo que, se $m > M$, $g^m = (g_1^m, g_2^m, \dots, g_M^m) : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^m$ fica determinada de acordo com a seguinte regra:

$$\begin{aligned} g_i^m(x, z) &= g_{j+1}^{m-1}(x, z), \quad 1 \leq j \leq M-1 \\ g_M^m(x, z) &= g_m^m(x, z) \end{aligned}$$

O problema (PA_m) foi obtido do problema (PA_{m-1}) da iteração anterior acrescentando-se a ele uma nova restrição linear $g_m^m(x, z) \leq 0$, sendo $g_m^m : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ construída a partir

do ponto (x_i^k, z_i^k) que pode estar ou não no interior do epígrafo de F e será definida pelo hiperplano de apoio ao epígrafo de F no ponto $(x_i^k, F(x_i^k))$

$$H^m := \{(x, z) \in \mathbb{R}^{n+1} \mid z = h^m(x)\}$$

onde $h^m(x) := F(x_i^k) + \langle s_i^k, x - x_i^k \rangle$, com $s_i^k \in \partial F(x_i^k)$.

Analogamente, define-se $g_m^m = h^m - z$. Da mesma forma, não se resolve (PA_m) . O algoritmo calcula $d^m = (d_x^m, d_z^m) \in \mathbb{R}^{n+1}$ através dos sistemas

$$\begin{cases} B^m d_1 + A^m \lambda_1 = -\nabla f \\ \Lambda^m (A^m)^t d_1 + G^m \lambda_1 = 0 \end{cases} \quad \text{e} \quad \begin{cases} B^m d_2 + A^m \lambda_2 = 0 \\ \Lambda^m (A^m)^t d_2 + G^m \lambda_2 = -\Lambda^m \omega^m \end{cases}$$

e encontra-se o maior passo $0 < t_m \leq T$ tal que $g_j^m((x^k, z^k) + t_m d^m) \leq 0$, $j = 1, 2, \dots, M$.

Ficam novamente definidos $(y_i^k, \gamma_i^k) = (x^k, z^k) + t_m d^m$ e $(x_i^k, z_i^k) = (x^k, z^k) + \mu t_m d^m$ com $(y_i^k, \gamma_i^k) \in B((x^k, z^k), T \|d^m\|)$ sendo i o contador interno na iteração k .

Se $F(x_i^k) < z_i^k$ então $(x_i^k, z_i^k) \in (\text{epi}F)^\circ$ e define-se $x^{k+1} = x_i^k$, $z^{k+1} = z_i^k$, como atualização de (x^k, z^k) a ser considerado na iteração $k + 1$. Neste caso, a direção $d^k = x^{k+1} - x^k$ será dita direção séria. Caso contrário, se $F(x_i^k) \geq z_i^k$, isto é, se $(x_i^k, z_i^k) \notin (\text{epi}F)^\circ$ então continua-se no ponto (x^k, z^k) .

Esse processo é repetido até que o algoritmo encontre uma direção d^k que satisfaça $\|d^k\| < \varepsilon$ onde $\varepsilon > 0$ é um parâmetro pré-estabelecido. Isto constitui um critério de parada uma vez que $d^k = 0$ na solução.

4.2 Algoritmo

Notação:

s = subgradiente de F no ponto x , isto é, $s \in \partial F(x)$.

A = matriz de subgradientes.

b = vetor de termos independentes.

g = vetor de restrições lineares.

$$A = \begin{pmatrix} s_1 & s_2 & \cdots & s_M \\ -1 & -1 & \cdots & -1 \end{pmatrix} = (\nabla g)^t.$$

$$b_i = \langle s^i, x^i \rangle - F(x^i).$$

$$g = A^t(x, z)^t - b.$$

Parâmetros: $\xi, \mu \in (0, 1)$, $\varphi > 0$, $M \in \mathbb{N}$ e $T > 0$.

Dados iniciais: $(x, z) \in (\text{epi}F)^\circ$, $0 < \lambda \in \mathbb{R}^m$, $0 < \omega \in \mathbb{R}^m$, $B \in \mathbb{R}^{n \times n}$ simétrica positiva.

Passo 1: Calcule $F(x)$, $s\partial F(x)$.

Passo 2: Atualize A , B , b , λ , ω e calcule g .

Passo 3:

1. Calcule d_1 e λ_1 resolvendo o sistema linear

$$\begin{cases} Bd_1 + A\lambda_1 = -\nabla f \\ \Lambda A^t d_1 + G\lambda_1 = 0 \end{cases}$$

se $d_1 = 0$ pare.

2. Calcule d_2 e λ_2 resolvendo o sistema linear

$$\begin{cases} Bd_2 + A\lambda_2 = 0 \\ \Lambda A^t d_2 + G\lambda_2 = -\Lambda\omega \end{cases}$$

3. Se $d_2 \nabla f > 0$ faça

$$\rho = \min \left\{ \varphi \|d_1\|^2, (\xi - 1) \frac{(d_1)^t \nabla f}{(d_2)^t \nabla f} \right\}$$

caso contrário, faça $\rho = \varphi \|d_1\|^2$.

4. Calcule $d = d_1 + \rho d_2$ e $\bar{\lambda} = \lambda_1 + \rho \lambda_2$.

Passo 4: Calcule $t = \max \{t_i \mid g_i((x, z) + t_i d) \leq 0, i = 1, 2, \dots, m\}$. Se $t \geq T$ então $t = T$.

Passo 5:

1. Calcule $(\bar{x}, \bar{z}) = (x, z) + \mu t d$.
2. Calcule $F(\bar{x})$, $s \in \partial F(\bar{x})$.
3. Se $F(\bar{x}) < \bar{z}$ faça $(x, z) = (\bar{x}, \bar{z})$.
4. Vá para o **Passo 2**.

5 NFDNA

Neste capítulo apresentaremos um algoritmo que é uma extensão do NFDA para problemas de otimização irrestrita com funções não necessariamente convexas ou diferenciáveis. Tal algoritmo foi proposto por Tanaka em [35] e é denominado Nonsmooth Feasible Direction Nonconvex Algorithm (NFDNA).

A teoria dos gradientes generalizados com Clarke [11] mostrou-se um bom instrumento para ir além e generalizar as ideias quando se está lidando com funções não diferenciáveis e não convexas, (veja [22, 26]).

Contudo, generalizar o método de planos de corte [10, 21] para o caso não convexo não é uma tarefa simples, entretanto algumas das ideias válidas no caso convexo também sejam úteis no caso não convexo. Por exemplo, em [16], a não convexidade é superada construindo-se aproximações poliedrais inferiores e superiores da função objetivo, e nos métodos de feixe, a forma mais comum de lidar com as dificuldades causadas pela não convexidade é usar o assim chamado subgradiente localmente mensurado ao invés do erro de linearização (veja [17, 20, 33]).

A não convexidade traz também algumas características adicionais para o problema, uma das quais é que a função objetivo pode ter vários mínimos e máximos locais. Como em todo método “não-global” de otimização, Tanaka provou em [35] a convergência do NFDNA para um ponto estacionário. Isto é, um ponto satisfazendo a condição necessária de otimalidade $0 \in \partial f(x)$, onde $\partial f(x)$ é o subdiferencial de f em x . Além disso, provou-se também que o NFDNA encontra um ponto estacionário x^* tal que $f(x^*) \leq f(x_1)$, onde x_1 é um ponto de partida determinado. Em outras palavras, o algoritmo é um método de descida.

No NFDNA, o emprego direto de um novo plano de corte pode, no caso não convexo, cortar o ponto de iteração atual e, portanto, algumas regras adicionais para a utilização dos planos de corte são necessárias.

5.1 Método - NFDNA

Considere o problema

$$(P) \begin{cases} \text{minimizar} & F(x) \\ \text{sujeito a} & x \in \mathbb{R}^n \end{cases}$$

onde $F : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função localmente Lipschitz contínua.

A não diferenciabilidade da função objetivo gera uma série de dificuldades bem conhecidas, logo substitui-se o problema original (P) por um problema equivalente (PE)

cuja função objetivo é linear, sujeito a uma restrição não diferenciável

$$(PE) \begin{cases} \text{minimizar} & f(x) = z \\ \text{sujeito a} & F(x) \leq z \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R} \end{cases}$$

A solução do problema equivalente (PE) é encontrada construindo-se uma sequência de problemas auxiliares lineares que vão sendo estabelecidos a medida que vai se aproximando o epígrafo da função f por planos de corte.

Desta forma tem-se o seguinte problema auxiliar

$$(PA_l) \begin{cases} \text{minimizar} & f(x, z) = z \\ \text{sujeito a} & g_1(x, z) \leq 0 \\ & g_2(x, z) \leq 0 \\ & \vdots \\ & g_l(x, z) \leq 0 \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R} \end{cases}$$

onde $g_1(x, z) = F(y_i) + \langle s_i, x - y_i \rangle$, $i = 1, 2, \dots, l$, y_i são pontos auxiliares, $s_i \in \partial F(y_i)$ são subgradientes arbitrários e l é o número de planos de corte corrente em uso.

Os problemas auxiliares (PA_l) não serão resolvidos e possivelmente eles não possuem soluções. Contudo serão utilizados para que o algoritmo, encontre uma direção de descida viável $d \in \mathbb{R}^{n+1}$. Essa direção é encontrada resolvendo-se dois sistemas lineares de maneira similar ao FDIPA e NFDA.

$$\begin{cases} Bd_1 + A\lambda_1 = -\nabla f \\ \Lambda(A)^t d_1 + G\lambda_1 = 0 \end{cases} \quad \text{e} \quad \begin{cases} Bd_2 + A\lambda_2 = 0 \\ \Lambda(A)^t d_2 + G\lambda_2 = -\Lambda\omega \end{cases}$$

Feito isso define-se $d = d_1 + \rho d_2$ e $\lambda = \lambda_1 + \rho \lambda_2$ onde

$$0 < \rho < (\xi - 1) \frac{d_1^t \nabla f}{d_2^t \nabla f}, \quad \xi \in (0, 1).$$

Depois de calcular uma direção de descida viável e um comprimento de passo, um novo ponto auxiliar (y_{l+1}, ω_{l+1}) viável com respeito a (PA_l) é calculado. Se o novo ponto (y_{l+1}, ω_{l+1}) é viável com respeito a (PE) e de descida para F , então atualiza-se a solução, ou seja, $(x^{k+1}, z^{k+1}) = (y_{l+1}, \omega_{l+1})$ passa a ser o novo ponto corrente e se diz que foi realizado um **passo sério de descida viável**.

Se o novo ponto (y_{l+1}, ω_{l+1}) é viável com respeito a (PE) mas não é de descida para F , considera-se que o ponto da iteração atual (x^k, z^k) está longe do epígrafo de F .

Neste caso, ao invés de usar a direção calculada pelos sistemas, usa-se a direção de máxima descida $-\nabla f = -(0, 0, \dots, 1)^t$ para obter um novo ponto estritamente viável mais próximo à fronteira do epígrafo de F . Assim tem-se $F(x^{k+1}) = F(x^k)$ no novo ponto (x^{k+1}, z^{k+1}) e a próxima direção de busca gerada pelos sistemas lineares será de descida para F . Este passo denomina-se **passo sério de máxima descida**.

Nestes dois casos de passo sério citados anteriormente, o algoritmo limpa todas as informações obtidas dos planos de corte armazenados até então.

Se nenhuma das alternativas anteriores acontecer, realiza-se um **passo nulo**, ou seja, um novo plano de corte é calculado no ponto (y_{l+1}, ω_{l+1}) , mas devido a não convexidade, pode acontecer que o novo plano de corte torne o ponto de iteração atual (x^k, z^k) inviável.

Se isto acontecer, recua-se ao longo da direção de busca e calcula-se um novo plano de corte. Esse processo de retorno continua até que se encontre um ponto viável com respeito ao plano de corte. Para fazer com que o método seja mais eficiente verifica-se a viabilidade do ponto $(x^k, \frac{F(x^k)+z^k}{2})$. Como a função objetivo é localmente Lipschitz contínua o plano de corte sempre existe.

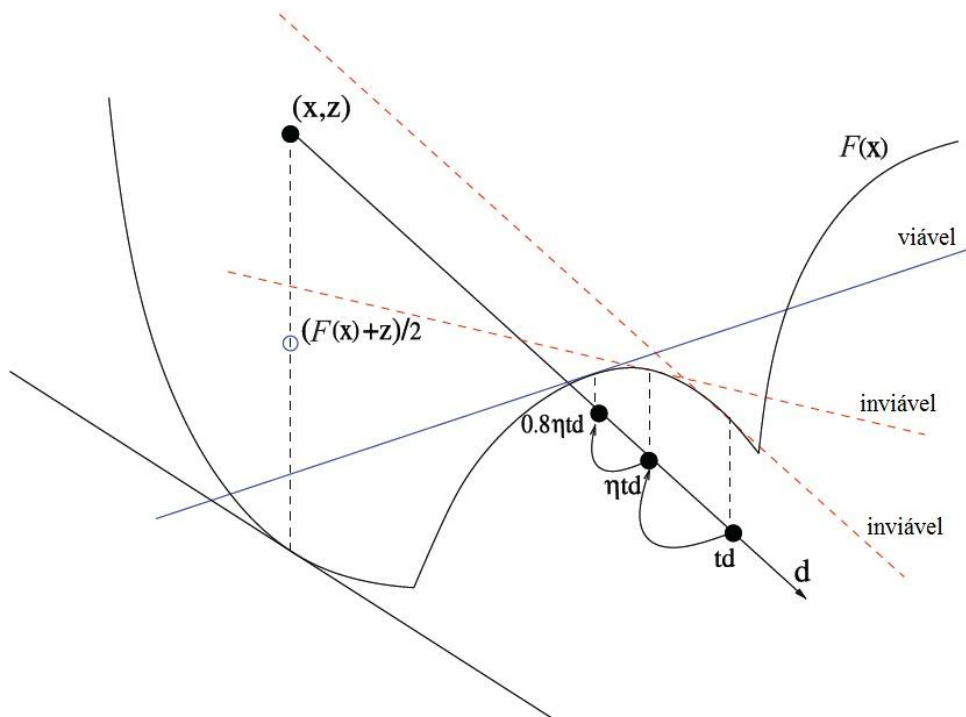


Figura 5.1 – Cálculo de um novo plano de corte viável.

5.2 Algoritmo - NFDNA

Parâmetros: Escolha uma tolerância aproximada final $\varepsilon > 0$. Selecione os parâmetros de controle $\varrho > 0$ e $\xi \in (1, 0)$ para o limite da deflexão. Selecione os multiplicadores $\mu \in (\frac{1}{2}, 1)$ para o tamanho de passo e o tamanho máximo de passo $t_{\max} > 0$.

Dados iniciais: Defina o contador de iterações $k = 1$ e o contador do número de planos de corte $l = 1$. Escolha um ponto inicial estritamente viável $(x^1, z^1) \in (\text{epi}F)^\circ$, um vetor $\lambda_1 \in \mathbb{R}^l$ positivo inicial e uma matriz simétrica definida positiva $B \in \mathbb{R}^{(n+1) \times (n+1)}$. Seja $y_1^1 = x^1$. Calcule $F(x^1)$.

Passo 1:(Plano de corte para passos sérios.) Calcule $s_l^k \in \partial F(x^k)$ e o primeiro plano de corte

$$g_l^k(x^k, z^k) = F(x^k) - z^k \in \mathbb{R}.$$

Faça

$$\nabla g_l^k(x^k, z^k) = (s_l^k, -1) \in \mathbb{R}^{n+1}.$$

Defina

$$\bar{g}_l^k(x^k, z^k) = [g_l^k(x^k, z^k)] \in \mathbb{R} \quad \text{e} \quad \nabla \bar{g}_l^k(x^k, z^k) = [\nabla g_l^k(x^k, z^k)].$$

Passo 2:(Encontrando a Direção) Calcule $d_l^k = (d_x^k, d_z^k) \in \mathbb{R}^{n+1}$, uma direção de descida viável para (PA_l) :

1. (Direção de descida) Determine $d_1 \in \mathbb{R}^{n+1}$ e $\lambda_1 \in \mathbb{R}^l$ resolvendo o sistema de equações lineares

$$B^k d_1 + A^k \lambda_1 = -\nabla f \tag{5.1}$$

$$\Lambda^k (A^k)^t d_1 + G^k \lambda_1 = 0 \tag{5.2}$$

2. (Direção viável) Determine $d_2 \in \mathbb{R}^{n+1}$ e $\lambda_2 \in \mathbb{R}^l$ resolvendo o sistema de equações lineares

$$B^k d_2 + A^k \lambda_2 = 0 \tag{5.3}$$

$$\Lambda^k (A^k)^t d_2 + G^k \lambda_2 = -\lambda_l^k \tag{5.4}$$

se $d_2 \nabla f > 0$ então

$$\rho = \min \left\{ \varrho \|d_1\|^2, (\xi - 1) \frac{d_1 \nabla f}{d_2 \nabla f} \right\} \tag{5.5}$$

caso contrário

$$\rho = \varrho \|d_1\|^2 \tag{5.6}$$

3. (Direção de descida viável)

$$d_l^k = d_1 + \rho d_2 \quad (5.7)$$

$$\lambda_k = \lambda_1 + \rho \lambda_2 \quad (5.8)$$

Passo 3:(Tamanho de passo e atualização da solução) Calcule o tamanho do passo

$$t^k = \min \left\{ t_{\max}, \max \left\{ t \mid \bar{g}_l^k((x^k, z^k) + t d_l^k) \leq 0 \right\} \right\}$$

se $\|d_l^k\| \leq \varepsilon$ e $t^k < t_{\max}$ pare com (x^k, z^k) sendo a solução. Caso contrário, defina

$$(y_{l+1}^k, \omega_{l+1}^k) = (x^k, z^k) + \mu t^k d_l^k$$

e calcule o valor correspondente $F(y_{l+1}^k)$.

Se $\omega_{l+1}^k \leq F(y_{l+1}^k)$, o passo é nulo, então vá para o passo 6. Caso contrário, se $F(x^k) \geq F(y_{l+1}^k)$, vá para o passo 4. Se não vá para o passo 5.

Passo 4:(Passo sério de descida viável) O ponto $(y_{l+1}^k, \omega_{l+1}^k)$ é considerado o novo ponto corrente, ou seja, $(x^{k+1}, z^{k+1}) = (y_{l+1}^k, \omega_{l+1}^k)$. Atualiza-se B^k para B^{k+1} e λ^k para λ^{k+1} . Faça $k = k + 1$, $l = 1$ e vá para o passo 1.

Passo 5:(Passo sério de máxima descida) Descarte o ponto $(y_{l+1}^k, \omega_{l+1}^k)$ calculado no passo 3 e calcule o novo ponto corrente da seguinte forma

$$(x^{k+1}, z^{k+1}) = (x^k, z^k) - \mu g_l^k(x^k, z^k) \nabla f$$

e $F(x^{k+1}) = F(x^k)$. Atualiza-se B^k para B^{k+1} e λ^k para λ^{k+1} . Faça $k = k + 1$, $l = 1$ e vá para o passo 1.

Passo 6:(Passo nulo)

1. (Erro de linearização) Calcule $s_{l+1}^k \in \partial F(y_{l+1}^k)$ e um erro de linearização

$$\alpha = F(x^k) - F(y_{l+1}^k) - (s_{l+1}^k)^t (x^k - y_{l+1}^k).$$

2. (Procedimento de retorno) Se $\alpha < g_l^k(x^k, z^k)/2$ retorna-se ao longo do vetor d^k até que um ponto viável seja encontrado, ou seja, descarta-se o ponto $(y_{l+1}^k, \omega_{l+1}^k)$ calculado no passo 3 e recalcula-se da seguinte forma

$$(y_{l+1}^k, \omega_{l+1}^k) = (x^k, z^k) + \eta \mu t^k d^k$$

com $\eta = 0.8\eta$ e vá para o passo 6(1).

3. (Planos de cortes para passos nulos) Calcule o novo plano de corte e seus gradientes

$$g_{l+1}^k(x^k, z^k) = -\alpha + F(x^k) - z^k \quad \text{e} \quad \nabla g_{l+1}^k(x^k, z^k) = (s_{l+1}^k, -1).$$

Defina

$$\bar{g}_{l+1}^k(x^k, z^k) = [g_1^k(x^k, z^k), \dots, g_l^k(x^k, z^k), g_{l+1}^k(x^k, z^k)] \in \mathbb{R}^{l+1}$$

e

$$\nabla \bar{g}_{l+1}^k(x^k, z^k) = [\nabla g_1^k(x^k, z^k), \dots, \nabla g_l^k(x^k, z^k), \nabla g_{l+1}^k(x^k, z^k)] \in \mathbb{R}^{(n+1) \times (n+1)}.$$

Faça $l = l + 1$ e vá para o passo 2.

5.3 Análise de Convergência

Nesta seção, apresentaremos a análise de convergência feita por Tanaka em [35], na qual mostra-se que a direção de busca d^k definida em (5.7) é uma direção de descida para F , isto é, para (PE) e (PA_l) . Então, mostra-se que, quando o ponto de iteração corrente está próximo o suficiente da fronteira do epígrafo de F , d^k também será uma direção de descida para F , ou seja, para (P) .

Em seguida, mostra-se que o número de passos nulos em cada iteração é finito, e que a sequência $(x^k, z^k)_{k \in \mathbb{N}}$ é limitada. Finalmente, tem-se que todo ponto de acumulação (x^*, z^*) da sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$ é estacionário para F .

Assume-se que a função objetivo F é localmente Lipschitz contínua, e as seguintes hipóteses são feitas:

Hipótese 5.1. *Existem números positivos ω_1 e ω_2 tais que $\omega_1 \|d\|^2 \leq d^t B d \leq \omega_2 \|d\|^2$ para todo $d \in \mathbb{R}^{n+1}$.*

Hipótese 5.2. *Existem números positivos λ^I , λ^S e g_{\max} tais que $0 < \lambda_i \leq \lambda^S$, $i = 1, \dots, l$ e $\lambda_i \geq \lambda^I$ para todo i tal que $\bar{g}_i(x, z) \geq g_{\max}$.*

Hipótese 5.3. *O conjunto $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^1)\}$ é compacto.*

Hipótese 5.4. *Para todo $(x, z) \in (\text{epi} F)^\circ$ e todo i tal que $g_i(x, z) = 0$ os vetores $\nabla g_i(x, z)$ são linearmente independentes.*

Inicia-se a análise teórica do NFDNA observando-se que as soluções d_1 , λ_1 , d_2 e λ_2 dos sistemas (5.1)-(5.2) e (5.3)-(5.4) são únicas como enuncia o próximo Lema.

Lema 5.5. *Dado um ponto $(x, z) \in (\text{epi} F)^\circ$, qualquer matriz $B \in \mathbb{R}^{(n+1) \times (n+1)}$ definida positiva e qualquer vetor não negativo $\lambda \in \mathbb{R}^l$ tal que $\lambda_i > 0$, se $g_i(x, z) = 0$, a matriz*

$$M = \begin{bmatrix} B & \nabla \bar{g}_l(x, z) \\ \Lambda \nabla \bar{g}_l(x, z)^t & G_l(x, z) \end{bmatrix}$$

é não singular.

Segue do Lema 5.5 que d_1 , λ_1 , d_2 e λ_2 são limitados superiormente.

Lema 5.6. *A direção d_1 definida por (5.1)-(5.2) satisfaz $d_1^t \nabla f(x, z) \leq -d_1^t B d_1$*

Como consequência do Lema 5.6, segue que a direção d_1 é uma direção de descida para f , isto é, para (PE) e (PA_l) .

Proposição 5.7. *A direção d definida em (5.7) é uma direção de descida para (PE) e (PA_l) .*

Demonstração. Como consequência de (5.7),

$$d^t \nabla f(x, z) = d_1^t \nabla f(x, z) + \rho d_2^t \nabla f(x, z).$$

Como

$$\rho \leq (\xi - 1) \frac{d_1^t \nabla f(x, z)}{d_2^t \nabla f(x, z)}, \quad \text{com } \xi \in (0, 1)$$

se $d_2^t \nabla f(x, z) > 0$ e, como d_1 é uma direção de descida para f pelo Lema 5.6 obtém-se

$$\begin{aligned} d^t \nabla f(x, z) &\leq d_1^t \nabla f(x, z) + (\xi - 1) d_1^t \nabla f(x, z) \\ &= \xi d_1^t \nabla f(x, z) \\ &\leq 0 \end{aligned}$$

nota-se que $d^t \nabla f(x, z) = 0$ somente se $d_1 = 0$.

Por outro lado, se $d_2^t \nabla f(x, z) \leq 0$ obtém-se a inequação

$$d^t \nabla f(x, z) \leq d_1^t \nabla f(x, z) < 0.$$

Portanto, d é uma direção de descida para f . □

Lema 5.8. *Seja $(x^k, z^k) \in (\text{epi}F)^\circ$ um ponto corrente gerado pelo algoritmo. Para todo $k \geq 1$, segue que $F(x^{k+1}) \leq F(x^k)$ e $z^{k+1} < z^k$. Além disso, o próximo ponto $(x^{k+1}, z^{k+1}) \in (\text{epi}F)^\circ$.*

Demonstração. O ponto corrente (x^k, z^k) é atualizado no Passo 4 ou no Passo 5 do algoritmo. No Passo 4, tem-se que $(x^{k+1}, z^{k+1}) = (y_{l+1}^k, \omega_{l+1}^k)$ somente se $\omega_{l+1}^k > F(y_{l+1}^k)$ e $F(x^k) \geq F(y_{l+1}^k)$. Portanto, tem-se $z^{k+1} > F(x^{k+1})$, ou seja, $(x^{k+1}, z^{k+1}) \in (\text{epi}F)^\circ$ e $F(x^{k+1}) \leq F(x^k)$. Além disso, $(d^k)^t \nabla f < 0$ pela Proposição 5.7. Portanto, $d_z^k < 0$ e o próximo componente z^{k+1} é calculado pela fórmula $z^{k+1} = z^k + \mu t^k d_z^k$, onde $\mu, t^k > 0$. Portanto $z^{k+1} < z^k$.

Por outro lado, no Passo 5, usa-se a direção de descida máxima $-\nabla f$ como direção de busca e portanto, $x^{k+1} = x^k$ e, naturalmente, $F(x^{k+1}) = F(x^k)$. Tem-se ainda $z^{k+1} = z^k - \mu(z^k - F(x^k))$, onde $\mu \in (0, 1)$ e $z^k - F(x^k) > 0$ pois $(x^k, z^k) \in (\text{epi}F)^\circ$. Portanto, tem-se novamente que $z^{k+1} < z^k$ e $z^{k+1} > F(x^k) = F(x^{k+1})$. □

Lema 5.9. *Seja $(x^k, z^k) \in (\text{epi}F)^\circ$ um ponto suficientemente próximo da fronteira do epígrafo de F , isto é, $z^k - F(x^k) < -\mu t^k d_z^k$. Se (x^k, z^k) não é um ponto estacionário, então a direção d^k definida por (5.7) é uma direção de descida para o problema (P) , isto é, para F .*

Demonstração. Como $(x^k, z^k) \in (\text{epi}F)^\circ$, tem-se que $z^k = F(x^k) + \varepsilon_1$ para algum $\varepsilon_1 \geq 0$. Também tem-se que $(d^k)^t \nabla f < 0$ pela Proposição 5.7, e portanto, $d_z^k < 0$. O próximo iterado é calculado pela fórmula $(x^{k+1}, z^{k+1}) = (x^k, z^k) + \mu t^k (d_x^k, d_z^k)$ com $\mu, t^k > 0$. Portanto, $z^{k+1} = z^k - \varepsilon_2 = F(x^k) + \varepsilon_1 - \varepsilon_2$, onde denota-se $\varepsilon_2 = -\mu t^k d_z^k > 0$. Quando ε_1 é suficientemente pequeno, ou seja, $\varepsilon_1 < \varepsilon_2$, tem-se obviamente que $z^{k+1} - F(x^k) < 0$. Também segue que $F(x^{k+1}) \leq z^{k+1}$, pois $(x^{k+1}, z^{k+1}) \in (\text{epi}F)^\circ$ pelo Lema 5.8. Agora combinando as duas coisas obtém-se $F(x^{k+1}) \leq z^{k+1} < F(x^k)$ e d^k é uma direção de descida para F por definição. \square

Corolário 5.10. *A sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$ gerada pelo algoritmo é limitada.*

Demonstração. Como $z^{k+1} < z^k$ para todo $k \in \mathbb{N}$ e pela Hipótese 5.4 a sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$ está limitada no conjunto $(\text{epi}F) \cap \{(x, z) \in \mathbb{R}^{n+1} \mid z < z^1\}$. \square

Lema 5.11. *A direção d definida em (5.7) é limitada.*

Demonstração. A escolha de ρ (veja (5.5)-(5.6)) assegura que se tenha

$$\rho \leq \varrho \|d_1\|^2 \text{ para algum } \varrho > 0. \quad (5.9)$$

Por outro lado, do Lema 5.6 e da Hipótese 5.2, obtém-se que $-d_1^t \nabla f \geq \omega_1 \|d_1\|^2$ e portanto, devido a (5.5), tem-se

$$\rho \geq \min \left\{ \varrho, \frac{(1 - \xi)\omega_1}{d_2^t \nabla f} \right\} \|d_1\|^2$$

se $d_2^t \nabla f > 0$.

Devido a (5.6) e como d_2 é limitada, existe um limite inferior $\varrho_{low} > 0$ tal que $\rho \geq \varrho_{low} \|d_1\|^2$.

Portanto, pela limitação de d_1 , a deflexão dada por ρ é positiva limitada como acima. De (5.7), (5.8) e (5.9), tem-se

$$\begin{aligned} \|d\| &= \|d_1 + \rho d_2\| \\ &\leq \|d_1\| + \|\rho d_2\| \\ &\leq \|d_1\| + \varrho \|d_1\|^2 \|d_2\| \\ &= (1 + \varrho \|d_1\| \cdot \|d_2\|) \|d_1\|. \end{aligned}$$

Portanto, existe $\delta > 1$ tal que $\|d\| \leq \delta \|d_1\|$ é válida. E devido a limitação de d_1 , tem-se que d é limitada como acima. \square

No próximo lema, mostra-se que no passo 6 do algoritmo, um ponto (y_{l+1}, ω_{l+1}) é encontrado depois de um número finito de loops dentro do passo, de modo que, o ponto corrente $(x, z) \in (\text{epi}F)^\circ$ seja viável com respeito ao plano de corte calculado no ponto y_{l+1} .

Lema 5.12. *Existe $(y_{l+1}, \omega_{l+1}) \in \mathbb{R}^{n+1}$ tal que $F(x) - z \leq \alpha$, onde*

$$\alpha = F(x) - F(y_{l+1}) - s_{l+1}^t(x - y_{l+1})$$

e $s_{l+1} \in \partial F(y_{l+1})$. *Este ponto é encontrado após um número finito de loops.*

Demonstração. Suponha por contradição que y_{l+1}, ω_{l+1} viável não pode ser encontrado. Um ponto $(y_{l+1,i}, \omega_{l+1,i})$ é calculado pela fórmula

$$(y_{l+1,i}, \omega_{l+1,i}) = (x, z) + \eta_i \mu t d$$

onde $\mu, t > 0$, $\eta_1 = 1$, $\eta_2 \in (\frac{1}{2}, 1)$ e $\eta_{i+1} = 0.8\eta_i$ com $(i = 2, 3, \dots)$. Como $(y_{l+1,i}, \omega_{l+1,i})$ viável não é encontrado segue que

$$F(x) - z > F(x) - F(y_{l+1,i}) - s_{l+1}^t(x - y_{l+1,i}) = \alpha_i, \quad \text{para todo } i \in \mathbb{N}.$$

Como $\eta_{i+1} < \eta_i$ para todo i , tem-se $\eta_i \rightarrow 0$. Isto implica que $y_{l+1,i} \rightarrow x$. Como F é localmente Lipschitz contínua, segue que $|F(y_{l+1,i}) - F(x)| \rightarrow 0$ e portanto $\alpha \rightarrow 0$. Mas $F(x) - z < 0$ pois $(x, z) \in (\text{epi}F)^\circ$. O que é uma contradição. \square

Lema 5.13. *Existe $\tau > 0$ tal que para todo $(x, z) \in (\text{epi}F)^\circ$ e para todo $d \in \mathbb{R}^{n+1}$ gerado pelo algoritmo, tem-se $\bar{g}_l((x, z) + td) \leq 0$ para todo $t \in [0, \tau]$.*

Demonstração. Denota-se por b um vetor tal que $b_i = s_i^t y_i - F(y_i)$ para todo $i = 1, 2, \dots, l$. Agora $\bar{g}_l(x, z) = (\nabla \bar{g}_l(x, z))^t(x, z) - b$, pois

$$\begin{aligned} g_i(x, z) &= F(y_i) + s_i^t(x - y_i) - z \\ &= F(y_i) + s_i^t x - s_i^t y_i - z \\ &= (s_i^t, -1)^t(x, z) - s_i^t y_i + F(y_i) \\ &= (\nabla g_i(x, z))^t(x, z) - b_i \end{aligned}$$

para todo $i = 1, 2, \dots, l$.

Por construção, tem-se que

$$t \leq \max \{t_i \mid g_i((x, z) + t_i d) \leq 0, \quad i = 1, 2, \dots, l\}.$$

Agora combinando estas duas observações, e observando-se que $\nabla g_i(x, z)$ não depende do ponto (x, z) mas dos pontos auxiliares y_i , $i = 1, 2, \dots, l$, obtém-se

$$\begin{aligned} g_i((x, z) + t_i d) &= (\nabla g_i((x, z) + t_i d))^t((x, z) + t_i d) - b_i \\ &= (\nabla g_i(x, z))^t((x, z) + t_i d) - b_i \\ &= (\nabla g_i(x, z))^t(x, z) - b_i + t_i (\nabla g_i(x, z))^t d \\ &= g_i(x, z) + t_i (\nabla g_i(x, z))^t d \leq 0 \end{aligned} \quad (5.10)$$

para todo $i = 1, 2, \dots, l$.

Se $(\nabla g_i(x, z))^t \leq 0$, a equação acima é satisfeita com qualquer $t_i > 0$. Considere o caso quando $(\nabla g_i(x, z))^t > 0$. Por (5.7) e (5.8), segue que $(\nabla g_i(x, z))^t d = (\nabla g_i(x, z))^t(d_1 + \rho d_2)$, e de (5.2) e (5.4) obtém-se

$$\nabla g_i(x, z)^t d_1 = -g_i(x, z) \frac{\lambda_{1,i}}{\lambda_i} \quad (5.11)$$

e

$$\nabla g_i(x, z)^t d_2 = -1 - g_i(x, z) \frac{\lambda_{2,i}}{\lambda_i}. \quad (5.12)$$

Agora combinando (5.10), (5.11), e (5.12) obtém-se

$$g_i(x, z) - t_i g_i(x, z) \frac{\lambda_{1,i} + \rho \lambda_{2,i}}{\lambda_i} - \rho t_i = g_i(x, z) \left(1 - t_i \frac{\bar{\lambda}_i}{\lambda_i}\right) - \rho t_i \leq 0,$$

onde se tinha denotado $\bar{\lambda}_i = \lambda_{1,i} + \rho \lambda_{2,i}$. Obviamente $\rho t_i > 0$ e $g_i(x, z) < 0$. Portanto, a inequação é satisfeita se

$$t_i \frac{\bar{\lambda}_i}{\lambda_i} \leq 1.$$

Agora, λ é limitado pela Hipótese 5.3, e λ_1 , λ_2 e ρ são limitados (veja a prova do Lema 5.6). Portanto, $\bar{\lambda}$ é limitado e existe $\tau > 0$ tal que $(\lambda_i/\bar{\lambda}_i) > \tau$ para todo $i = 1, 2, \dots, l$. Portanto, para todo $t \in [0, \tau]$, segue que $g_i((x, z) + td) \leq 0$. \square

Lema 5.14. *Seja X um conjunto convexo e sejam $x^0 \in X^\circ$ e $\bar{x} \in X$. Seja $\{\bar{x}^k\} \subset \mathbb{R}^n \setminus X$ uma sequência tal que $\bar{x}^k \rightarrow \bar{x}$ e seja x^k definida por $x^k = x^0 + \mu(\bar{x}^k - x^0)$ para algum $\mu \in (0, 1)$. Então existe $k_0 \in \mathbb{N}$ tal que $x^k \in X^\circ$ para todo $k \geq k_0$.*

Demonstração. Observa-se que $x^k = x^0 + \mu(\bar{x}^k - x^0) \rightarrow x^0 + \mu(\bar{x} - x^0) = x_\mu$ quando $k \rightarrow \infty$. Como o segmento $[x^0, \bar{x}] \subset X$ e $\mu \in (0, 1)$ tem-se que $x_\mu \in X^\circ$ e, como consequência existe $\delta > 0$ tal que $B(x_\mu, \delta) \subset X^\circ$. Como $x^k \rightarrow x_\mu$ existe $k_0 \in \mathbb{N}$ tal que $x^k \in B(x_\mu, \delta) \subset X^\circ$, para todo $k > k_0$. \square

Lema 5.15. *Seja $(\bar{y}, \bar{\omega})$ um ponto de acumulação da sequência $\{(y_i, \omega_i)\}_{i \in \mathbb{N}}$ gerada pelo algoritmo. Então $\bar{\omega} = F(\bar{y})$.*

Demonstração. Um novo ponto auxiliar (y_i, ω_i) é calculado no passo 3 do algoritmo. Se $\omega_i > F(y_i)$ realiza-se um passo sério, ou seja, vai para o passo 4 ou 5. Portanto, no ponto de acumulação tem-se $\bar{\omega} \leq F(\bar{y})$. Suponha agora que $\bar{\omega} < F(\bar{y})$. Considere o plano de corte $\bar{F}_{s_i}(x) = F(\bar{y}) + s_i(x - \bar{y})$ para algum $s_i \in \partial F(\bar{y})$. Seja \bar{F}_{s_i} uma nova restrição para (PA_l) , isto é, $g_i(x, z) = \bar{F}_{s_i}(x) - z$.

Denota-se por $r = D((\bar{y}, \bar{\omega}), \bar{F}_{s_i})$ a distância entre o ponto $(\bar{y}, \bar{\omega})$ e o plano \bar{F}_{s_i} . Como $\bar{\omega} < F(\bar{y})$ tem-se que $r > 0$. Seja $\bar{B} = B((\bar{y}, \bar{\omega}), r/2)$. Obviamente, $\bar{B} \cap \bar{F}_{s_i} = \emptyset$.

Agora, $(y_i, \omega_i) \in \text{epi} \bar{F}_{s_i}$ para qualquer i e $B \subset (\text{epi} \bar{F}_{s_i})^c$. Portanto $(y_i, \omega_i) \notin \bar{B}$. O que é uma contradição. \square

Lema 5.16. *Seja $(x^k, z^k) \in (\text{epi} F)^\circ$. O próximo iterado $(x^{k+1}, z^{k+1}) \in (\text{epi} F)^\circ$ é encontrado após um número finito de iterações.*

Demonstração. O novo ponto (x^{k+1}, z^{k+1}) está no interior do $\text{epi} F$ pelo Lema 5.8. Portanto, necessita-se mostrar somente que o novo ponto é encontrado depois de um número finito de iterações.

Um novo ponto auxiliar (y_i, ω_i) é encontrado após um número finito de loops dentro do passo 6, pelo Lema 5.12. Se $\omega_i > F(y_i)$ tem-se um passo sério (passo 4 ou 5) e, obviamente $(x^{k+1}, z^{k+1}) \in (\text{epi} F)^\circ$. A sequência $\{(y_i, \omega_i)\}_{i \in \mathbb{N}}$ é limitada por construção logo existe um ponto de acumulação $(\bar{y}, \bar{\omega})$. Pelo Lema 5.15 este ponto de acumulação está na fronteira do epígrafo de f .

Denota-se por $\bar{F}_{s_i}(x) = F(y_i) + s_i^t(x - y_i)$ o plano de corte correspondente a i -ésima restrição e por $\hat{F}(x) = \max \{ \bar{F}_{s_i}(x) \mid i = 1, \dots, l \}$ a função linear por partes definida pelo máximo de todos os planos de corte no ponto de acumulação $(\bar{y}, \bar{\omega})$. Pelo Lema 5.14 existe $i_o \in \mathbb{N}$ tal que $(y_{i_o}, \omega_{i_o}) \in (\text{epi} \hat{F})^\circ$. Mostrar-se-a que $(y_{i_o}, \omega_{i_o}) \in (\text{epi} F)^\circ$ ainda que o $\text{epi} F$ não seja convexo.

Por absurdo, supõe-se que $(y_{i_o}, \omega_{i_o}) \notin (\text{epi} F)^\circ$, ou seja, que $F(y_{i_o}) \geq \omega_{i_o}$. Um passo nulo ocorre e adiciona-se um novo plano de corte. Agora (y_{i_o}, ω_{i_o}) está num segmento de reta que liga o ponto de acumulação $(\bar{y}, \bar{\omega})$ e o ponto de iteração corrente (x^k, z^k) , abaixo do epígrafo de F . Portanto, o novo plano de corte torna o ponto $(\bar{y}, \bar{\omega})$ inviável. Mas então $(\bar{y}, \bar{\omega})$ não pode ser um ponto de acumulação, o que é uma contradição.

Portanto, tem-se que $(y_{i_o}, \omega_{i_o}) \in (\text{epi} F)^\circ$ e ainda define-se $(x^{k+1}, z^{k+1}) = (y_{i_k}, \omega_{i_k})$ (no passo 4 do algoritmo) ou um passo sério de máxima descida ocorre (passo 5 do algoritmo). \square

Lema 5.17. *Seja d_1^* um ponto de acumulação da sequência $\{d_1^k\}_{k \in \mathbb{N}}$. Então $d_1^* = 0$.*

Demonstração. Por construção tem-se que

$$(x^{k+1}, z^{k+1}) = (x^k, z^k) + \mu t^k d^k \quad \text{ou} \quad (x^{k+1}, z^{k+1}) = (x^k, z^k) - \mu(z^k - F(x^k))\nabla f.$$

A sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$ é limitada pelo Corolário 5.10. Denotando $x^* = \lim_{k \rightarrow \infty} x^k$ e $z^* = \lim_{k \rightarrow \infty} z^k$ e seja $K \subset \mathbb{N}$ tal que $\{t^k\}_{k \in \mathbb{N}} \rightarrow t^*$. Isto segue do Lema 5.13 onde tem-se $t^* > 0$.

Quando $k \rightarrow \infty$, $k \in K$ segue que

$$z^* = z^* + \mu t^* d^* \quad \text{ou} \quad z^* = (1 - \mu)z^* + \mu F(x^*),$$

ou seja, tem-se que $d_z^* = 0$ ou $z^* = F(x^*)$. Contudo, devido ao Lema 5.9, o último não acontece (d^* é uma direção de descida para F ou $d^* = 0$). Portanto $d_z^* = 0$.

Pela Proposição 5.7, segue que

$$0 = d_z^* = (d^*)^t \nabla f = \xi (d_1^*)^t \nabla f = \xi d_{1,z}^* \leq 0$$

com algum $\xi > 0$ e assim $d_{1,z}^* = 0$. Além disso, pelo Lema 5.6 tem-se

$$0 = d_{1,z}^* = (d_1^*)^t \nabla f \leq -(d_1^*)^t B d_1^* \leq 0$$

e como B é definida positiva, pode-se concluir que $d_1^* = 0$. □

Lema 5.18. *Seja $(s_i^t, -1)^t$ o gradiente de uma restrição ativa no ponto de acumulação (x^*, z^*) da sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$. Então $s_i \in \partial F(x^*)$.*

Demonstração. Como no ponto de acumulação (x^*, z^*) tem-se que $f(x^*) = z^*$ a primeira restrição $g_1(x^*, z^*)$ é ativa e $s_1 \in \partial F(x^*)$ por construção (veja passo 1 do algoritmo).

Suponha agora que a restrição $g_i(x^*, z^*)$, $i > 1$, é ativa. Denota-se por

$$\bar{F}_{s_i} = F(y_i) + s_i^t(x - y_i)$$

o plano de corte correspondente na restrição ativa. Ou seja, $\bar{F}_{s_i}(x^*) = z^*$. Na vizinhança do ponto de acumulação $x \in B(x^*, \sigma)$, para algum $\sigma > 0$, tem-se que $\bar{F}_{s_i}(x)$ é uma aproximação inferior da função objetivo $F(x)$ ou $s_1 = 0$ (no caso do algoritmo já ter parado). Portanto tem-se que para todo $x \in B(x^*, \sigma)$, $\sigma > 0$ e $s_i \in \partial F(y_i)$

$$\begin{aligned} F(x) &\geq F(y_i) + s_i^t(x - y_i) \\ &= F(y_i) - F(x^*) + F(x^*) + s_i^t(x - y_i) - s_i^t(x - x^*) + s_i^t(x - x^*) \\ &= F(x^*) + s_i^t(x - x^*) + F(y_i) + s_i^t(x^* - y_i) - F(x^*) \\ &= F(x^*) + s_i^t(x - x^*) + g_i(x^*, z^*) \\ &= F(x^*) + s_i^t(x - x^*), \end{aligned}$$

pois $F(x^*) = z^*$ e $g_i(x^*, z^*) = 0$.

Agora, se for utilizado $x = x^* + tv$, onde $v \in \mathbb{R}^n$, $t > 0$ pode-se escrever

$$F(x) - F(x^*) \geq s_i^t(x - x^*) = t s_i^t v$$

para todo $x \in B(x^*, \sigma)$ e obtém-se

$$\begin{aligned} F^\circ(x^*, v) &= \limsup_{\substack{x' \rightarrow x^* \\ t \downarrow 0}} \frac{F(x' + tv) - F(x')}{t} \\ &\geq \limsup_{t \downarrow 0} \frac{F(x^* + tv) - F(x^*)}{t} \\ &\geq \limsup_{t \downarrow 0} \frac{t s_i^t v}{t} = s_i^t v. \end{aligned}$$

Portanto, pela definição de subdiferencial $s_i \in \partial F(x^*)$. □

Lema 5.19. *Para k suficientemente grande, tem-se que $\lambda_1^k \geq 0$.*

Demonstração. Considera-se o seguinte problema de otimização convexa

$$\begin{cases} \text{minimizar} & \Phi(x, z) \\ \text{sujeito a} & \bar{g}_l(x, z) \leq 0, \end{cases}$$

onde $\Phi(x, z) = z + d_1^t Bx$. Um ponto KKT (x^Φ, z^Φ) do problema satisfaz

$$\nabla z + B d_1 + \nabla \bar{g}_l(x^\Phi, z^\Phi) \lambda^\Phi = 0 \quad (5.13)$$

$$G_l(x^\Phi, z^\Phi) \lambda^\Phi = 0 \quad (5.14)$$

$$\lambda^\Phi \geq 0 \quad (5.15)$$

$$\bar{g}_l(x, z) \leq 0. \quad (5.16)$$

O sistema (5.1)-(5.2) no passo 2 do algoritmo pode ser reescrito como

$$\begin{aligned} \nabla z + B d_1^k + \nabla \bar{g}_l^k(x^k, z^k) \lambda_1^k &= 0 \\ G_l^k(x^k, z^k) \lambda_1^k &= \varphi^k, \end{aligned}$$

onde $\varphi^k = -\Lambda_1^k [\nabla \bar{g}_l^k(x^k, z^k)]^t d_1^k$.

Quando $d_1^k \rightarrow 0$ tem-se que $\varphi^k \rightarrow 0$ e então, para dado $\varepsilon_1 > 0$, existe $K_1 > 0$ tal que

$$\|\lambda_1^k - \lambda^\Phi\| < \varepsilon_1, \text{ para } k > K_1.$$

Então como $\lambda^\Phi \geq 0$ por (5.15) deduz-se que $\lambda_1^k \geq 0$ para k suficientemente grande. □

Teorema 5.20. *Para qualquer ponto de acumulação (x^*, z^*) da sequência $\{(x^k, z^k)\}_{k \in \mathbb{N}}$ tem-se que $0 \in \partial F(x^*)$.*

Demonstração. Considere as equações (5.1)-(5.2). Quando $k \rightarrow \infty$ tem-se que $d_1^* = 0$ pelo Lema 5.17. Portanto, obtém-se

$$\nabla \bar{g}_l^*(x^*, z^*) \lambda_1^* = -\nabla f \quad \text{e} \quad \bar{g}_l^*(x^*, z^*) \lambda_1^* = 0,$$

onde denota-se por λ_1^* o vetor de multiplicação de Lagrange correspondente a d_1^* e por $\bar{g}_l^*(x^*, z^*)$ a restrição correspondente.

Como

$$\nabla \bar{g}_l^*(x^*, z^*) = \begin{bmatrix} s_1 & s_2 & \cdots & s_l \\ -1 & -1 & \cdots & -1 \end{bmatrix} \quad \text{e} \quad \lambda_1^* = [\lambda_{1,1}^*, \lambda_{1,2}^*, \dots, \lambda_{1,l}^*]^t,$$

obtém-se

$$\sum_{i=1}^l \lambda_{1,i}^* s_i = 0 \quad \text{e} \quad \sum_{i=1}^l \lambda_{1,i}^* = 1.$$

Agora denota-se por $I = \{i \mid g_i^*(x^*, z^*) = 0\}$ o conjunto dos índices das restrições ativas e por $J = \{j \mid g_j^*(x^*, z^*) < 0\}$ o conjunto das restrições inativas em (x^*, z^*) . Com isso, segue que

$$\begin{aligned} g_i^*(x^*, z^*) \lambda_{1,i}^* &= 0 \quad \text{para todo } i \in I \\ g_j^*(x^*, z^*) \lambda_{1,j}^* &= 0 \quad \text{para todo } j \in J. \end{aligned}$$

Portanto $\lambda_{1,j}^* = 0$ para todo $j \in J$ e além disso

$$\sum_{i \in I} \lambda_{1,i}^* s_i = 0 \quad \text{e} \quad \sum_{i \in I} \lambda_{1,i}^* = 1.$$

Pelo Lema 5.18 tem-se que $s_i \in \partial F(x^*)$ para todo $i \in I$. Pela convexidade do subdiferencial e como $\lambda_{1,i}^* \geq 0$ pelo Lema 5.19 para todo $i \in I$ obtém-se que

$$0 = \sum_{i \in I} \lambda_{1,i}^* s_i \in \partial F(x^*).$$

□

5.4 Resultados Numéricos

Apresentamos agora alguns resultados numéricos que obtivemos com o método NFDNA, o qual programamos em Matlab - (R2012b) - em um microcomputador (TM) i5 de 2.60 GHz com 8.00 GB de RAM.

A performance do NFDNA foi testada através de um conjunto de problemas obtidos em Hock Schittkowski Collection [19]. Os problemas encontrados nesta referência são amplamente utilizados e foram cuidadosamente selecionados, dada a dificuldade de solução,

para se verificar a eficiência de novos algoritmos propostos para a solução de problemas de otimização, sejam eles diferenciáveis ou não. Todos os problemas, exceto o de Rosenbrock, são não diferenciáveis.

A tabela 1 apresenta os resultados obtidos pelo NFDNA quando aplicado aos 8 problemas lá relacionados. A solução de tais problemas teve como objetivo comprovar que a programação que fizemos do NFDNA estava correta.

Na Tabela 1 são utilizadas as seguintes abreviações:

n : Representa a dimensão do problema;

$+$: Representa problemas convexos;

$-$: Representa problemas não convexos;

x : Ponto encontrado pelo algoritmo;

x^* : Ponto ótimo;

f : Valor final da função objetivo obtido com o algoritmo;

f^* : Valor ótimo.

Problema	n	Convexo	x	x^*	f	f^*
Colville 1	5	-	(0.3, 0.3334, 0.4, 0.2241)	(0.3, 0.3334, 0.4, 0.4283, 0.2239)	-32.3486	-32.3486
Crescent	2	-	(-0.0163, -0.1297) · 10 ⁻⁴	(0, 0)	0	0
Rosenbrock	2	-	(0.9999, 0.9998)	(1, 1)	0	0
CB2	2	+	(1.1319, 0.8996)	(1.1393, 0.8994)	1.9522	1.9522
DEM	2	+	(0, -3)	(0,-3)	-2.9999	-3
Mifflin 1	2	+	(1, 0.0002)	(1, 0)	-1	-1
Rosen	4	+	(0.0001, 0.9998, 2, -1)	(0, 1, 2, -1)	-43.9999	-44
Shor	5	+	(1.1246, 0.9794, 1.4781, 0.9205, 1.1242)	(1.1243, 0.9794, 1.4777, 0.9202, 1.1242)	22.6001	22.6001

Tabela 1 – Resultados obtidos pelo NFDNA.

6 APLICAÇÃO

Apresentaremos uma aplicação do NFDNA a um método para resolver problemas de otimização não convexa, não diferenciável e com restrições, chamado Interior Epigraph Directions (IED) method, publicado por Burachik, Freire e Kaya [5].

Considere o problema

$$(P) \begin{cases} \text{minimizar} & f(x) \\ \text{sujeito a} & g(x) = 0 \\ & x \in X \end{cases}$$

onde $X \subset \mathbb{R}^n$ é compacto e as funções $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ são contínuas.

Apesar de (P) conter apenas restrições de igualdade, restrições de desigualdade podem ser consideradas por meio do operador $a^+ = \max\{a, 0\}$, $a \in \mathbb{R}$.

Uma opção para se resolver o problema (P) é a utilização da técnica de dualidade Lagrangeana, onde cada restrição é anexada a função objetivo por meio de um multiplicador.

Em um processo via dualidade Lagrangeana, em vez de se resolver o problema primal (P) , resolve-se o problema dual associado ao problema (P) , no qual a função dual é maximizada. Segundo esta abordagem, o valor da função dual em um ponto do espaço dual é calculado minimizando-se a Lagrangeana. A nossa proposta de aplicação consiste em utilizar o NFDNA na minimização desta Lagrangiana.

A função Lagrangiana associada ao problema primal (P) utilizada pelo IED é

$$L(x, (u, c)) = f(x) + c \sigma(g(x)) + \langle A u, g(x) \rangle \quad (6.1)$$

onde $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ é o multiplicador associado ao termo linear, $c \in \mathbb{R}_+ = [0, \infty)$ é o parâmetro de penalidade associado ao termo aumentado, $A \in \mathbb{R}^{m \times m}$ é a matriz real simétrica e $\sigma : \mathbb{R}^m \rightarrow \mathbb{R}$ uma função contínua que satisfaz $\sigma(x) > 0$, $\forall x \in \mathbb{R}^m \setminus \{0\}$ e $\sigma(0) = 0$.

O problema dual gerado pela Lagrangiana como em (6.1) é um problema convexo não diferenciável, que normalmente é resolvido por meio de técnicas de otimização convexa não diferenciável.

A abordagem baseada na Lagrangeana L para resolver (P) pode ser descrita do seguinte modo: Dado um ponto dual corrente (u^k, c^k) um ponto primal é calculado pela regra

$$x_k \in \text{Arg min}_{x \in X} L(x, (u^k, c^k)) \quad (6.2)$$

onde L é como em (6.1).

O ponto x_k obtido em (6.2) pode ser utilizado para calcular a direção de busca, que leva a próxima iteração dual (u^{k+1}, c^{k+1}) . Além disso, x_k é utilizado para obter uma

deflexão do subgradiente que por sua vez é utilizado para melhorar os valores duais. Esta é a ideia por trás dos métodos estudados em [1, 4, 6, 7, 8, 9 e 13]. Tais métodos são conhecidos como deflected subgradients (DSG) methods.

A versão original do método IED [5] combina o método DSG com o Nonsmooth Feasible Direction Algorithm (NFDA).

Apresentaremos de forma resumida, como funciona a versão original do método IED. Inicia-se em um ponto no interior do epígrafo da função dual. Em seguida, uma deflexão na direção dos subgradientes é utilizada para definir uma aproximação linear ao epígrafo. Um ponto auxiliar é obtido resolvendo-se as condições de otimização de um problema linear. Se o ponto auxiliar pertence ao interior do epígrafo, é declarado passo sério e o ponto auxiliar torna-se o próximo ponto a partir do qual o processo é repetido. Se o ponto auxiliar não pertence ao interior do epígrafo, o passo é declarado nulo e uma versão adequada do passo do DSG é aplicada a partir do ponto original. Devido às propriedades do DSG este ponto permanece no interior do epígrafo. O novo ponto substitui o ponto inicial e o processo é repetido novamente até que um passo sério seja realizado. Se tivermos apenas a ocorrência de passos nulos ocorrer, o IED converge para uma solução primal, como uma consequência das propriedades de convergência do DSG.

O IED sempre gera uma sequência no interior do epígrafo da função dual, bem como uma sequência primal. Isso motivou o nome Interior Epigraph Directions (IED).

O passo obtido resolvendo-se as condições de otimização de aproximação linear do problema dual foi inspirado pelo NFDA.

Minimizar a Lagrangeana, isto é, obter x^k como em (6.2) de forma eficaz é fundamental para o sucesso dos métodos com base na dualidade Lagrangeana.

Na versão original do IED [5], a rotina "fminsearch" do MatLab foi usada para minimizar a Lagrangeana. A rotina fminsearch se aplica a problemas de otimização não linear irrestritos e é baseada no algoritmo simplex Nelder-Mead que é um método que não utiliza o cálculo de gradientes ou hessianas. Recomendamos a referência [23] para maiores detalhes sobre este assunto.

Aqui, substituímos a rotina fminsearch do Matlab pelo NFDNA e comparamos o desempenho desta nova versão do IED com o desempenho da versão original, resolvendo alguns problemas testes. Também resolvemos alguns outros problemas para testar a eficácia do novo algoritmo.

6.1 Problemas Testes Utilizados

PPR-P1-2

Dimensão	3
Função objetivo	$f(x) = 5x_1 + 50000/x_1 + 20x_2 + 72000/x_2 + 10x_3 + 144000/x_3$
Restrições	$f_1(x) = \max(-1 + 4/x_1 + 32/x_2 + 120/x_3, 0)$ $f_2(x) = \max(1e^{-5} - x_1, 0)$ $f_3(x) = \max(1e^{-5} - x_2, 0)$ $f_4(x) = \max(1e^{-5} - x_3, 0)$
Ponto inicial	$x_0 = [1 \ 1 \ 1]$
Ponto ótimo	$x^* = [108.73472 \ 85.12614 \ 204.32471]$

LGR-P1-1

Dimensão	3
Função objetivo	$f(x) = 0.2x_3 - 0.8x_1$
Restrições	$f_1(x) = \max(\exp(x_1) - x_2, 0)$ $f_2(x) = \max(\exp(x_2) - x_3, 0)$ $f_3(x) = \max(x_1 - 100, 0)$ $f_4(x) = \max(-x_1, 0)$ $f_5(x) = \max(x_2 - 100, 0)$ $f_6(x) = \max(-x_2, 0)$ $f_7(x) = \max(x_3 - 10, 0)$ $f_8(x) = \max(-x_3, 0)$
Ponto inicial	$x_0 = [1 \ 1 \ 1]$
Ponto ótimo	$x^* = [0.18412 \ 1.2016 \ 3.32732]$

QQR-T1-8

Dimensão	3
Função objetivo	$f(x) = x_1^2 + x_2^2 + x_3^2$
Restrições	$f_1(x) = \max(-x_1^2 - x_2^2 + 1, 0)$ $f_2(x) = \max(1 - x_1, 0)$ $f_3(x) = \max(x_1 - 10, 0)$ $f_4(x) = \max(-10 - x_2, 0)$ $f_5(x) = \max(x_2 - 10, 0)$ $f_6(x) = \max(-10 - x_3, 0)$ $f_7(x) = \max(x_3 - 10, 0)$
Ponto inicial	$x_0 = [1 \ 1 \ 1]$
Ponto ótimo	$x^* = [1 \ 0 \ 0]$

GPR-P1-2

Dimensão	5
Função objetivo	$f(x) = \exp(x_1x_2x_3x_4x_5) - 0.5(x_1^3 + x_2^3 + 1)^2$
Restrições	$f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10$ $f_2(x) = x_2x_3 - 5x_4x_5$ $f_3(x) = x_1^3 + x_2^3 + 1$ $f_4(x) = \max(x_1 - 2.3, 0)$ $f_5(x) = \max(-2.3 - x_1, 0)$ $f_6(x) = \max(x_2 - 2.3, 0)$ $f_7(x) = \max(-2.3 - x_2, 0)$ $f_8(x) = \max(x_3 - 3.2, 0)$ $f_9(x) = \max(-3.2 - x_3, 0)$ $f_{10}(x) = \max(x_4 - 3.2, 0)$ $f_{11}(x) = \max(-3.2 - x_4, 0)$ $f_{12}(x) = \max(x_5 - 3.2, 0)$ $f_{13}(x) = \max(-3.2 - x_5, 0)$
Ponto inicial	$x_0 = [-2 \ 2 \ 2 \ -1 \ -1]$
Ponto ótimo	$x^* = [-1.7171 \ 1.1596 \ 1.8272 \ -0.7636 \ -0.7636]$

PPR-P1-3

Dimensão	4
Função objetivo	$f(x) = x_1x_4(x_1 + x_2 + x_3) + x_3$
Restrições	$f_1(x) = \max(25 - x_1x_2x_3x_4, 0)$ $f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 40$ $f_3(x) = \max(1 - x_1, 0)$ $f_4(x) = \max(x_1 - 5, 0)$ $f_5(x) = \max(1 - x_2, 0)$ $f_6(x) = \max(x_2 - 5, 0)$ $f_7(x) = \max(1 - x_3, 0)$ $f_8(x) = \max(x_3 - 5, 0)$ $f_9(x) = \max(1 - x_4, 0)$ $f_{10}(x) = \max(x_4 - 5, 0)$
Ponto inicial	$x_0 = [5 \ 5 \ 5 \ 5]$
Ponto ótimo	$x^* = [1 \ 4.7430 \ 3.8211 \ 1.3794]$

LPR-P1-1

Dimensão	4
Função objetivo	$f(x) = 1 + x_1 + x_2 + x_3 + x_4$
Restrições	$f_1(x) = \max(4/x_1 + 2.25/x_2 + 1/x_3 + 0.25/x_4 - 0.0401, 0)$ $f_2(x) = \max(0.16/x_1 + 0.36/x_2 + 0.64/x_3 + 0.64/x_4 - 0.010085, 0)$ $f_3(x) = \max(x_1 - 4e5, 0)$ $f_4(x) = \max(0.001 - x_1, 0)$ $f_5(x) = \max(x_2 - 3e5, 0)$ $f_6(x) = \max(0.001 - x_2, 0)$ $f_7(x) = \max(x_3 - 2e5, 0)$ $f_8(x) = \max(0.001 - x_3, 0)$ $f_9(x) = \max(x_4 - 1e5, 0)$ $f_{10}(x) = \max(0.001 - x_4, 0)$
Ponto inicial	$x_0 = [1 \ 1 \ 1 \ 1]$
Ponto ótimo	$x^* = [193.4071 \ 179.5475 \ 185.0186 \ 168.7062]$

QLR-T1-3

Dimensão	3
Função objetivo	$f(x) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$
Restrições	$f_1(x) = \max(-3 + x_1 + x_2 + 2x_3, 0)$ $f_2(x) = \max(-x_1, 0)$ $f_3(x) = \max(-x_2, 0)$ $f_4(x) = \max(-x_3, 0)$
Ponto inicial	$x_0 = [0.5 \ 0.5 \ 0.5]$
Ponto ótimo	$x^* = [1.3333 \ 0.7777 \ 0.4444]$

QQR-T1-3

Dimensão	2
Função objetivo	$f(x) = 0.5x_1^2 + x_2^2 - x_2x_1 - 7x_1 - 7x_2$
Restrições	$f_1(x) = \max(-25 + 4x_1^2 + x_2^2, 0)$
Ponto inicial	$x_0 = [0 \ 0.5]$
Ponto ótimo	$x^* = [2 \ 3]$

PPR-P1-3

Dimensão	4
Função objetivo	$f(x) = x_1x_4(x_1 + x_2 + x_3) + x_3$
Restrições	$f_1(x) = \max(-x_1x_2x_3x_4 + 25, 0)$ $f_2(x) = \max(-x_1^2 - x_2^2 - x_3^2 - x_4^2 + 40, 0)$ $f_3(x) = \max(-x_1 + 1, 0)$ $f_4(x) = \max(x_1 - 5, 0)$ $f_5(x) = \max(-x_2 + 1, 0)$ $f_6(x) = \max(x_2 - 5, 0)$ $f_7(x) = \max(-x_3 + 1, 0)$ $f_8(x) = \max(x_3 - 5, 0)$ $f_9(x) = \max(-x_4 + 1, 0)$ $f_{10}(x) = \max(x_4 - 5, 0)$
Ponto inicial	$x_0 = [1 \ 5 \ 5 \ 1]$
Ponto ótimo	$x^* = [1 \ 4.7429 \ 3.8211 \ 1.3794]$

QPR-T1-1

Dimensão	2
Função objetivo	$f(x) = (x_1 - 2)^2 + x_2^2$
Restrições	$f_1(x) = \max(x_2 - (1 - x_1)^3, 0)$ $f_2(x) = \max(-x_1, 0)$ $f_3(x) = \max(-x_2, 0)$
Ponto inicial	$x_0 = [-2 \ -2]$
Ponto ótimo	$x^* = [1 \ 0]$

QQR-P1-3

Dimensão	3
Função objetivo	$f(x) = (x_1 - x_2)^2 + (x_1 + x_2 - 10)^2 + (x_3 - 5)^2$
Restrições	$f_1(x) = \max(-48 + x_1^2 + x_2^2 + x_3^2, 0)$ $f_2(x) = \max(-4.5 - x_1, 0)$ $f_3(x) = \max(x_1 - 4.5, 0)$ $f_4(x) = \max(-4.5 - x_2, 0)$ $f_5(x) = \max(x_2 - 4.5, 0)$ $f_6(x) = \max(-5 - x_3, 0)$ $f_7(x) = \max(x_3 - 5, 0)$
Ponto inicial	$x_0 = [-5 \ 5 \ 0]$
Ponto ótimo	$x^* = [3.6504 \ 3.6504 \ 4.6204]$

QLR-P1-1

Dimensão	4
Função objetivo	$f(x) = x_1^2 + 0.5x_2^2 + x_3^2 + 0.5x_4^2 - x_1x_3 + x_3x_4 - x_1 - 3x_2 + x_3 - x_4$
Restrições	$f_1(x) = \max(-5 + x_1 + 2x_2 + x_3 + x_4, 0)$ $f_2(x) = \max(-4 + 3x_1 + x_2 + 2x_3 - x_4, 0)$ $f_3(x) = \max(-x_2 - 4x_3 + 1.5, 0)$ $f_4(x) = \max(-x_1, 0)$ $f_5(x) = \max(-x_2, 0)$ $f_6(x) = \max(-x_3, 0)$ $f_7(x) = \max(-x_4, 0)$
Ponto inicial	$x_0 = [0.5 \ 0.5 \ 0.5 \ 0.5]$
Ponto ótimo	$x^* = [0.2727 \ 2.0909 \ -0.26 \cdot 10^{-10} \ 0.5454]$

PQR-T1-7

Dimensão	3
Função objetivo	$f(x) = -x_1x_2x_3$
Restrições	$f_1(x) = \max(x_1^2 + 2x_2^2 + 4x_3^2 - 48, 0)$
Ponto inicial	$x_0 = [1 \ 1 \ 1]$
Ponto ótimo	$x^* = [4 \ 2\sqrt{2} \ 2]$

PBR-T1-1

Dimensão	2
Função objetivo	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
Restrições	$f_1(x) = \max(-x_2 - 1.5, 0)$
Ponto inicial	$x_0 = [-2 \ 1]$
Ponto ótimo	$x^* = [1 \ 1]$

QQR-T1-6

Dimensão	2
Função objetivo	$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$
Restrições	$f_1(x) = \max(x_1 + x_2 - 2, 0)$ $f_2(x) = \max(x_1^2 - x_2, 0)$
Ponto inicial	$x_0 = [2 \ 2]$
Ponto ótimo	$x^* = [1 \ 1]$

6.2 Experimentos Numéricos

Nesta seção, vamos mostrar os resultados obtidos pela nossa versão do método IED que utiliza o NFDNA na minimização da Lagrangeana. Chamamos esta versão de IED/NFDNA.

Primeiramente comparamos os resultados para cinco problemas obtidos pela versão original do IED, chamado aqui de IED/fmin.

As tabelas 2, 3 e 4 fornecem os pontos iniciais usados pelo IED/NFDNA. A tabela 5 compara os resultados obtidos pelas duas versões do algoritmo e as tabelas 6 e 7 apresentam os resultados de dez problemas resolvidos pelo IED/NFDNA, que podem ser comparados com as soluções dadas na tabela 8.

Aqui, o objetivo foi comparar a performance das duas versões do IED ao que tange ao volume de cálculo da função Lagrangeana. A quantidade de vezes que um algoritmo calcula as funções envolvidas no problema bem como o número de iterações são medidas de sua eficiência quando se trata de otimização não linear.

Todos os problemas testes foram retirados do Hock Schittkowski Collection [19]. As restrições de desigualdade da forma $g(x) \leq 0$ foram substituídas por $\max\{g(x), 0\}$.

Executamos os experimentos numéricos, considerando dois casos, a saber:

- (i) $A = I$, isto é, utilizando a Lagrangiana aumentada;
- (ii) $A = 0$, empregando apenas uma função de penalidade.

Observamos na tabela 5 que o número de vezes que a função Lagrangeana foi calculada pela versão IED \ NFDNA foi sempre inferior em comparação com a versão IED \ fmin. Também observamos pelas tabelas 5, 6 e 7 que a norma $\max(\|\cdot\|_2, \|\cdot\|_2^2)$ nos permitiu encontrar uma solução em todos os problemas independentemente da escolha da matriz A . Os casos em que a solução não foi encontrada foram indicadas por — — —.

É importante mencionarmos que todos os problemas resolvidos têm Lagrangianas que satisfazem as hipóteses exigidas pelo NFDNA.

O algoritmo foi implementado em Matlab - (R2012b) - em um microcomputador (TM) i5 de 2.60 GHz com 8.00 GB de RAM.

A	$\sigma(\cdot)$	PPR-P1-2		LGR-P1-1		GPR-P1-2		PPR-P1-3		LPR-P1-1	
		c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0
I	$\ \cdot\ _2$	0	[1111]	1	[00000000]	3	[000000000000]	10	[1111111111]	2	[0000000000]
	$\ \cdot\ _1$	0	[1111]	0.1	[00000000]	3	[000000000000]	10	[1111111111]	2	[0000000000]
	$\sqrt{m}\ \cdot\ _\infty$	0	[1111]	1	[00000000]	3	[000000000000]	1	[0000000000]	2	[0000000000]
	$e^{\ \cdot\ _2} - 1$	0	[1111]	3	[00000000]	1	[555555555555]	10	[1111111111]	0.1	[0000000000]
	$e^{\ \cdot\ _1} - 1$	0	[1111]	3	[11111111]	3	[000000000000]	10	[1111111111]	0.05	[0000000000]
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0	[1111]	2	[11111111]	3	[000000000000]	10	[1111111111]	0.1	[0000000000]
0	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0	[1111]	0.1	[00000000]	3	[111111111111]	1	[0000000000]	2	[0000000000]
	$\ \cdot\ _2$	0	[1111]	1	[00000000]	3	[000000000000]	15	[1111111111]	2	[0000000000]
	$\ \cdot\ _1$	0	[1111]	0.1	[00000000]	3	[000000000000]	10	[1111111111]	2	[0000000000]
	$\sqrt{m}\ \cdot\ _\infty$	0	[1111]	1	[00000000]	3	[000000000000]	1	[0000000000]	2	[0000000000]
	$e^{\ \cdot\ _2} - 1$	0	[1111]	3	[00000000]	0.5	[555555555555]	10	[1111111111]	0.05	[0000000000]
	$e^{\ \cdot\ _1} - 1$	0	[1111]	3	[11111111]	3	[000000000000]	10	[1111111111]	0.05	[0000000000]
0	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0	[1111]	3	[11111111]	3	[000000000000]	10	[1111111111]	0.1	[0000000000]
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0	[1111]	1	[00000000]	1	[111111111111]	1	[0000000000]	2	[0000000000]

Tabela 2 – Estimativas iniciais usados pelo IED / NFDNA.

A	$\sigma(\cdot)$	QQR-T1-3		GLR-T1-3		PPR-P1-3		QQR-T1-8		QPR-T1-1	
		c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0
I	$\ \cdot\ _2$	3	1	2	[1111]	3	[1111]	3	[1111111]	2	[111]
	$\ \cdot\ _1$	3	1	2	[1111]	3	[1111]	3	[1111111]	2	[111]
	$\sqrt{m}\ \cdot\ _\infty$	3	1	2	[1111]	4	[2222]	3	[1111111]	2	[111]
	$e^{\ \cdot\ _2} - 1$	3	1	2	[1111]	0.01	[1111]	0	-[1111111]	0.01	[111]
	$e^{\ \cdot\ _1} - 1$	3	1	2	[1111]	0.01	[1111]	3	[1111111]	0.000001	[111]
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	3	1	2	[1111]	0.01	[1111]	3	[1111111]	1	[111]
0	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	3	1	2	[1111]	3	[1111]	3	[1111111]	1	[111]
	$\ \cdot\ _2$	3	1	2	[1111]	3	[1111]	3	[1111111]	2	[111]
	$\ \cdot\ _1$	3	1	2	[1111]	3	[1111]	3	[1111111]	2	[111]
	$\sqrt{m}\ \cdot\ _\infty$	3	1	2	[1111]	4	[2222]	3	[1111111]	2	[111]
	$e^{\ \cdot\ _2} - 1$	3	1	2	[1111]	0.01	[1111]	3	[1111111]	0.001	[111]
	$e^{\ \cdot\ _1} - 1$	3	1	2	[1111]	0.01	[1111]	3	[1111111]	0	[111]
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	3	1	2	[1111]	0.01	[1111]	3	[1111111]	0	[111]
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	3	1	2	[1111]	3	[1111]	3	[1111111]	0	[111]
		3	1	2	[1111]	3	[1111]	3	[1111111]	0.01	[111]

Tabela 3 – Estimativas iniciais usados pelo IED / NFDNA.

A	$\sigma(\cdot)$	QQR-P1-3		QLR-P1-1		PQR-T1-7		PBR-T1-1		QQR-T1-6	
		c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0	c_0	u_0
I	$\ \cdot\ _2$	2	[0000000]	4	[1111111]	2	1	1	1	1	[11]
	$\ \cdot\ _1$	2	[1111111]	4	[1111111]	2	1	1	1	1	[11]
	$\sqrt{m}\ \cdot\ _\infty$	2	[1111111]	10	[1111111]	2	1	1	1	1	[11]
	$e^{\ \cdot\ _2} - 1$	0	-[1111111]	5	[1111111]	2	1	1	1	0.1	[11]
	$e^{\ \cdot\ _1} - 1$	0.1	[0000000]	4	[1111111]	2	1	1	1	0.01	[11]
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.1	[0000000]	2	[1111111]	2	1	1	1	0.1	[11]
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	2	[1111111]	4	[1111111]	2	1	1	1	0.1	[11]
	$\ \cdot\ _2$	2	[0000000]	4	[1111111]	2	1	1	1	1	[11]
	$\ \cdot\ _1$	2	[1111111]	10	[1111111]	2	1	1	1	1	[11]
	$\sqrt{m}\ \cdot\ _\infty$	2	[1111111]	15	[2222222]	2	1	1	1	1	[11]
0	$e^{\ \cdot\ _2} - 1$	2	[1111111]	5	[1111111]	1	1	1	1	0.1	[11]
	$e^{\ \cdot\ _1} - 1$	0.1	[0000000]	5	[1111111]	1	1	1	1	0.1	[11]
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.1	[0000000]	2	[0000000]	1	1	1	1	0.1	[11]
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	2	[1111111]	4	[1111111]	2	1	1	1	0.1	[11]

Tabela 4 – Estimativas iniciais usados pelo IED / NFDNA.

A	$\sigma(\cdot)$	PPR-P1-2				LGR-P1-1				GPR-P1-2				PPR-P1-3				LPR-P1-1			
		fmin	n_L	n_{its}	NFDNA	fmin	n_L	n_{its}	NFDNA	fmin	n_L	n_{its}	NFDNA	fmin	n_L	n_{its}	NFDNA	fmin	n_L	n_{its}	NFDNA
I	$\ \cdot\ _2$	3	1306	2	102	—	—	1	111	—	—	1	84	—	—	1	118	7	4166	7	115
	$\ \cdot\ _1$	3	1306	2	102	4	1877	1	111	—	—	—	—	—	—	—	—	7	5148	7	115
	$\sqrt{m}\ \cdot\ _\infty$	3	993	2	102	2	1482	1	76	—	—	—	—	—	—	2	118	8	7764	3	115
	$e^{\ \cdot\ _2} - 1$	—	—	2	102	3	1039	1	76	2	1173	1	123	3	5105	—	—	7	3648	7	115
	$e^{\ \cdot\ _1} - 1$	—	—	2	102	4	2180	1	76	2	3941	—	—	2	5797	—	—	7	4266	5	115
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	—	—	2	102	2	1196	1	111	4	5352	—	—	2	4813	—	—	8	4934	5	115
0	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	1	1145	2	102	2	1582	1	76	2	3060	2	123	2	2254	44	118	7	4420	9	115
	$\ \cdot\ _2$	3	1345	2	102	—	—	1	111	—	—	1	84	—	—	1	118	8	4354	10	115
	$\ \cdot\ _1$	3	1345	2	102	4	1877	1	111	—	—	—	—	—	—	—	—	7	4984	10	115
	$\sqrt{m}\ \cdot\ _\infty$	3	1236	2	102	2	1482	1	76	—	—	—	—	—	—	1	118	8	6237	5	115
	$e^{\ \cdot\ _2} - 1$	—	—	2	102	3	1039	1	76	1	3048	1	123	2	1663	—	—	7	4555	9	115
	$e^{\ \cdot\ _1} - 1$	—	—	2	102	4	2180	1	76	1	1790	—	—	2	3769	—	—	7	3926	5	115
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	—	—	2	102	2	1196	1	76	1	3484	—	—	2	5637	—	—	8	5993	5	115
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	3	1309	2	102	3	1303	1	76	1	3060	1	123	2	2890	2	118	7	4252	9	115

Tabela 5 – Comparação dos resultados obtidos por ambas as versões do algoritmo. m_{its} denota o número de iterações e n_L denota o número de vezes que a função Lagrangeana foi calculada.

A	$\sigma(\cdot)$	QPR-TI-1			PQR-TI-7			QQR-TI-8			QQR-P1-3			QQR-TI-3		
		n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_L
I	$\ \cdot\ _2$	3362	0.9818	2	3013	-22.6274	1	532	1	381	0.9666	1	63	-30		
	$\ \cdot\ _1$	22844	0.9885	2	3013	-22.6274	1	21	1	478	0.9535	1	63	-30		
	$\sqrt{m}\ \cdot\ _\infty$	7375	1.0538	2	3013	-22.6274	1	2285	1	641	1.0613	1	63	-30		
	$e^{\ \cdot\ _2} - 1$	—	—	1	5665	-22.6274	—	—	—	1083	0.9541	—	—	—		
	$e^{\ \cdot\ _1} - 1$	792	0.9811	1	7131	-22.6274	1	21	1	5439	0.9540	—	—	—		
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	18063	1.1801	1	7131	-22.6274	1	261	1.0003	4168	0.9535	—	—	—		
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	763	0.9893	5	2359	-22.6274	1	669	1	143	1.0518	1	63	-30		
	$\ \cdot\ _2$	—	—	1	4880	-22.6274	1	179	1.0015	381	0.9666	1	63	-30		
	$\ \cdot\ _1$	325	0.9925	1	4880	-22.6274	1	4293	1	520	0.9578	1	63	-30		
	$\sqrt{m}\ \cdot\ _\infty$	—	—	1	4880	-22.6274	1	1940	1	717	0.9544	1	63	-30		
0	$e^{\ \cdot\ _2} - 1$	8728	0.9897	1	2288	-22.6274	—	—	—	—	—	—	—	—		
	$e^{\ \cdot\ _1} - 1$	—	—	1	2288	-22.6274	1	290	1	6686	0.9564	—	—	—		
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	57201	1.3005	1	2288	-22.6274	1	214	1	4168	0.9535	—	—	—		
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	353	1.0445	1	10167	-22.6274	1	216	1.0098	360	1.1377	2	63	-30		

Tabela 6 – Alguns resultados obtidos pelo IED / NFDNA. n_{its} denota o número de iterações e n_L denota o número de vezes que a função Lagrangeana foi calculada.

A	$\sigma(\cdot)$	QLR-T1-3			QLR-P1-1			PPR-P1-7			PBR-T1-1			QQR-T1-6		
		n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_{its}	n_L	Resultados	n_L
I	$\ \cdot\ _2$	69	0.1112	1	71	-4.6780	1	69	680.6301	1	6368	0	1	116	1.0181	
	$\ \cdot\ _1$	69	0.1129	1	71	-4.6763	1	69	680.6314	1	6368	0	1	109	1.0002	
	$\sqrt{m}\ \cdot\ _\infty$	69	0.1113	1	71	-4.0723	1	69	680.6306	1	6368	0	1	301	1.2478	
	$e^{\ \cdot\ _2} - 1$	69	0.1111	1	71	-4.6047	1	69	705.0114	1	6047	0	2	451	1	
	$e^{\ \cdot\ _1} - 1$	69	0.1377	1	71	-4.5652	1	69	704.4187	1	6047	0	2	10087	1.0018	
	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	69	0.1122	1	71	-4.4560	—	—	—	—	1	6047	0	2	758	1.0027
0	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	69	0.1118	1	71	-4.6773	1	69	680.6301	1	6911	0	9	4016	1	
	$\ \cdot\ _2$	69	0.1111	1	71	-4.6251	1	69	680.6302	1	5391	0	2	718	1.0258	
	$\ \cdot\ _1$	69	0.1111	1	71	-4.6779	1	69	680.6327	1	5391	0	1	382	1.0013	
	$\sqrt{m}\ \cdot\ _\infty$	69	0.1111	1	71	-4.0619	1	69	680.6301	1	5391	0	2	469	1.0080	
	$e^{\ \cdot\ _2} - 1$	69	0.1111	1	71	-4.6587	1	69	705.0114	1	5391	0	1	254	1.0378	
	$e^{\ \cdot\ _1} - 1$	69	0.1184	1	71	-4.6712	1	69	704.4187	1	5391	0	1	251	1.0643	
0	$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	69	0.1169	1	71	-1.4746	—	—	—	1	5391	0	1	211	1.2135	
	$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	69	0.1112	1	71	-4.6676	1	69	680.6302	1	5391	0	1	157	1.0908	

Tabela 7 – Alguns resultados obtidos pelo IED / NFDNA. n_{its} denota o número de iterações e n_L denota o número de vezes que a função Lagrangeana foi calculada.

Problemas	Valor Ótimo
QQR-T1-3	-30
QLR-T1-3	0.1111
QLR-P1-1	-4.6818
PPR-P1-7	680.6300
QPR-T1-1	1
PQR-T1-7	-22.6274
QQR-T1-8	1
QQR-P1-3	0.9535
QQR-T1-6	1
PBR-T1-1	0

Tabela 8 – Valor ótimo.

7 Conclusão

Neste trabalho, primeiramente, estudamos um algoritmo para minimização não convexa e não diferenciável denominado NFDNA.

Posteriormente, programamos e testamos o algoritmo para então alcançarmos nosso principal objetivo o qual foi incorporar o NFDNA ao IED para a minimização da função Lagrangeana em substituição à rotina `fminsearch` do Matlab.

Através dos resultados que obtivemos, podemos concluir que tal substituição melhorou a performance do IED, a qual depende fortemente do método empregado para a minimização da Lagrangeana.

Para trabalhos futuros, sugerimos a busca de novos algoritmos para a minimização da Lagrangeana ou até mesmo uma modificação do NFDNA para tal fim.

REFERÊNCIAS

- [1] BAGIROV, A. M.; GANJEHLOU, A. N. A quasisecant method for minimizing nonsmooth functions. *Optim. Methods Softw*, 25, p. 3–18, 2010.
- [2] BAZARAA, M.S.; SHERALI, H.D.; SHETTY, C.M. *Nonlinear Programming - Theory and Algorithms*. New York: John Wiley and Sons, 2ed, 1983.
- [3] BONNANS, J. F., GILBERT, J. C., LEMARÉCHAL, C., et al. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Verlag, 2003.
- [4] BURACHIK, R.S. On primal convergence for augmented Lagrangian duality. *Optim*, 60(8-9), p. 979–990, 2011.
- [5] BURACHIK, R.S.; FREIRE, W.P.; KAYA, C.Y. Interior Epigraph Directions Method for Nonsmooth and Nonconvex Optimization via Generalized Augmented Lagrangian Duality. *J. Global Opt.*, 60, p. 501-529, 2014.
- [6] BURACHIK, R. S.; GASIMOV, R. N.; ISMAYILOVA, N. A.; KAYA, C. Y. On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian. *J. Global Opt.*, 34, p. 55–78, 2006.
- [7] BURACHIK, R. S.; KAYA, C. Y. An update rule and a convergence result for a penalty function method. *J. Ind. Man. Opt.*, 3, p. 381–398, 2007.
- [8] BURACHIK, R. S.; IUSEM, A. N.; MELO, J. G. A primal dual modified subgradient algorithm with sharp Lagrangian. *J. Global Optim.*, 46, p. 347–361, 2010.
- [9] BURACHIK, R. S.; IUSEM, A. N.; MELO, J. G. Strong duality and exact penalization for general augmented Lagrangians, *J. Optim. Theory Appl.*, 147, p. 125–140, 2010.
- [10] CHENEY, E. W.; GOLDSTEIN, A. A. Newton’s method for convex programming and Tchebycheff approximation. *Numeriche Mathematic*, v.1, p. 253–268, 1959.
- [11] CLARKE, F. H. *Optimization and Nonsmooth Analysis*. New York: John Wiley and Sons, 1983.
- [12] CLARKE, F., LEDYAEV, Y. S., STERN, R., et al. *Nonsmooth Analysis and Control Theory*. Springer, 1998.
- [13] DANSKIN, J. M. The theory of max-min with applications. *Siam J. Appl. Math.*, 14, p. 641–664, 1966.
- [14] FLETCHER, R. *Practical Methods of Optimization*. Jhon Wiley and Sons, 1987.
- [15] FREIRE, P. W. Um Algoritmo de Direções Viáveis para Otimização Convexa Não Diferenciável. Tese de Doutorado, COPPE – Universidade Federal do Rio de Janeiro, 2005.
- [16] FUDULI, A., GAUDIOSO, M., GIALLOMBARDO, G. “Minimizing nonconvex nonsmooth functions via cutting planes and proximity control”, *SIAM Journal Optimization*, v. 14, pp. 743–756, 2004.

- [17] HERSKOVITS, J. A Two-Stage Feasible Directions Algorithm for Nonlinear Constrained Optimization. *Mathematical Programming*, v.36, p. 19-38, 1986.
- [18] HERSKOVITS, J. A View on Nonlinear Optimization, ed. Herskovits, J. *Advances in Structural Optimization*, Kluwer Academic Publishers, p. 71-116, 1995.
- [19] HERSKOVITS, J. A Feasible Directions Interior Point Technique for Nonlinear Optimization, JOTA. *Journal of Optimization Theory and Applications*, v. 99, p. 121-146, 1998.
- [20] IZMAILOV, A. F., SOLODOV, M. V. Otimização Volume 2 - Métodos Computacionais. IMPA, 2007.
- [21] KELLEY, J. J. E. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, v.8, p. 703-712, 1960.
- [22] KIWIEL, K. C. Methods of Descent for Non-differentiable Optimization. *Springer-Verlag*, 1985.
- [23] Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, *SIAM Journal of Optimization*, Vol. 9, Number 1, pp. 112-147, 1998.
- [24] LEMARECHAL, C. Bundle Method in Nonsmooth Optimization. *Nonsmooth Optimization: Proceedings of IIASA Workshop*, C. Lemarechal and R. Mifflin, eds., p. 79-109, 1978.
- [25] LUENBERGER, D.G. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 2 ed., 1984.
- [26] MAKELA, M.; NEITTAANMAKI, P. Nonsmooth Optimization, Analysis and Algorithms with Applications to Optimal Control. *Word Scientific Publishing*, 1992.
- [27] MIFFLIN, R. An Algorithm for Constrained Optimization with Semismooth Functions. *Mathematics of Operations Research*, 2, p. 191-207, 1977.
- [28] MOREAU, J., PANAGIOTOPOULOS, P., STRANG, G. Topics in Nonsmooth Mechanics. Birkhäuser Verlag, Basel, 1988.
- [29] NOCEDAL, J.; WRIGHT, S.J. *Numerical Optimization*. New York: Springer-Verlag, 1999.
- [30] OUTRATA, J., KOČVARA, M., ZOWE, J. Nonsmooth Approach to Optimization Problems With Equilibrium Constraints. Theory, Applications and Numerical Results. Kluwer Academic Publishers, 1998.
- [31] PERESSINI, A.L.; SULLIVAN, F.E.; UHL, J.J.,JR. *The Mathematics of Nonlinear Programming*. New York: Springer-Verlag, 1988.
- [32] ROCKAFELLAR, R.T. *Convex Analysis*. New Jersey: Princeton University Press, 1970.
- [33] SCHRAMM, H., ZOWE, J. "A version of the bundle idea for minimizing a nonsmooth functions: conceptual idea, convergence analysis, numerical results", *SIAM Journal on Optimization*, v. 2, pp. 121-152, 1992.

- [34] Shor, N.Z., Minimization Methods for Nondifferentiable Functions, Springer-Verlag, Berlin, 1985.
- [35] TANAKA M. Algoritmos de Direções Viáveis para Otimização Não Diferenciável, Tese de Doutorado, COPPE – Rio de Janeiro, 2011.
- [36] VLČEK, J, LUKŠAN, L.. Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, N-798. Relatório técnico, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.