

Universidade Estadual de Campinas

INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA

Departamento de Matemática Aplicada

Tese de Doutorado

**Um Método do tipo Lagrangiano
Aumentado
com Região de Confiança**

por

Emerson Vitor Castelani [†]

Doutorado em Matemática Aplicada - Campinas - SP

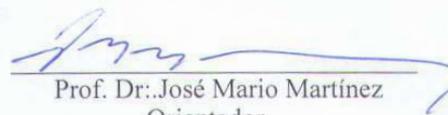
Orientador: Prof. Dr. José Mario Martínez

[†]Este trabalho contou com apoio financeiro do CNPq.

UM MÉTODO DO TIPO LAGRANGIANO AUMENTADO COM REGIÃO DE
CONFIANÇA

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Emerson Vitor Castellani e aprovada pela comissão julgadora.

Campinas, 07 de agosto de 2009


Prof. Dr.: José Mario Martínez
Orientador

Banca Examinadora:

Prof. Dr. José Mario Martínez
Prof. Dr. Roberto Andreani
Prof. Dr. Fermin Sinforiano Viloche Bazan
Prof. Dr. Ernesto Julián Goldberg Birgin
Profa. Dra. Elizabeth Wegner Karas

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de Doutor em Matemática Aplicada.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8 / 5094

Castelani, Emerson Vitor

C276m Um método do tipo Lagrangiano Aumentado com Região de Confiança/Emerson Vitor Castelani-- Campinas, [S.P. : s.n.], 2009.

Orientador : José Mario Martínez

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

I. Programação não-linear. 2. Otimização matemática. 3. Método de região de confiança. 4. Lagrange, Funções de. I. Martínez, José Mario. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: On Augmented Lagrangian methods with Trust-Region

Palavras-chave em inglês (Keywords): 1. Nonlinear Programming. 2. Mathematics Optimization. 3. Trust-Region methods. 4. Lagrange functions.

Área de concentração: Otimização

Titulação: Doutor em Matemática Aplicada

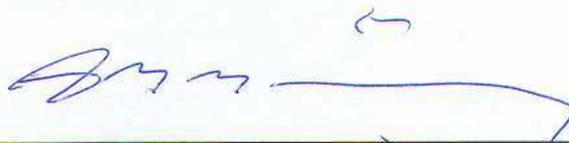
Banca examinadora: Prof. Dr. José Mario Martínez (IMECC-UNICAMP)
Prof. Dr. Roberto Andreani (IMECC-UNICAMP)
Prof. Dr. Fermin Sinfoniano Viloche Bazan (UFSC)
Prof. Dr. Ernesto Julián Goldberg Birgin (IME-USP)
Profa. Dra. Elizabeth Wegner Karas (UFPR)

Data da defesa: 17/06/2009

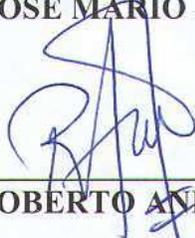
Programa de Pós-Graduação: Doutorado em Matemática Aplicada

Tese de Doutorado defendida em 17 de junho de 2009 e aprovada

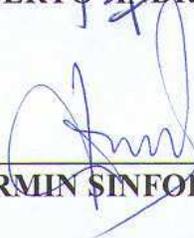
Pela Banca Examinadora composta pelos Profs. Drs.



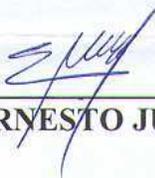
Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ



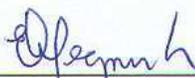
Prof(a). Dr(a). ROBERTO ANDREANI



Prof(a). Dr(a). FERMIN SINFORIANO VILOCHE BAZAN



Prof(a). Dr(a). ERNESTO JULIÁN GOLDBERG BIRGIN



Prof(a). Dr(a). ELIZABETH WEGNER KARAS

*Aos meus pais
Maria e Mauri
e minha irmã Giselle*

AGRADECIMENTOS

A Deus por ter me amparado durante esta jornada.

À minha família que sempre esteve do meu lado.

Ao Prof. Martínez, ao qual sou imensamente grato pelo auxílio prestado através de suas valiosas sugestões e por sempre estar disposto à discutir sobre os assuntos desta tese. Meu sincero muito obrigado!

Aos Profs. Ernesto, Sandra, Benar e Andreani pelas ótimas sugestões.

Aos Profs. Fermin, Elizabeth, Lucio e Susana, por gentilmente terem aceito o convite para compor a banca examinadora.

Aos amigos presentes intensamente nesta jornada, André “o super”, Wesley “o monstro”, Jair “Aderbal”, Evandro “o ingênuo”. Destes, tenho que dedicar um agradecimento especial ao André, grande amigo, que me ajudou muito neste trabalho.

Aos amigos do IMECC, em especial, Douglas, Cris, Rodrigo, Mael, Cristiano, Giovane, Feodor, Ademir, Ederson, Esdras, Juliana e Janete que proporcionaram boas discussões matemáticas e também momentos de descontração.

Aos meus amigos de Maringá, Waldir, Fran, Kleber, Michelly, André (cunhado), João, Leticia e Carlos, pelo apoio incondicional. Um agradecimento muito especial à uma pessoa especial, Fernanda, por estar do meu lado em momentos difíceis.

À Cidinha, Ednaldo e Tânia por estarem sempre a disposição para solucionar as questões

burocráticas.

Ao CNPq pelo apoio financeiro.

A todos aqueles que eu esqueci de citar aqui mas que, de uma forma ou outra, participaram desta jornada: MUITO OBRIGADO!

RESUMO

Ao resolver problemas de programação não linear usando métodos do tipo Lagrangiano Aumentado, um fenômeno chamado voracidade pode ocorrer. Quando este fenômeno ocorre, o método busca pontos muito ineficazes com valor de função objetivo muito pequeno. Tais fatos ocorrem, em geral, na primeiras iterações e então, o parâmetro de penalidade precisa crescer excessivamente, tornando os subproblemas mal condicionados, prejudicando assim a convergência. Desta forma, o propósito deste trabalho é adicionar restrições de caixas adaptativas (região de confiança) a cada subproblema em cada iteração externa, de modo que, a distância entre dois iterando consecutivos das iterações externas é controlada. O novo método inibe a possibilidade do fenômeno de voracidade. Resultados de convergência, limitação de parâmetro de penalidade e exemplos numéricos são apresentados.

ABSTRACT

When we solve nonlinear programming problems by means of algorithms of kind of Augmented Lagrangian, a phenomenon called greediness may occur. Unconstrained minimizers attract the iterates at early stages of the calculations and, so, the penalty parameter needs to grow excessively, in such a way that ill-conditioning harms the overall convergence. In this sense, the proposal of this work is to add an adaptive artificial box constraint (trust-region) to the subproblem at every outer iteration, in such a way that the distance between consecutive outer iterates is controlled. The new method inhibits the possibility of greediness phenomenon. Convergence proofs and numerical examples are given.

CONTEÚDO

Agradecimentos	v
Resumo	vii
Abstract	viii
Notações Básicas	1
1 Conceitos preliminares	5
1.1 A condição CPLD	5
1.2 O método de Lagrangiano Aumentado	7
2 O método Lagrangiano Aumentado Regularizado	13
2.1 O problema de voracidade	13
2.2 Algoritmo Lagrangiano Aumentado Regularizado	16
2.3 Convergência	22
3 Lagrangiano Aumentado com Região de Confiança	26
3.1 O algoritmo	26
3.2 Convergência e limitação do parâmetro de penalidade	32
3.3 Minimizadores globais	36

4	Implementação e testes	43
4.1	Detalhes de implementação	43
4.2	Testes numéricos	46
4.2.1	Exemplos específicos de voracidade	46
4.2.2	Voracidade e minimização global	49
4.2.3	Comparação dos resultados	51
4.3	Comparação entre Algencan e RAlgencan	56
4.4	Comparação entre Algencan-OTR e RAlgencan	59
5	Conclusões	65
	Bibliografia	67

Notações Básicas

Neste texto usaremos as seguintes notações:

\mathbb{N} : denota o conjunto dos números naturais começando em 0.

\mathbb{R} : denota o conjunto dos números reais.

$\|\cdot\|$: denota a norma Euclidiana.

se $u \in \mathbb{R}^n$, $\|u\|_\infty = \max\{|u_1|, \dots, |u_n|\}$

se $v \in \mathbb{R}^n$, denotamos o vetor v_+ o vetor cujas componentes são $\max\{0, v_1\}, \max\{0, v_2\}, \dots, \max\{0, v_n\}$.

$K \subset_\infty \mathbb{N}$ significa um conjunto infinito de índices que preserva a ordem.

$P_\Omega v$ representa a projeção ortogonal do vetor v no conjunto Ω .

∇ denota derivada de primeira ordem.

∇^2 representa derivada de segunda ordem.

Introdução

No presente trabalho, vamos considerar o seguinte problema de programação não linear:

$$\text{minimizar } f(x) \tag{1}$$

$$\text{s.a. } h(x) = 0,$$

$$g(x) \leq 0,$$

$$x \in \Omega,$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ são funções contínuas com derivadas contínuas. O conjunto Ω é um subconjunto simples do \mathbb{R}^n , por exemplo, uma caixa ($\Omega = \{x \in \mathbb{R}^n; a_i \leq x_i \leq b_i, i = 1, 2, \dots, n\}$).

O método do Lagrangiano Aumentado consiste, basicamente, em resolver o problema de minimizar em Ω a função Lagrangiano Aumentado definida por:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left\{ \sum_{i=1}^m \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x) + \frac{\mu_i}{\rho}) \right]^2 \right\}$$

e decidir se a solução deste problema pode ou não ser aceita como solução do problema original. Caso não seja, devemos aumentar o parâmetro (de penalidade) ρ e atualizar os termos (multiplicadores de Lagrange) λ e μ . Resumidamente, para encontrar a solução de (1) resolvemos subproblemas de minimizar o Lagrangiano Aumentado em Ω .

As idéias para o método citado acima foram introduzidas em [23, 24, 25, 26]. Recentemente, em [1, 27] foram apresentados resultados deste tipo usando a condição CPLD [22].

Uma implementação do método do Lagrangiano Aumentado para resolver problemas na forma (1) é chamada Algencan [28]. Tal implementação, utiliza o método do Gradiente Projetado Espectral [7].

Embora eficiente, esta implementação apresenta algumas dificuldades práticas, a saber, o problema de voracidade [9]. Em linhas gerais, este problema consiste do seguinte: uma vez que o parâmetro de penalidade é definido para ser pequeno nas primeiras iterações, pois, parâmetros grandes podem gerar subproblemas mal condicionados [10, 12, 13] as primeiras iterações tendem a privilegiar a otimalidade sobre a factibilidade e como consequência o método pode divergir.

Em [25], para resolver problemas convexos, Rockafellar introduziu o método de Lagrangiano Aumentado Proximal no qual foi acrescentado um termo regularizador na função Lagrangiano Aumentado.

Usando uma regularização similar, em [9] foi apresentado um método para inibir o problema da voracidade. Este método foi denominado Lagrangiano Aumentado Regularizado. Motivados pela relação natural entre o uso de termo regularizador e o uso de região de confiança, vamos apresentar neste trabalho um método de Lagrangiano Aumentado fazendo uso de estratégias de região de confiança. Além disso, vamos estabelecer relações teóricas entre os métodos, isto é, vamos verificar que tanto o método do Lagrangiano Aumentado Regularizado quanto a nova proposta com região de confiança preservam propriedades de convergência e limitação de parâmetro de penalidade como o método de Lagrangiano Aumentado visto em [1, 27]. Para fins práticos, vamos fazer comparações numéricas maciças entre implementações dos citados métodos.

A tese está organizada da seguinte forma. No capítulo 1 vamos expor alguns conceitos clássicos em otimização bem como os principais resultados envolvendo o método do Lagrangiano Aumentado. Estes resultados serão úteis no capítulo 2 onde vamos expor o problema de voracidade e o método de Lagrangiano Aumentado Regularizado e verificar que, de fato, os resultados teóricos de convergência e limitação de parâmetro de penalidade são preserva-

dos. O capítulo 3 contém, do ponto de vista teórico, os principais resultados do trabalho, ou seja, apresentaremos o método de Lagrangiano Aumentado com Região de Confiança e para este novo método serão estabelecidos teoremas de convergência e limitação do parâmetro de penalidade. Ainda neste capítulo, vamos fornecer resultados que garantem convergência para minimizadores globais para a nova proposta. A confiabilidade da nova proposta é apresentada, no capítulo 4, através de exaustivas comparações numéricas entre as implementações dos algoritmos apresentados na tese. Considerações finais são feitas no capítulo 5.

CAPÍTULO 1

Conceitos preliminares

Neste capítulo vamos expor os conceitos necessários para o bom desenvolvimento deste trabalho. Essencialmente, vamos apresentar o método de Lagrangiano Aumentado como visto em [1, 19, 27]. Dessa forma, vamos apresentar resultados de convergência usando a condição CPLD. Resultados de limitação do parâmetro de penalidade também serão apresentados. Os estudos expostos aqui serão almeçados nos próximos capítulos, ou seja, vamos apresentar resultados equivalentes aos presentes neste capítulo para os algoritmos apresentados nos capítulos 2 e 3.

1.1 A condição CPLD

Antes de apresentar a condição CPLD vamos resgatar algumas definições populares na literatura de otimização. Por simplicidade, vamos nos concentrar no problema:

$$\text{Minimizar } f(x) \text{ sujeita à } x \in D \tag{1.1}$$

onde

$$D = \{x \in \mathbb{R}^n \mid h(x) = 0, g(x) \leq 0\}, \tag{1.2}$$

com

$$h : \mathbb{R}^n \rightarrow \mathbb{R}^m, g : \mathbb{R}^n \rightarrow \mathbb{R}^p, f : \mathbb{R}^n \rightarrow \mathbb{R}$$

funções contínuas em \mathbb{R}^n com derivadas contínuas.

Definição 1.1.1. Dizemos que um ponto x^* é factível (viável ou admissível), quando este satisfaz as restrições de (1.1), isto é, $x^* \in D$. Ainda, dizemos que um ponto x^* é um minimizador global de (1.1) quando, x^* é factível e $f(x^*) \leq f(x)$, $\forall x \in D$. Um ponto x^* é um minimizador local de (1.1) quando existe $\varepsilon > 0$ tal que $f(x^*) \leq f(x)$, para todo $x \in D$ que está a uma distância menor que ε de x^* .

Posto a dificuldade de encontrar minimizadores segundo estas definições vamos estabelecer em sequência condições necessárias de otimalidade.

Definição 1.1.2. Dizemos que um ponto x^* , factível, satisfaz as condições Karush-Kuhn-Tucker quando este ponto satisfaz ao sistema:

$$\nabla f(x^*) + \nabla h(x^*)\lambda + \nabla g(x^*)\mu = 0$$

onde $g_i(x^*) < 0 \Rightarrow \mu_i = 0$.

Neste caso, dizemos que o ponto x^* é um ponto KKT.

Naturalmente, sabemos que KKT não é uma condição necessária de otimalidade. Entretanto, sabemos que se os gradientes das restrições de igualdade junto com restrições de desigualdade ativas em x^* (isto é, as restrições tais que $g_i(x^*) = 0$) formam um conjunto linearmente independente então as condições KKT ocorrem num minimizador local. Tal condição é denominada de *regularidade* e, neste caso, dizemos que x^* é um ponto *regular*. Regularidade é um exemplo de *condição de qualificação*.

Definição 1.1.3. Uma condição de qualificação (CQ) é uma propriedade que quando satisfeita em um minimizador local implica que este é um ponto KKT.

Definição 1.1.4. (CPLD) Seja $I_A(x) = \{i \in \{1, \dots, p\} | g_i(x) = 0\}$. Dizemos que x satisfaz a Condição de Dependência Linear Positiva Constante (CPLD) se a existência de $I_h \subset$

$\{1, \dots, m\}$, $I_g \subset I_A(x)$, $\lambda_i \in \mathbb{R}$ para todo $i \in I_h$, $\mu_i \geq 0$ para todo $i \in I_g$ tais que

$$\sum_{i \in I_h} \lambda_i \nabla h_i(x) + \sum_{i \in I_g} \mu_i \nabla g_i(x) = 0$$

com $\sum_{i \in I_h} |\lambda_i| + \sum_{i \in I_g} \mu_i > 0$, implica que os gradientes

$$\{\nabla h_i(z)\}_{i \in I_h}, \{\nabla g_i(z)\}_{i \in I_g}$$

são linearmente dependentes para todo z numa vizinhança de x .

1.2 O método de Lagrangiano Aumentado

Nesta seção, daremos uma versão do método de Lagrangiano Aumentado citado anteriormente. Para este algoritmo serão apresentados resultados de convergência e limitação de parâmetro de penalidade. Reiteramos que o problema considerado é:

$$\text{minimizar } f(x) \tag{1.3}$$

$$\text{s.a. } h(x) = 0,$$

$$g(x) \leq 0,$$

$$x \in \Omega,$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ são funções contínuas com derivadas contínuas. Estaremos nos referindo a este problema por PNL.

O conjunto Ω é um subconjunto simples do \mathbb{R}^n , por exemplo, uma caixa ($\Omega = \{x \in \mathbb{R}^n; a_i \leq x_i \leq b_i, i = 1, 2, \dots, n\}$). Desta forma, se Ω é conjunto simples estaremos assumindo que o problema de minimizar uma função $L(x)$ em Ω é um problema possível de se resolver na prática. Para efeito geral, estaremos assumindo que:

$$\Omega = \{x \in \mathbb{R}^n; \underline{h}(x) = 0, \underline{g}(x) \leq 0\}$$

onde

$$\underline{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m, \underline{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$$

têm derivadas primeiras contínuas em um subconjunto suficientemente grande de \mathbb{R}^n .

Para exemplificar tais funções, considere o caso em que Ω é uma caixa. Neste caso,

$$\underline{m} = 0, \underline{p} = 2n$$

e

$$\underline{g}(x) = (a_1^{\min} - x_1, \dots, a_n^{\min} - x_n, x_1 - a_1^{\max}, \dots, x_n - a_n^{\max})^T.$$

Como visto anteriormente, dado $\rho > 0$ e aproximações $\lambda_i \in \mathbb{R}$ e $\mu_i \geq 0$ para os multiplicadores, definimos a função Lagrangiano Aumentado por:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left\{ \sum_{i=1}^m \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x) + \frac{\mu_i}{\rho}) \right]^2 \right\}. \quad (1.4)$$

A seguir, temos uma versão mais detalhada do método citado.

Algoritmo 1.

Seja $x^0 \in \mathbb{R}^n$ um ponto inicial arbitrário.

Os parâmetros para a execução do algoritmo são:

$$\tau \in [0, 1), \eta > 1, -\infty < \lambda_{\min} < \lambda_{\max} < +\infty,$$

$$\mu_{\max} > 0, \rho_1 > 0,$$

$$\bar{\lambda}_i^1 \in [\lambda_{\min}, \lambda_{\max}], \quad \forall i = 1, \dots, m,$$

$$\bar{\mu}_i^1 \in [0, \mu_{\max}], \quad \forall i = 1, \dots, p,$$

$\{\varepsilon_k\} \subset \mathbb{R}_+$ uma sequência de parâmetros de tolerância tal que,

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0. \quad (1.5)$$

Passo 1 Inicialização

$k \leftarrow 1$. Definir $V^0 = g(x^0)_+$;

Passo 2 Resolução do subproblema

Calcular (se possível) $x^k \in \mathbb{R}^n$ tal que existam $v^k \in \mathbb{R}^m, u^k \in \mathbb{R}^p$ satisfazendo

$$\left\| \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \sum_{i=1}^m v_i^k \nabla h_i(x^k) + \sum_{i=1}^p u_i^k \nabla \underline{g}_i(x^k) \right\| \leq \varepsilon_k, \quad (1.6)$$

$$u_i^k \geq 0, \underline{g}_i(x^k) < \varepsilon_k \text{ para todo } i = 1, \dots, \underline{p}, \quad (1.7)$$

$$\underline{g}_i(x^k) < -\varepsilon_k \Rightarrow u_i^k = 0 \text{ para todo } i = 1, \dots, \underline{p}, \quad (1.8)$$

$$\|\underline{h}(x^k)\| \leq \varepsilon_k. \quad (1.9)$$

Se não é possível encontrar x^k cumprindo (1.6) - (1.9), parar a execução do algoritmo.

Passo 3 *Estimar multiplicadores*

Para todo $i = 1, \dots, m$, calcular

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k) \quad (1.10)$$

e

$$\bar{\lambda}_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}]. \quad (1.11)$$

Geralmente, $\bar{\lambda}_i^{k+1}$ é a projeção de λ_i^{k+1} no intervalo $[\lambda_{\min}, \lambda_{\max}]$.

Para todo $i = 1, \dots, p$, calcular

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\} \quad (1.12)$$

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}$$

$$\bar{\mu}_i^{k+1} \in [0, \mu_{\max}]. \quad (1.13)$$

(Geralmente, $\bar{\mu}_i^{k+1} = \min\{\mu_i^{k+1}, \bar{\mu}_{\max}\}$).

Passo 4 *Atualizar o parâmetro de penalidade*

Se

$$R^k = \max\{\|h(x^k)\|_{\infty}, \|V^k\|_{\infty}\} \leq \tau \max\{\|h(x^{k-1})\|_{\infty}, \|V^{k-1}\|_{\infty}\},$$

definir

$$\rho_{k+1} = \rho_k.$$

Caso contrário, definir

$$\rho_{k+1} = \eta \rho_k.$$

Passo 5 *Começar uma nova iteração*

Atualizar $k \leftarrow k + 1$. Voltar ao Passo 2.

Note que, da forma que foi definido o método não apresenta critério de parada. Naturalmente, na prática adotamos critérios para interromper o algoritmo em um número finito de passos.

Agora, vamos supor que sempre podemos encontrar x^k satisfazendo (1.6) -(1.9). Vamos supor também que, $\Omega \neq \emptyset$ e existe $\varepsilon > 0$ tal que o conjunto

$$\{x \in \mathbb{R}^n; \underline{g}_i(x) \leq \varepsilon, \|\underline{h}(x)\| \leq \varepsilon\}$$

é limitado. Note que, esta última condição pode ser forçada acrescentando restrições artificiais em Ω . Desta forma temos que a sequência gerada pelo algoritmo possui um ponto limite. Os resultados que vamos apresentar agora estão relacionados a convergência do Algoritmo 1 para pontos KKT. Como suas demonstrações podem ser encontradas em [27, 1] as mesmas serão omitidas.

O primeiro resultado que vamos enunciar nos fornece características sobre a admissibilidade dos pontos limites das sequências geradas pelo método citado.

Teorema 1.2.1. *Seja $\{x^k\}$ uma sequência infinita gerada pelo Algoritmo 1. Seja x^* um ponto limite de x^k . Então, se a sequência de parâmetros de penalidade ρ_k é limitada, x^* é admissível. Do contrário, pelo menos uma das seguintes possibilidades acontece:*

(i) *O ponto x^* é um ponto KKT do problema*

$$\text{Minimizar } [\sum_{i=1}^m h_i(x)^2 + \sum_{i=1}^p \max\{0, g_i(x)\}^2] \text{ sujeita a } x \in \Omega.$$

(ii) *x^* não satisfaz a condição CPLD associada a Ω .*

Essencialmente, o teorema anterior fornece que, se x^* não é admissível então, provavelmente, este será um minimizador da inadmissibilidade.

No resultado a seguir, veremos que, se uma restrição de desigualdade se verifica estritamente no ponto limite, então, a estimativa do multiplicador associado se anula, para k suficientemente grande.

Teorema 1.2.2. *Seja $\{x^k\}_{k \in \mathbb{N}}$ uma sequência gerada pelo Algoritmo 1. Suponhamos que x^* é um ponto limite (não necessariamente admissível) desta sequência e $K \subset \mathbb{N}$ é tal que*

$\lim_{k \in K} x^k = x$. Então, para $k \in K$ suficientemente grande,

$$g_i(x^*) < 0 \Rightarrow \mu_i^{k+1} = 0 \text{ e } \underline{g}_i(x^*) < 0 \Rightarrow u_i^k = 0 \quad (1.14)$$

Finalizando o estudo de convergência para pontos KKT, temos que sob CPLD, os pontos limites da sequência gerada pelo Algoritmo 1 são pontos KKT do problema original. Este fato é expresso pelo seguinte teorema.

Teorema 1.2.3. *Seja $\{x^k\}_{k \in \mathbb{N}}$ uma sequência gerada pelo Algoritmo 1. Suponhamos que x^* é um ponto limite admissível que satisfaz CPLD com respeito a todas as restrições do problema original. Então, x^* é ponto KKT do problema original.*

Uma observação importante é que, quando o parâmetro de penalidade é grande, os subproblemas do método de Lagrangiano Aumentado são difíceis de resolver. A razão para isto é que, ao ser multiplicado por um número muito grande, o termo penalizado domina a função objetivo do problema. Assim, $f(x)$ funciona como uma pequena perturbação do segundo termo do problema e o algoritmo que resolve o subproblema desprezará tal perturbação. Em ponto flutuante, a soma de um número muito grande com um número relativamente pequeno é o número muito grande. Portanto, quando se minimiza uma função do tipo Lagrangiano Aumentado com um parâmetro de penalidade grande, a tendência é que, na maioria dos casos, o método interno (que resolve o subproblema) se comporte como se o termo $f(x)$ não existisse.

Dessa forma, é muito importante que os métodos de Lagrangiano Aumentado tenham a propriedade de que o parâmetro de penalidade não cresça indefinidamente. Ou seja, desejamos que o teste do passo 4 tenha resultado positivo para todo k suficientemente grande.

Assim, vamos apresentar um resultado que garante que o parâmetro de penalidade fica limitado sob condições que são verificadas na prática.

As hipóteses para o teorema que vamos apresentar são as seguintes:

1. A sequência $\{x^k\}$ gerada pelo Algoritmo 1 converge à x^* .
2. O ponto x^* é admissível, ou seja, $h(x^*) = 0$ e $g(x^*) \leq 0$.

3. O ponto x^* é regular. Isto implica que as condições KKT se verificam em x^* . Portanto, existem multiplicadores de Lagrange $\lambda^* \in \mathbb{R}^m$ e $\mu \in \mathbb{R}_+^p$ associados a estas restrições.
4. O multiplicador λ_i^* pertence a $(\lambda_{min}, \lambda_{max})$ para todo $i = 1, \dots, m$. Analogamente, cada multiplicador μ_i^* pertence a $[0, \mu_{max})$ para todo $i = 1, \dots, p$.
5. As funções f , h e g admitem derivadas segundas contínuas em uma vizinhança de x^* . (Os operadores ∇ e ∇^2 denotarão sempre derivadas em relação a x).
6. Para cada $i = 1, \dots, p$ tal que $g_i(x^*) = 0$ temos que $\mu_i > 0$. Esta condição se denomina *Complementariedade estrita*.
7. Seja $A \in \mathbb{R}^{(m+q) \times n}$ a matriz cujas linhas são os gradientes das restrições de igualdade seguidas dos gradientes das q restrições ativas ($g_i(x^*) = 0$) em x^* . Seja $Z \in \mathbb{R}^{n \times (n-m-q)}$ uma matriz cujas colunas formam uma base do núcleo de A . Então, $Z^T \nabla^2 L_0(x^*, \lambda^*, \mu^*) Z$ é definida positiva.

Observação 1.2.4. *A complementariedade estrita junto com a última hipótese definida acima fornecem uma condição suficiente para o minimizador local do problema PNL considerado.*

Teorema 1.2.5. *Suponhamos, além das hipóteses gerais desta seção, que exista uma sequência $\{\eta_k\}$ que converge a zero, tal que*

$$\varepsilon_k \leq \eta_k \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}$$

para todo $k \in \mathbb{N}$. Então, a sequência de parâmetros de penalidade $\{\rho_k\}$ está limitada.

Demonstração: Veja [1]

□

CAPÍTULO 2

O método Lagrangiano Aumentado Regularizado

Neste capítulo estamos, novamente, considerando o problema PNL definido no capítulo 1. Uma implementação para o método visto no capítulo anterior é denominada Algencan [28]. Contudo, métodos deste tipo (e portanto esta implementação) apresentam uma dificuldade prática, isto é, iterandos podem ser atraídos para regiões inactiváveis pois primam mais pela otimalidade do que pela factibilidade. Tal fenómeno é chamado *voracidade* e será melhor detalhado a seguir. Os resultados que serão apresentados neste capítulo estão baseados em [9]. De maneira sucinta, a idéia do método que vamos expor agora é introduzir um termo regularizador no método de Lagrangiano Aumentado definido no capítulo 1.

2.1 O problema de voracidade

Suponha que o problema PNL tenha uma caixa artificial (isto é, impomos limitante inferior e superior para as restrições de caixa) e que a função f tenha um forte decréscimo. Um fato que pode acontecer nas primeiras iterações de problemas com esta configuração é que o

parâmetro de penalidade inicial não esteja penalizando suficientemente a inadmissibilidade, ou seja, o método encontra a solução do subproblema priorizando o decréscimo da função objetivo, mesmo perdendo factibilidade. Geralmente, o iterando vai para o bordo da caixa artificial. O problema que surge então é o seguinte: o método do Lagrangiano Aumentado começa a penalizar na expectativa de retirar os iterandos do bordo, entretanto, os problemas ficam cada vez mais complicados de serem resolvidos e pode acontecer que o parâmetro de penalidade necessário para retirar os pontos da infactibilidade seja muito grande e portanto o método não resolve o problema. Aos problemas que apresentam este fenômeno damos o nome de *voracidade*. Para detalhar um pouco mais este comportamento considere o seguinte exemplo:

$$\begin{array}{ll} \min & -x^6 \\ \text{s.a} & x - 1 = 0 \end{array}$$

Note que este problema não possui restrições de caixa. Para efeito de exemplo, vamos considerar a implementação Algencan. Algencan resolve problemas na forma:

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & h(x) = 0, g(x) \leq 0 \\ & l \leq x \leq u. \end{array}$$

Assim, para colocar este problema na forma padrão de Algencan, este deve receber uma caixa artificial colocando como limitantes números reais de grande valor absoluto, digamos, $l = -10^8$ e $u = 10^8$. Observe na Figura 2.1 que a função objetivo tem forte decréscimo para valores maiores que $x = 1$ (que é o minimizador e restrição deste problema).

Note que, se o parâmetro de penalidade não está penalizando suficientemente a inadmissibilidade o iterando obtido ao resolver o subproblema será atraído pelo forte decréscimo que função objetivo apresenta para valores maiores que 1. De fato, ao resolver o primeiro subproblema deste exemplo Algencan considera $\mu = 0$, $\lambda = 0$ e $\rho_0 = 10$. Dessa forma, o

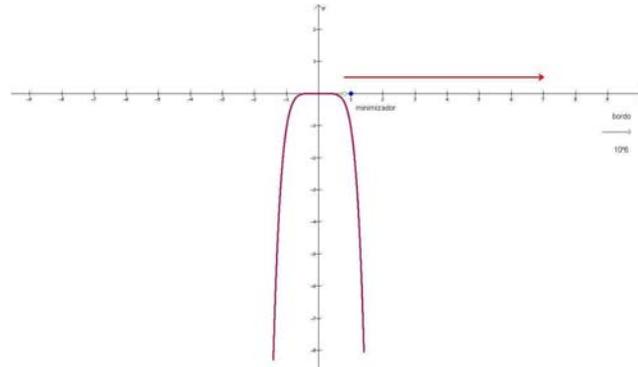


Figura 2.1: Exemplo de função com voracidade.

primeiro iterando é o minimizador em $[l, u]$ de $L_{10}(x, 0, 0)$. Observe na Figura 2.2 que mesmo ao perder factibilidade o decréscimo de $L_{10}(x, 0, 0)$ é tão grande que o método obtém um ponto bastante infactível (no bordo da caixa artificial) atraído por este decréscimo. Assim, o ponto obtido ao resolver este subproblema está muito longe da região factível, logo o método começa a penalizar cada vez mais para retirar este ponto da infactibilidade e ao fazer isto os subproblemas ficam cada vez mais complicados. E o método pode divergir. Além disso, os multiplicadores ficam comprometidos.

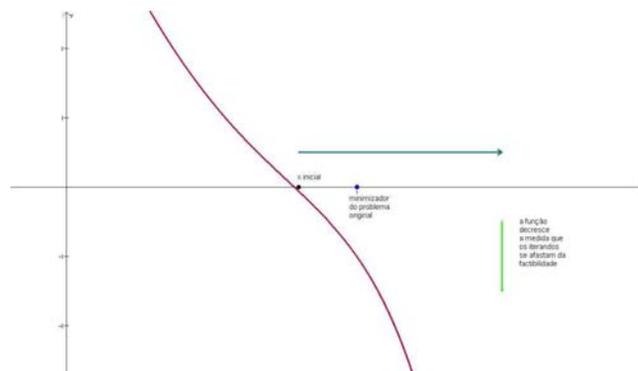


Figura 2.2: Comportamento de $L_{10}(x, 0, 0)$.

2.2 Algoritmo Lagrangiano Aumentado Regularizado

Nesta seção, vamos expor o algoritmo formal do método Lagrangiano Aumentado Regularizado. Embora a principal motivação para este algoritmo seja contornar a voracidade, outros aspectos são considerados:

- Aproveitar melhor a aproximação inicial x^0 ;
- Evitar que o método se distancie muito da região factível nas suas primeiras iterações;
- Melhorar o condicionamento do problema.

Visando estes objetivos, vamos somar o termo regularizador

$$\gamma \frac{\|x - \bar{x}\|_2^2}{2} \quad (2.1)$$

ao Lagrangiano Aumentado, onde $\gamma \geq 0$ e \bar{x} é o *ponto de referência*. Este ponto é o melhor iterando até o momento em relação à factibilidade-complementariedade. Assim definimos a função *Lagrangiano Aumentado Regularizado* como:

$$L_{\rho, \gamma, \bar{x}}(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left\{ \sum_{i=1}^m \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x) + \frac{\mu_i}{\rho}) \right]^2 \right\} + \gamma \frac{\|x - \bar{x}\|_2^2}{2}. \quad (2.2)$$

A seguir apresentamos o algoritmo do Lagrangiano Aumentado regularizado (sem critério de parada):

Algoritmo 2

Seja $x^0 \in \mathbb{R}^n$ um ponto inicial arbitrário.

Os parâmetros para a execução do algoritmo são:

$$\tau \in [0, 1), \gamma_1 \geq 0, \gamma_{\max} > 0, R_{tol} > 0, \eta > 1, \beta > 0$$

$$-\infty < \lambda_{\min} < \lambda_{\max} < +\infty,$$

$$\mu_{\max} > 0,$$

$$\rho_1 > 0,$$

$$\bar{\lambda}_i^1 \in [\lambda_{\min}, \lambda_{\max}] \forall i = 1, \dots, m,$$

$$\bar{\mu}_i^1 \in [0, \mu_{\max}], \forall i = 1, \dots, p,$$

$\{\varepsilon_k\} \subset \mathbb{R}_+$ uma sequência de parâmetros de tolerância tal que

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0. \quad (2.3)$$

Passo 1 Inicialização

$$k \leftarrow 1.$$

$$\text{Definir } V^0 = g(x^0)_+;$$

$$\bar{x}^0 = x^0;$$

$$\bar{R} = R^0 = \max\{\|h(x^0)\|_\infty, \|V^0\|_\infty\};$$

Passo 2 Resolução do subproblema

Calcular (se possível) $x^k \in \mathbb{R}^n$ tal que existam $v^k \in \mathbb{R}^m$, $u^k \in \mathbb{R}^p$ satisfazendo

$$\|\nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \gamma_k(x^k - \bar{x}^{k-1}) + \sum_{i=1}^m v_i^k \nabla h_i(x^k) + \sum_{i=1}^p u_i^k \nabla g_i(x^k)\| \leq \varepsilon_k, \quad (2.4)$$

$$u_i^k \geq 0, g_i(x^k) < \varepsilon_k \text{ para todo } i = 1, \dots, p, \quad (2.5)$$

$$g_i(x^k) < -\varepsilon_k \Rightarrow u_i^k = 0 \text{ para todo } i = 1, \dots, p, \quad (2.6)$$

$$\|h(x^k)\| \leq \varepsilon_k. \quad (2.7)$$

Se não é possível encontrar x^k cumprindo (2.4) - (2.7), parar a execução do algoritmo.

Passo 3 Estimar multiplicadores

Para todo $i = 1, \dots, m$, calcular

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k). \quad (2.8)$$

Para todo $i = 1, \dots, p$, calcular

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\} \quad (2.9)$$

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}.$$

Passo 4 *Atualizar o parâmetro de penalidade*

$$R_k = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}.$$

Se

$$R_k \leq \tau R_{k-1} \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\},$$

definir

$$\rho_{k+1} = \rho_k.$$

Caso contrário, definir

$$\rho_{k+1} = \eta \rho_k.$$

Passo 5 *Atualizar o parâmetro de regularização e o ponto de referência*

Se

$$R_k = \min\{\max\{R_{tol}, R_0\}, \dots, R_k\}, \quad (2.10)$$

definimos $\bar{x}^k = x^k$.

Caso contrário, definimos $\bar{x}^k = \bar{x}^{k-1}$.

Para todo $i = 1, \dots, m$ computar:

$$\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}]. \quad (2.11)$$

Para todo $i = 1, \dots, p$ computar:

$$\bar{\mu}_i^{k+1} \in [0, \mu_{max}]. \quad (2.12)$$

Escolhemos γ_{k+1} de maneira que

$$\gamma_{k+1} \leq \min\{\gamma_k, \beta R_k\}. \quad (2.13)$$

Passo 6 *Começar uma nova iteração*

Atualizar $k \leftarrow k + 1$. Voltar ao Passo 2.

Observação 2.2.1. *As condições (2.4)-(2.7) dizem que x^k é um ponto KKT aproximado para o subproblema*

$$\begin{aligned} \min \quad & L_{\rho_k, \gamma_k, \bar{x}^{k-1}}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \\ \text{s.a} \quad & \underline{h}(x) = 0 \\ & \underline{g}(x) \leq 0 \end{aligned}$$

Portanto, a cada iteração externa, minimizamos (aproximadamente) o Lagrangiano Aumentado Regularizado no conjunto Ω .

Observação 2.2.2. *Os passos 3 e 4 são como no Algoritmo 1. No passo 5, o ponto de referência e o parâmetro de regularização são atualizados. A idéia é que o ponto de referência seja o melhor iterando até o momento em relação à factibilidade-complementariedade que é medida por R_k . A condição (2.13) impõe que a sequência do parâmetro de regularização seja monótona decrescente, e convirja a zero se alguma subsequência de $\{R_k\}_{k=0}^\infty$ convergir a zero. Assim o efeito da regularização tende a desaparecer quando o algoritmo se comporta bem em termos da obtenção de pontos factíveis.*

Observação 2.2.3. *Se atualizamos o ponto de referência, os multiplicadores $\bar{\lambda}$ e $\bar{\mu}$ em (2.11) e (2.12) serão calculados da seguinte forma:*

$$\bar{\lambda}_i^{k+1} = \max\{\lambda_{\min}, \min\{\lambda_{\max}, \lambda_i^{k+1}\}\},$$

para $i = 1, \dots, m$ e

$$\bar{\mu}_i^{k+1} = \min\{\mu_{\max}, \mu_i^{k+1}\},$$

para $i = 1, \dots, p$.

Quando não atualizamos o ponto de referência, definimos $\bar{\lambda}^{k+1} = \bar{\lambda}^k$ e $\bar{\mu}^{k+1} = \bar{\mu}^k$.

Para provar a convergência global do Algoritmo 2 usamos o seguinte fato: se $K \subset \mathbb{N}$,

$$\lim_{k \in K} R_k = 0 \Rightarrow \lim_{k \rightarrow \infty} \gamma_k = 0. \quad (2.14)$$

Esta propriedade é verdadeira pois $\lim_{k \in K} R_k = 0$ implica, por (2.13), que:

$$\lim_{k \in K} \gamma_{k+1} = 0.$$

Agora, como $\gamma_{k+1} \leq \gamma_k$ para todo k , portanto $\lim_{k \rightarrow \infty} \gamma_k = 0$. Isto significa que a condição

$$\gamma_{k+1} \leq \gamma_k, \quad \forall k, \quad (2.15)$$

imposta em (2.13) é necessária para provar o teorema de convergência. Contudo, a condição (2.15) pode não ser interessante no caso de desejarmos aumentar o parâmetro de regularização se a factibilidade se deteriorar. Noutras palavras, são razoáveis estratégias de regularização que exigem somente $\gamma_{k+1} \leq \beta R_k$, mas não que $\gamma_{k+1} \leq \gamma_k$.

Agora, vamos ver que, com uma certa modificação do Algoritmo 2, a condição de monotonicidade sobre γ_k pode ser eliminada.

Algoritmo 3

Este algoritmo coincide com o Algoritmo 2, exceto pelos seguintes fatos:

- Para a solução do subproblema, emprega um algoritmo que usa \bar{x}^{k-1} como ponto inicial, e então garantimos que

$$L_{\rho_k, \gamma_k, \bar{x}^{k-1}}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_k, \gamma_k, \bar{x}^{k-1}}(\bar{x}^{k-1}, \bar{\lambda}^k, \bar{\mu}^k) \quad (2.16)$$

- A condição (2.13) é reescrita por

$$\gamma_{k+1} \leq \beta R_k. \quad (2.17)$$

A condição (2.16) é bastante natural para um algoritmo de minimização monótona que preserva viabilidade das restrições para a resolução dos subproblemas.

O que permite relaxar (2.17) é o resultado seguinte.

Lema 2.2.4. *Seja $\{x^k\}$ uma sequência gerada pelo Algoritmo 3 e suponha que existe um conjunto infinito de índices K tal que*

$$\lim_{k \in K} R_k = 0.$$

Então,

$$\lim_{k \rightarrow \infty} R_k = 0. \quad (2.18)$$

Demonstração: Se a sequência $\{\rho_k\}$ é limitada, então $R_k \leq \tau R_{k-1}$ para todo k suficientemente grande, a tese esta provada.

Assumimos, agora, que $\lim_{k \rightarrow \infty} \rho_k = \infty$.

Por (2.16), temos para todo $k \in \mathbb{N}$,

$$\begin{aligned} f(x^k) + \frac{\rho_k}{2} \left\{ \sum_{i=1}^m \left[h_i(x_k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k} \right) \right]^2 \right\} + \frac{\gamma_k}{2} \|x^k - \bar{x}^{k-1}\|^2 \\ \leq f(\bar{x}^{k-1}) + \frac{\rho_k}{2} \left\{ \sum_{i=1}^m \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i^k}{\rho_k} \right) \right]^2 \right\}. \end{aligned}$$

Dividindo por ρ_k obtemos:

$$\begin{aligned} \frac{1}{\rho_k} f(x^k) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(x_k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k} \right) \right]^2 \right\} + \frac{\gamma_k}{2} \|x^k - \bar{x}^{k-1}\|^2 \\ \leq \frac{1}{\rho_k} f(\bar{x}^{k-1}) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i^k}{\rho_k} \right) \right]^2 \right\}. \quad (2.19) \end{aligned}$$

Pela definição de \bar{x}^{k-1} podemos escrever:

$$\{\bar{x}^s, \bar{x}^{s+1}, \bar{x}^{s+2}, \dots\} = \{x^{k_0}, x^{k_1}, x^{k_2}, \dots\}$$

onde $k_0 \leq k_1 \leq k_2 \leq \dots$. No entanto, como $\lim_{k \in K} R_k = 0$, temos que

$$\lim_{j \rightarrow \infty} R_{k_j} = 0. \quad (2.20)$$

Claramente, (2.20) implica que

$$\lim_{j \rightarrow \infty} \|h(x^{k_j})\|^2 + \sum_{i=1}^p \max\{0, g_i(x^{k_j})\}^2 = 0.$$

Portanto,

$$\lim_{k \rightarrow \infty} \|h(\bar{x}^{k-1})\|^2 + \sum_{i=1}^p \max\{0, g_i(\bar{x}^{k-1})\}^2 = 0.$$

Dessa forma, o lado direito da desigualdade (2.19) tende a zero quando k tende ao infinito.

Assim,

$$\lim_{k \rightarrow \infty} \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(x_k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k} \right) \right]^2 \right\} = 0. \quad (2.21)$$

Como $\rho_k \rightarrow \infty$ e $\bar{\mu}^k$ é limitado, (2.21) implica que

$$\lim_{k \rightarrow \infty} \|h(x^k)\| = 0 \quad (2.22)$$

e

$$\lim_{k \rightarrow \infty} \max\{0, g_i(x^k)\} = 0 \quad \forall i = 1, \dots, p.$$

Então,

$$\lim_{k \rightarrow \infty} V_i^k = 0 \quad \forall i = 1, \dots, p. \quad (2.23)$$

Por (2.22) e (2.23) obtemos (2.18). \square

Lema 2.2.5. *Seja $\{x^k\}$ uma sequência gerada pelo Algoritmo 3 e suponha que existe um conjunto infinito de índices K tal que*

$$\lim_{k \in K} R_k = 0.$$

Então,

$$\lim_{k \rightarrow \infty} \gamma_k = 0.$$

Demonstração: O resultado segue de (2.17) e do Lema 2.2.4. \square

2.3 Convergência

Nesta seção mostraremos a convergência global dos Algoritmos 2 e 3. Essencialmente, mostraremos que os resultados de convergência para o método do Lagrangiano Aumentado vistos no capítulo 1 continuam válidos para os algoritmos apresentados neste capítulo.

Vamos supor que os Algoritmos 2 e 3 não param no Passo 2 e que a sequência gerada $\{x^k\}$ é limitada. Assim, a sequência possui um ponto limite. Uma condição para que isto ocorra é que $\Omega \neq \emptyset$ e existe $\varepsilon > 0$ tal que o conjunto

$$\{x \in \mathbb{R}^n \mid g_i(x) \leq \varepsilon, \|h(x)\| \leq \varepsilon\}$$

seja limitado. Esta condição pode ser forçada acrescentando restrições artificiais no conjunto Ω .

O primeiro resultado é interessante e útil, ele independe da CPLD. Este resultado nos diz que, se uma restrição de desigualdade se verifica estritamente no ponto limite, então a estimativa do multiplicador associado se anula, para k suficientemente grande.

Proposição 2.3.1. *Seja $\{x^k\}_{k \in \mathbb{N}}$ uma sequência gerada pelo Algoritmo 2 ou 3. Suponhamos que x^* é um ponto limite desta sequência e $K \subset \mathbb{N}$ é tal que $\lim_{k \in K} x^k = x^*$. Então, para $k \in K$ suficientemente grande,*

$$g_i(x^*) < 0 \Rightarrow \mu_i^{k+1} = 0 \text{ e } \underline{g}_i(x^*) < 0 \Rightarrow u_i^k = 0. \quad (2.24)$$

Demonstração: Segue do Teorema 1.2.2. □

Lema 2.3.2. *Seja $\{x^k\}$ uma sequência limitada gerada pelo Algoritmo 2 ou 3. Então,*

$$\lim_{k \rightarrow \infty} \frac{\gamma_k}{\rho_k} \|x^k - \bar{x}^{k-1}\| = 0 \quad (2.25)$$

Demonstração: A sequência $\{\bar{x}^k\}$ é limitada por definição. Uma vez que $\{\bar{x}^k\} \subset \{x^k\}$ temos $\|x^k - \bar{x}^{k-1}\|$ é limitado.

Do Passo 5 dos Algoritmos 2 e 3 segue que a sequência $\{\gamma_k\}$ é limitada.

Temos duas possibilidades a considerar:

1. A sequência $\{\rho_k\}$ é limitada.

Então o parâmetro pára de ser incrementado para k suficientemente grande. Portanto, segue do passo 4 do Algoritmo 2 e 3 que, $\lim_{k \rightarrow \infty} R_k = 0$. Então, como $\gamma_{k+1} \leq \beta R_k$, segue que $\lim_{k \rightarrow \infty} \gamma_k = 0$. Além disso, como $\|x^k - \bar{x}^{k-1}\|$ é limitado, obtemos (2.25).

2. A sequência $\{\rho_k\}$ não é limitada.

As limitações de $\{\gamma_k\}$ e $\|x^k - \bar{x}^{k-1}\|$ implicam que (2.25) ocorre. □

Teorema 2.3.3. *Seja $\{x^k\}$ uma sequência limitada gerada pelo Algoritmo 2 ou 3, e seja x^* um ponto limite desta sequência. Então:*

1. *Se a sequência de parâmetros de penalidade ρ_k é limitada, x^* é admissível. Em caso contrário, pelo menos uma das seguintes possibilidades acontece:*

(i) O ponto x^* é um ponto KKT do problema

$$\begin{aligned} \min \quad & \left[\sum_{i=1}^m h_i(x)^2 + \sum_{i=1}^p \max\{0, g_i(x)\}^2 \right] \\ \text{s.a} \quad & x \in \Omega \end{aligned}$$

(ii) x^* não satisfaz a condição CPLD associada às restrições do conjunto Ω .

2. Se x^* é um ponto limite admissível que satisfaz CPLD com respeito a todas as restrições do problema PNL. Então, x^* é ponto KKT do problema.

Demonstração: Seja $K \subset \mathbb{N}$ tal que

$$\lim_{k \in K} x^k = x^*.$$

A primeira parte do teorema é obtida seguindo a mesma sequência de argumentos usada no Teorema 1.2.1 que pode ser encontrada em [1], com a diferença que teremos que utilizar o Lema 2.3.2 e o fato de que a sequência $\{x^k\}$ é limitada.

Vamos provar a segunda parte do teorema. Segue de (2.4) - (2.7) e da Proposição 2.3.1, que para $k \in K$ suficientemente grande temos que:

$$\begin{aligned} \nabla f(x^k) + \sum_{i=1}^m (\lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i|g_i(x^*)=0} \mu_i^{k+1} \nabla g_i(x^k) + \gamma_k(x^k - \bar{x}^{k-1}) \\ + \sum_{i=1}^m v_i^k \nabla \underline{h}_i(x^k) + \sum_{j|\underline{g}_j(x^k)=0} u_j^k \nabla \underline{g}_j(x^k) = \underline{\delta}^k, \end{aligned}$$

onde $\mu^{k+1} \in \mathbb{R}_+^p$, $w^k \in \mathbb{R}_+^2$ e $\|\underline{\delta}^k\| \rightarrow 0$.

Como x^* é factível, temos que $\lim_{k \in K} R_k = 0$. Assim por (2.13) para o Algoritmo 2 temos:

$$\lim_{k \rightarrow \infty} \gamma_k = 0. \quad (2.26)$$

No caso do Algoritmo 3, (2.26) segue do Lema 2.2.5. Portanto, podemos definir

$$\delta^k = \underline{\delta}^k - \gamma_k(x^k - \bar{x}^{k-1}),$$

e então, para k suficientemente grande,

$$\begin{aligned} \nabla f(x^k) + \sum_{i=1}^m (\lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i|g_i(x^*)=0} \mu_i^{k+1} \nabla g_i(x^k) \\ + \sum_{i=1}^m v_i^k \nabla \underline{h}_i(x^k) + \sum_{j|\underline{g}_j(x^k)=0} u_j^k \nabla \underline{g}_j(x^k) = \delta^k, \end{aligned}$$

onde $\mu^{k+1} \in \mathbb{R}_+^p$, $w^k \in \mathbb{R}_+^p$ e $\|\delta^k\| \rightarrow 0$.

A partir deste ponto, seguem os mesmos argumentos do Teorema 1.2.3 (veja [1] para mais detalhes), e obtemos o resultado. \square

CAPÍTULO 3

Lagrangiano Aumentado com Região de Confiança

No capítulo anterior, apresentamos um estudo de um método de Lagrangiano Aumentado com o acréscimo do termo regularizador. Este estudo foi composto pela exposição de um algoritmo e a partir deste algoritmo apresentamos teoremas que garantem convergência global. O termo regularizador citado anteriormente motiva, naturalmente, ao estudo de região de confiança. Dessa forma, vamos apresentar neste capítulo um método que, em vez de lidar com o termo regularizador para contornar a voracidade, utiliza estratégias de região de confiança.

3.1 O algoritmo

A idéia do nosso novo método é incorporar restrições de caixa artificiais adaptativas para os subproblemas de forma que a distância entre iterandos (externos, isto é, as soluções dos subproblemas) consecutivos é controlada. Estas caixas adaptativas são as regiões de confiança, isto é, esperamos que nestas caixas o problema de voracidade possa ser controlado.

Caso isto não ocorra, diminuimos o tamanho da caixa até que a patologia seja contornada. Quando não ocorre voracidade, a caixa cresce pois assim damos liberdade para o método encontrar o minimizador do problema original sem as restrições adaptativas.

Lembremos que o problema considerado é:

$$\text{Minimizar } f(x) \text{ sujeito a } h(x) = 0, g(x) \leq 0, x \in \Omega.$$

Lembremos também que podemos escrever $\Omega = \{x \in \mathbb{R}^n | \underline{h}(x) = 0, \underline{g}(x) \leq 0\}$ e assumir que as funções $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $\underline{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\underline{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ têm derivadas de primeira ordem contínuas no \mathbb{R}^n . Assim, Ω é um conjunto fechado. A seguir, vamos definir o primeiro algoritmo da nova proposta.

Algoritmo 4

Os parâmetros que definem o algoritmo são: $\tau \in [0, 1)$, $\eta > 1$, $\lambda_{min} < \lambda_{max}$, $\mu_{max} > 0$, $\beta_1 > 0$, $\beta_2 > 0$, $R_{tol} > 0$. Na primeira iteração externa usamos um parâmetro de penalidade $\rho_1 > 0$ e estimativas $\bar{\lambda}^1 \in \mathbb{R}^m$ e $\bar{\mu}^1 \in \mathbb{R}^p$ para os multiplicadores de Lagrange tais que

$$\bar{\lambda}_i^1 \in [\lambda_{min}, \lambda_{max}] \forall i = 1, \dots, m \text{ e } \bar{\mu}_i^1 \in [0, \mu_{max}] \forall i = 1, \dots, p.$$

Assumimos que $x^0 \in \mathbb{R}^n$ é um ponto inicial arbitrário que coincide com o *ponto de referência* inicial \bar{x}^0 e $\Delta_1 > 0$ é um raio arbitrário para a região de confiança.

Para cada $k \in \{1, 2, \dots\}$ definimos

$$B_k = \{x \in \mathbb{R}^n | \|x - \bar{x}^{k-1}\|_\infty \leq \Delta_k\}.$$

Finalmente, $\{\varepsilon_k\}$ é uma sequência de números positivos que satisfaz

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0.$$

Passo 1. Inicialização.

$$k \leftarrow 1.$$

Passo 2. Resolver o subproblema.

Calcular $x^k \in B^k \cap \Omega$ tal que existem $v^k \in \mathbb{R}^m$, $w^k \in \mathbb{R}^p$ satisfazendo

$$\|P_{B^k \cap \Omega}[x^k - (\nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \mu^k) + \sum_{i=1}^m v_i^k \nabla h_i(x^k) + \sum_{i=1}^p w_i^k \nabla g(x^k))] - x^k\| \leq \varepsilon_k, \quad (3.1)$$

$$w^k \geq 0, \underline{g}(x^k) \leq \varepsilon_k \quad (3.2)$$

$$\underline{g}_i(x^k) < -\varepsilon_k \Rightarrow w_i^k = 0 \text{ para todo } i = 1, \dots, p, \quad (3.3)$$

$$\|\underline{h}(x^k)\| \leq \varepsilon_k \quad (3.4)$$

e

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_k}(\bar{x}^k, \bar{\lambda}^k, \bar{\mu}^k). \quad (3.5)$$

Passo 3. Calcular a factibilidade e a complementariedade. Para cada $i = 1, \dots, p$ calculamos

$$V_i^k = \max\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\},$$

e definimos

$$R_k = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}.$$

Passo 4. Atualizar os multiplicadores de Lagrange e o ponto de referência.

Se $R_k \neq \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$, definimos $\bar{x}^k = \bar{x}^{k-1}$, $\lambda^{k+1} = \lambda^k$, $\bar{\lambda}^{k+1} = \bar{\lambda}^k$, $\mu^{k+1} = \mu^k$, $\bar{\mu}^{k+1} = \bar{\mu}^k$. Senão, definimos $\bar{x}^k = x^k$ e calculamos, para cada $i = 1, \dots, m$

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k) \quad (3.6)$$

e

$$\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}].$$

Para cada $i = 1, \dots, p$, calculamos

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}, \quad (3.7)$$

e

$$\bar{\mu}_i^{k+1} \in [0, \mu_{max}].$$

Passo 5. *Atualizar o raio da região de confiança e o parâmetro de penalidade.*

Se $k > 1$ e $R_k > \tau R_{k-1}$ definimos

$$\rho_{k+1} = \eta \rho_k.$$

Senão, definimos

$$\rho_{k+1} = \rho_k.$$

Escolhemos $\Delta_{k+1} > 0$ de maneira que

$$\Delta_{k+1} \geq \frac{\beta_1}{R_k} \quad (3.8)$$

e

$$\Delta_{k+1} \geq \beta_2 \rho_{k+1} \quad (3.9)$$

$k \leftarrow k + 1$ e voltamos ao passo 2.

Observação 3.1.1. *As condições (3.1)-(3.4) significam que o ponto x^k é um ponto KKT aproximado do subproblema*

$$\text{Minimizar } L_\rho(x, \bar{\lambda}^k, \bar{\mu}^k) \text{ s.a. } \underline{h}(x) = 0, \underline{g}(x) \leq 0, x \in B_k.$$

Assim, para cada iteração externa desejamos minimizar (aproximadamente) o Lagrangiano Aumentado sujeito a restrições (simples) definidas pelo conjunto Ω e pela região de confiança B_k .

Observação 3.1.2. *No passo 5 a região de confiança é atualizada. Note que o ponto de referência é o ponto que define a região de confiança e assim como no Algoritmo 2 e 3 o ponto de referência é o ponto com melhor medida de complementariedade-factibilidade. A regra (3.8) impõe que o raio da região de confiança deve tender ao infinito se a medida de factibilidade-complementariedade R_k tende a zero. Em (3.9) temos que, se o parâmetro de penalidade está adequado a factibilidade do problema, então, podemos fornecer uma região de confiança maior.*

Lema 3.1.3. *Seja $\{x^k\}$ uma sequência limitada gerada pelo Algoritmo 4 e suponhamos que existe um conjunto infinito K , de índices, tais que*

$$\lim_{k \in K} R_k = 0. \quad (3.10)$$

Então,

$$\lim_{k \rightarrow \infty} R_k = 0. \quad (3.11)$$

Demonstração: Primeiro, vamos supor que $\{\rho_k\}$ é uma sequência limitada. Posto que, $R_k \leq \tau R_{k-1}$ para todo k suficientemente grande, a tese está provada.

Agora, vamos supor que $\lim_{k \rightarrow \infty} \rho_k = \infty$.

Por (3.5), temos, para todo $k \in \mathbb{N}$,

$$\begin{aligned} & f(x^k) + \frac{\rho_k}{2} \left\{ \sum_{i=1}^m \left[h_i(x^k) + \frac{\bar{\lambda}_i}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x^k) + \frac{\bar{\mu}_i}{\rho_k}) \right]^2 \right\} \\ & \leq f(\bar{x}^{k-1}) + \frac{\rho_k}{2} \left\{ \sum_{i=1}^m \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i}{\rho_k}) \right]^2 \right\}. \end{aligned}$$

Dividindo por ρ_k obtemos:

$$\begin{aligned} & \frac{1}{\rho_k} f(x^k) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(x^k) + \frac{\bar{\lambda}_i}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x^k) + \frac{\bar{\mu}_i}{\rho_k}) \right]^2 \right\} \\ & \leq \frac{1}{\rho_k} f(\bar{x}^{k-1}) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i}{\rho_k}) \right]^2 \right\}. \end{aligned} \quad (3.12)$$

Pela definição de \bar{x}^{k-1} nós podemos escrever:

$$\{\bar{x}^0, \bar{x}^1, \bar{x}^2, \dots\} = \{x^{k_0}, x^{k_1}, x^{k_2}, \dots\}$$

onde $k_0 \leq k_1 \leq k_2 \leq \dots$. Além disso, como $\lim_{k \in K} R_k = 0$, nós temos que

$$\lim_{j \rightarrow \infty} R_{k_j} = 0. \quad (3.13)$$

Claramente, (3.13) implica que

$$\lim_{j \rightarrow \infty} \|h(x^{k_j})\|^2 + \sum_{i=1}^p \max\{0, g_i(x)\}^2 = 0.$$

Portanto,

$$\lim_{k \rightarrow \infty} \|h(\bar{x}^{k-1})\|^2 + \sum_{i=1}^p \max\{0, g_i(\bar{x}^{k-1})\}^2 = 0.$$

Dessa forma, o lado direito da equação (3.12) tende a zero quando k tende ao infinito.

Assim,

$$\lim_{k \rightarrow \infty} \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(x^k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max\{0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k}\} \right]^2 \right\} = 0.$$

Como $\rho_k \rightarrow \infty$ e $\bar{\mu}^k, \bar{\lambda}^k$ são limitados, segue que

$$\lim_{k \rightarrow \infty} \|h(x^k)\| = 0 \quad (3.14)$$

e

$$\lim_{k \rightarrow \infty} \max\{0, g_i(x^k)\} = 0, \forall i = 1, \dots, p.$$

Então,

$$\lim_{k \rightarrow \infty} V_i^k = 0, \forall i = 1, \dots, p. \quad (3.15)$$

Por (3.14) e (3.15) nós obtemos o resultado. \square

Lema 3.1.4. *Seja $\{x^k\}$ uma sequência limitada gerada pelo Algoritmo 4 e suponhamos que existe um conjunto infinito K de índices tais que*

$$\lim_{k \in K} R_k = 0.$$

Então,

$$\lim_{k \rightarrow \infty} \Delta_k = \infty.$$

Demonstração: O resultado segue diretamente de (3.8) e do lema anterior. \square

3.2 Convergência e limitação do parâmetro de penalidade

Lema 3.2.1. *Suponhamos que $\{x^k\}$ é uma sequência limitada gerada pelo Algoritmo 4. Então, para cada $k = 1, 2, 3, \dots$ temos que $x^k \in B_k$ e:*

$$\|P_{B_k \cap \Omega}[x^k - (\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k)] - x^k\| \leq \varepsilon_k, \quad (3.16)$$

onde

$$\begin{aligned} w^k &\leq 0, w_i^k = 0 \text{ sempre que } \underline{g}(x^k) < -\varepsilon_k, \\ \underline{g}_i(x^k) &\leq \varepsilon_k \forall i = 1, \dots, \underline{p}, \|\underline{h}(x^k)\| \leq \varepsilon_k. \end{aligned}$$

Demonstração: A prova segue de (3.1)-(3.4) usando a definição (3.6) e (3.7). □

Lema 3.2.2. *Suponha que $\{x^k\}$ é uma sequência limitada gerada pelo Algoritmo 4 e suponha que $\lim_{k \rightarrow \infty} \rho_k = \infty$. Então, existe uma constante $c > 0$ tal que para cada k suficientemente grande temos:*

$$\|\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k\| \leq c\varepsilon_k, \quad (3.17)$$

onde

$$\begin{aligned} w^k &\leq 0, w_i^k = 0 \text{ sempre que } \underline{g}(x^k) < -\varepsilon_k, \\ \underline{g}_i(x^k) &\leq \varepsilon_k \forall i = 1, \dots, \underline{p}, \|\underline{h}(x^k)\| \leq \varepsilon_k. \end{aligned}$$

Demonstração: Defina:

$$g^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k. \quad (3.18)$$

Então, por (3.16), temos:

$$\|P_{B_k \cap \Omega}(x^k - g^k) - x^k\| \leq \varepsilon_k \quad (3.19)$$

para todo $k = 1, 2, \dots$

Pela equivalência de normas em \mathbb{R}^n , existe $c_1 > 0$ tal que

$$\|P_{B_k \cap \Omega}(x^k - g^k) - x^k\|_\infty \leq c_1 \varepsilon_k \quad (3.20)$$

Agora, pela definição de B_k ,

$$[P_{B_k \cap \Omega}(x^k - g^k)]_i = \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\}.$$

Portanto, para cada $k \in \mathbb{N}$, $i \in \{1, \dots, n\}$,

$$|\max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k| \leq c_1 \varepsilon_k.$$

Assim,

$$-c_1 \varepsilon_k \leq \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k \leq c_1 \varepsilon_k.$$

Dividindo por ρ_k , obtemos:

$$\frac{-c_1 \varepsilon_k + x_i^k}{\rho_k} \leq \max\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} - \frac{\Delta_k}{\rho_k}, \min\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} + \frac{\Delta_k}{\rho_k}, \frac{x_i^k}{\rho_k} - \frac{g_i^k}{\rho_k}\right\}\right\} \leq \frac{c_1 \varepsilon_k - x_i^k}{\rho_k}. \quad (3.21)$$

Como $\{\bar{x}_i^{k-1}\}$ e $\{x_i^k\}$ são limitados, nós temos que \bar{x}_i^{k-1}/ρ_k e x_i^k/ρ_k tendem a zero. Por (3.9) temos, para k suficientemente grande, que:

$$\begin{aligned} \frac{\bar{x}_i^{k-1}}{\rho_k} - \frac{\Delta_k}{\rho_k} &\leq -\frac{\beta_2}{2}, \\ \frac{\bar{x}_i^{k-1}}{\rho_k} + \frac{\Delta_k}{\rho_k} &\geq \frac{\beta_2}{2} \end{aligned}$$

e

$$\left| \frac{\pm c_1 \varepsilon_k + x_i^k}{\rho_k} \right| \leq \frac{\beta_2}{3}.$$

Portanto, por (3.21), temos, para k suficientemente grande que:

$$\frac{-c_1 \varepsilon_k + x_i^k}{\rho_k} \leq \frac{x_i^k}{\rho_k} - \frac{g_i^k}{\rho_k} \leq \frac{c_1 \varepsilon_k + x_i^k}{\rho_k}. \quad (3.22)$$

Então,

$$\frac{-c_1 \varepsilon_k}{\rho_k} \leq -\frac{g_i^k}{\rho_k} \leq \frac{c_1 \varepsilon_k}{\rho_k}. \quad (3.23)$$

Multiplicando ambos os lados de (3.23) por ρ_k , obtemos, para k suficientemente grande que:

$$|g_i^k| \leq c_1 \varepsilon_k.$$

Por (3.18) e pela equivalência de normas em \mathbb{R}^n temos o resultado. \square

Teorema 3.2.3. *Suponha que x^* é um ponto limite de uma sequência limitada gerada pelo Algoritmo 4. Então:*

1. *Ao menos uma das seguintes possibilidades ocorre:*

- *O ponto x^* satisfaz as condições KKT do problema*

$$\text{Minimizar } \|h(x)\|^2 + \|g(x)_+\|^2 \text{ s.a. } \underline{h}(x) = 0, \underline{g}(x) \leq 0.$$

- *A condição CPLD não é satisfeita em x^* com respeito as condições $\underline{h}(x) = 0$ e $\underline{g}(x) \leq 0$.*

2. *Suponha que x^* é um ponto limite factível de PNL. Então, pelo menos uma das seguintes possibilidades ocorrem:*

- *O ponto x^* satisfaz as condições KKT de PNL.*
- *A condição CPLD não é satisfeita em x^* com respeito as condições $h(x) = 0$, $g(x) \leq 0$, $\underline{h}(x) = 0$ e $\underline{g}(x) \leq 0$.*

Demonstração: Vamos provar a primeira parte do teorema. Se $\{\rho_k\}$ é uma sequência limitada, segue que a partir de uma certa iteração ρ_k não é incrementado. Dessa forma, a factibilidade de qualquer ponto limite segue do passo 5 do Algoritmo 4. Se a sequência $\{\rho_k\}$ é ilimitada, temos pelo Lema 3.2.2 que, para k suficientemente grande o Algoritmo 4 pode ser considerado um caso particular do Algoritmo 1. Portanto, a tese segue do Teorema 1.2.1.

Agora, vamos provar a segunda parte do teorema. Neste caso, pelo Lema 3.1.3, nós temos que $\lim_{k \rightarrow \infty} R_k = 0$. Portanto, por (3.8), $\lim_{k \rightarrow \infty} \Delta_k = \infty$.

Como no Lema 3.2.2, definimos:

$$g^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k.$$

Então, por (3.16),

$$\|P_{B_k \cap \Omega}(x^k - g^k) - x^k\| \leq \varepsilon_k$$

para todo $k = 1, 2, \dots$

Pela equivalência de normas em \mathbb{R}^n , existe $c_1 > 0$ tal que

$$\|P_{B_k \cap \Omega}(x^k - g^k) - x^k\|_\infty \leq c_1 \varepsilon_k.$$

Pela definição de B_k ,

$$[P_{B_k \cap \Omega}(x^k - g^k)]_i = \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\}.$$

Portanto, para cada $k \in \mathbb{N}$, $i \in \{1, \dots, n\}$,

$$|\max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k| \leq c_1 \varepsilon_k. \quad (3.24)$$

Assim,

$$-c_1 \varepsilon_k + x_i^k \leq \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} \leq c_1 \varepsilon_k + x_i^k.$$

Portanto, pela limitação de $\{x^k\}$, existem constantes $c_2, c_3 \in \mathbb{R}$ tais que:

$$-c_2 \leq \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} \leq c_3.$$

Como $\Delta_k \rightarrow \infty$ e $\{\bar{x}^{k-1}\}$ é limitada, a desigualdade acima pode ocorrer somente se, para k suficientemente grande, ocorrer:

$$\bar{x}_i^{k-1} - \Delta_k < x_i^k - g_i^k < \bar{x}_i^{k-1} + \Delta_k.$$

Portanto, por (3.24), $|g_i^k| \leq c_1 \varepsilon_k$ para k suficientemente grande. Isto implica que, para k suficientemente grande, a sequência $\{x^k\}$ pode ser pensada como uma sequência gerada pelo Algoritmo 1. Dessa forma, a tese segue pelo Teorema 1.2.3. \square

Uma vez que provamos os resultados de convergência, vamos provar o resultado de limitação de parâmetro de penalidade. Dessa forma, vamos considerar as seguintes hipóteses gerais (para o Algoritmo 4):

1. A sequência $\{x^k\}$ gerada pelo Algoritmo 4 converge à x^* .
2. O ponto x^* é admissível (ou seja, $h(x^*) = 0$ e $g(x^*) \leq 0$).
3. O ponto x^* é regular.

4. O multiplicador λ_i^* pertence a $(\lambda_{min}, \lambda_{max})$ para todo $i = 1, \dots, m$. Analogamente, cada multiplicador μ_i^* pertence a $[0, \mu_{max})$ para todo $i = 1, \dots, p$.
5. As funções f , h e g admitem derivadas segundas contínuas em uma vizinhança de x^* .
6. Para cada $i = 1, \dots, p$ tal que $g_i(x^*) = 0$ temos que $\mu_i > 0$ (*Complementariedade estrita*).
7. Seja $A \in \mathbb{R}^{(m+q) \times n}$ a matriz cujas linhas são os gradientes das restrições de igualdade seguidas dos gradientes das q restrições ativas ($g_i(x^*) = 0$) em x^* . Seja $Z \in \mathbb{R}^{n \times (n-m-q)}$ uma matriz cujas colunas formam uma base do núcleo de A . Então, $Z^T \nabla^2 L_0(x^*, \lambda^*, \mu^*) Z$ é definida positiva.
8. Existe $k_0 \in \mathbb{N}$, tal que para $k \geq k_0$, $R_k = \min\{\max\{R_{tol}, R_0\}, \dots, R_k\}$, ou seja, para k suficientemente grande atualizamos os multiplicadores como no Algoritmo 1.

Teorema 3.2.4. *Suponhamos, além das hipóteses anteriores, que existe uma sequência $\{\eta_k\}$ que converge a zero, tal que*

$$\varepsilon_k \leq \eta_k \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}$$

para todo $k \in \mathbb{N}$. Então, a sequência de parâmetros de penalidade $\{\rho_k\}$ está limitada.

Demonstração: A demonstração deste teorema segue imediata do Lema 3.2.2 e do Teorema 1.2.5. □

3.3 Minimizadores globais

Nesta seção provaremos que pontos limites de sequências geradas pelo Algoritmo 4, são admissíveis se a região factível é não vazia, e mais são minimizadores globais de PNL. Para isto, vamos supor que a cada iteração externa o subproblema a ser resolvido tem minimizador global. Dessa forma, uma pequena modificação do Algoritmo é proposta.

Algoritmo 5

Os parâmetros que definem o algoritmo são: $\tau \in [0, 1)$, $\eta > 1$, $\lambda_{min} < \lambda_{max}$, $\mu_{max} > 0$, $\beta_1 > 0$, $\beta_2 > 0$, R_{tol} . Na primeira iteração externa usamos um parâmetro de penalidade $\rho_1 > 0$ e estimativas $\bar{\lambda}^1 \in \mathbb{R}^m$ e $\bar{\mu}^1 \in \mathbb{R}^p$ para os multiplicadores de Lagrange tais que,

$$\bar{\lambda}_i^1 \in [\lambda_{min}, \lambda_{max}], \forall i = 1, \dots, m \text{ e } \mu_i^1 \in [0, \mu_{max}], \forall i = 1, \dots, p.$$

Assumimos que $x^0 \in \mathbb{R}^n$ é um ponto inicial arbitrário que coincide com o *ponto de referência* inicial \bar{x}^0 e $\Delta_1 > 0$ é um raio arbitrário para a região de confiança.

Para cada $k \in \{1, 2, \dots\}$ definimos

$$B_k = \{x \in \mathbb{R}^n \mid \|x - \bar{x}^{k-1}\| \leq \Delta_k\}.$$

Finalmente, $\{\varepsilon_k\}$ é uma sequência de números positivos que satisfaz

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0.$$

Passo 1. *Inicialização.*

$$k \leftarrow 1.$$

Passo 2. *Resolver o subproblema.*

Calcular $x^k \in B^k \cap \Omega$ solução global do subproblema

$$\min f(x) + \frac{\rho_k}{2} \left\{ \sum_{i=1}^m \left[h_i(x) + \frac{\lambda_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max(0, g_i(x) + \frac{\mu_i^k}{\rho_k}) \right]^2 \right\}$$

sujeito a $x \in \Omega \cap B_k$.

Passo 3. *Calcular a factibilidade e a complementariedade.*

Para cada $i = 1, \dots, p$ calculamos

$$V_i^k = \max\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\},$$

e definimos

$$R_k = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}.$$

Passo 4. *Atualizar os multiplicadores de Lagrange e o ponto de referência.*

Se $R_k \neq \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$, definimos $\bar{x}^k = \bar{x}^{k-1}$, $\lambda^{k+1} = \lambda^k$, $\bar{\lambda}^{k+1} = \bar{\lambda}^{k+1} = \bar{\lambda}^k$, $\mu^{k+1} = \mu^k$, $\bar{\mu}^{k+1} = \bar{\mu}^k$. Senão, definimos $\bar{x}^k = x^k$ e calculamos, para cada $i = 1, \dots, m$

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k) \quad (3.25)$$

e

$$\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}].$$

Para cada $i = 1, \dots, p$, calculamos

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}, \quad (3.26)$$

e

$$\bar{\mu}_i^{k+1} \in [0, \mu_{max}].$$

Passo 5. *Atualizar o raio da região de confiança e o parâmetro de penalidade.*

Se $k > 1$ e $R_k > \tau R_{k-1}$ definimos

$$\rho_{k+1} = \eta \rho_k.$$

Senão, definimos

$$\rho_{k+1} = \rho_k.$$

Escolhemos $\Delta_{k+1} > 0$ de maneira que

$$\Delta_{k+1} \geq \frac{\beta_1}{R_k} \quad (3.27)$$

e

$$\Delta_{k+1} \geq \beta_2 \rho_{k+1} \quad (3.28)$$

$k \leftarrow k + 1$ e voltamos ao passo 2.

Teorema 3.3.1. *Suponhamos que a região admissível do problema PNL é não vazia. Suponhamos além disso, que a sequência $\{x^k\}$ gerada pelo Algoritmo 5 é limitada, e seja x^* um ponto limite desta sequência. Então x^* é admissível.*

Demonstração: Seja $K \subset \mathbb{N}$ um conjunto infinito de índices tal que

$$\lim_{k \in K} x^k = x^*.$$

Dividiremos a demonstração em dois casos:

- I. A sequência ρ_k é limitada.
- II. A sequência ρ_k não é limitada.

Caso I. Neste caso, a partir de uma certa iteração, o parâmetro de penalidade não é mais atualizado. Portanto, segue do passo 3 e 5 do algoritmo que:

$$\lim_{k \rightarrow \infty} \|h(x^k)\| = \lim_{k \rightarrow \infty} \|V^k\| = 0.$$

Assim,

$$h(x^*) = 0.$$

Ainda mais, se $g_j(x^*) > 0$ temos que existe $c > 0$ com $g_j(x^k) > c > 0$ para $k \in K$ suficientemente grande, isso contradiz o fato de que $V_j^k \rightarrow 0$.

Logo, $g_i(x^*) \leq 0$ para todo $i = 1, 2, \dots, p$. Posto que, $x^k \in \Omega$, $\forall k \in \mathbb{N}$ e Ω é fechado então, $x^* \in \Omega$ e portanto x^* é admissível. Assim, o resultado está provado para o caso em $\{\rho_k\}$ é limitado.

Caso II. Novamente, como $x^k \in \Omega$ para todo k , e Ω é fechado, então, $x^* \in \Omega$. Suponhamos que x^* não é admissível então existe $z \in \Omega$ tal que

$$\|h(x^*)\|^2 + \|g(x^*)_+\|^2 > \|h(z)\|^2 + \|g(z)_+\|^2 = 0.$$

Como h e g são contínuas, $\bar{\lambda}^k$ e $\bar{\mu}^k$ são limitados e $\rho_k \rightarrow +\infty$ segue que, existe uma constante $c > 0$ e $k_0 \in K$ tais que

$$\left\| h(x^k) + \frac{\bar{\lambda}^k}{\rho_k} \right\| + \left\| \left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\| > \left\| h(z) + \frac{\bar{\lambda}^k}{\rho_k} \right\| + \left\| \left(g(z) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\| + c. \quad (3.29)$$

Portanto, para esses índices k ,

$$f(x^k) + \frac{\rho_k}{2} \left[\left\| h(x^k) + \frac{\bar{\lambda}^k}{\rho_k} \right\| + \left\| \left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\| \right] > \quad (3.30)$$

$$f(z) + \frac{\rho_k}{2} \left[\left\| h(z) + \frac{\bar{\lambda}^k}{\rho_k} \right\| + \left\| \left(g(z) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\| \right] + \frac{\rho_k c}{2} + f(x^k) - f(z). \quad (3.31)$$

Como $\rho_k \rightarrow +\infty$, segue que $\Delta_k \rightarrow +\infty$, logo, para k suficientemente grande, $z \in B_k$. Além disso,

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) > L_{\rho_k}(z, \bar{\lambda}^k, \bar{\mu}^k) + \frac{(\rho_k - 1)c}{2} + f(x^k) - f(z).$$

Agora, como $\lim_{k \in K} x^k = x^*$ e f é contínua, temos que, para k suficientemente grande,

$$\frac{(\rho_k - 1)c}{2} + f(x^k) - f(z) > 0.$$

Logo,

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) > L_{\rho_k}(z, \bar{\lambda}^k, \bar{\mu}^k).$$

O que contradiz o fato de que x^k é uma solução global do subproblema definido por ρ_k . Assim, x^* é admissível. □

Teorema 3.3.2. *Suponhamos que a região admissível do problema PNL é não vazia e que a sequência $\{x^k\}$ está bem definida pelo Algoritmo 5 e é limitada sendo x^* um ponto limite. Então, x^* é um minimizador global do problema PNL.*

Demonstração: Seja $K \subset \mathbb{N}$ um subconjunto infinito de índices tal que

$$\lim_{k \in K} x^k = x^*.$$

Segue do teorema anterior que x^* é admissível.

Novamente, vamos dividir a demonstração em dois casos:

- I. A sequência ρ_k é limitada.
- II. A sequência ρ_k não é limitada.

Caso I. Neste caso, existe $k_0 \in \mathbb{N}$ de forma que não atualizamos mais o parâmetro de penalidade para todo $k \geq k_0$. Como as sequências $\{\bar{\lambda}^k\}$ e $\{\bar{\mu}^k\}$ estão em um compacto, existem $K_1 \subset K$ (K_1 infinito), $\lambda^* \in [\lambda_{\min}, \lambda_{\max}]^m$, $\mu^* \in [0, \mu_{\max}]^p$ tais que

$$\lim_{k \in K_1} x^k = x^*, \lim_{k \in K_1} \bar{\lambda}^k = \lambda^*, \lim_{k \in K_1} \bar{\mu}^k = \mu^*.$$

Mas pelo passo 3 do Algoritmo 5 devemos ter $\lim_{k \rightarrow \infty} \max\{g_i(x^k), -\bar{\mu}_i^k/\rho_{k_0}\} = 0$. Portanto, se $g_i(x^*) < 0$ devemos ter, forçosamente, que $\mu_i^* = 0$. Dito de outra forma,

$$g_i(x^*)\mu_i^* = 0 \text{ para todo } i = 1, \dots, p.$$

Agora, seja $z \in \Omega$ um ponto admissível arbitrário. Como o parâmetro de penalidade não é atualizado para k suficientemente grande, temos que, $R_k \rightarrow 0$ ($k \geq k_0$). Logo, pelo Lema 3.2.4 segue que $\lim_{k \rightarrow +\infty} \Delta_k = +\infty$. Assim, existe $k_1 \in K_1$ tal que

$$L_{\rho_{k_1}}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_{k_1}}(z, \bar{\lambda}^k, \bar{\mu}^k), \forall k \geq k_1.$$

Como $g(z) \leq 0$ e $\bar{\mu}^k/\rho_k \geq 0$,

$$\left\| \left(g(z) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\|^2 \leq \left\| \frac{\bar{\mu}^k}{\rho_k} \right\|^2.$$

Então, como $h(z) = 0$ temos:

$$L_{\rho_{k_1}}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq f(z) + \frac{\rho_{k_1}}{2} \left[\left\| \frac{\bar{\lambda}^k}{\rho_{k_1}} \right\|^2 + \left\| \frac{\bar{\mu}^k}{\rho_{k_1}} \right\|^2 \right], \forall k \geq k_1.$$

Passando o limite na desigualdade acima obtemos:

$$f(x^*) + \frac{\rho_{k_1}}{2} \left\| \left(g(x^*) + \frac{\mu}{\rho_{k_1}} \right)_+ \right\|^2 \leq f(z) + \frac{\rho_{k_1}}{2} \left\| \frac{\mu^*}{\rho_{k_1}} \right\|^2,$$

ou seja,

$$f(x^*) + \frac{\rho_{k_1}}{2} \sum_{i=1}^p \left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_1}} \right)_+^2 \leq f(z) + \frac{\rho_{k_1}}{2} \sum_{i=1}^p \left(\frac{\mu_i^*}{\rho_{k_1}} \right)^2. \quad (3.32)$$

Mas

$$\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_1}} \right)_+^2 = \max \left\{ 0, g_i(x^*) + \frac{\mu_i^*}{\rho_{k_1}} \right\}^2.$$

Portanto, se $g_i(x^*) = 0$,

$$\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_1}} \right)_+^2 = \left(\frac{\mu_i^*}{\rho_{k_1}} \right)^2.$$

Agora, se $g_i(x^*) < 0$ temos, $\mu_i^* = 0$. Logo,

$$\left(g_i(x^*) + \frac{\mu_i^*}{\rho_{k_1}} \right)_+^2 = 0.$$

Dessa forma, por (3.32), obtemos que $f(x^*) \leq f(z)$. Posto que z é um ponto admissível arbitrário, resulta que x^* é minimizador global.

Caso II. Seja $z \in \Omega$ um ponto admissível. Como $\rho_k \rightarrow +\infty$ temos, pelo Algoritmo 5, que $\Delta_k \rightarrow +\infty$. Assim, existe k_0 tal que $z \in B_k$ para todo $k \geq k_0$. Logo,

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_k}(z, \bar{\lambda}^k, \bar{\mu}^k), \forall k \geq k_0.$$

Como $h(z) = 0$ segue que:

$$\left\| h(z) + \frac{\bar{\lambda}^k}{\rho_k} \right\|^2 = \left\| \frac{\bar{\lambda}^k}{\rho_k} \right\|^2, \forall k \geq k_0.$$

Agora, como $g(z) \leq 0$ temos que:

$$\left\| \left(g(z) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\|^2 = \left\| \frac{\bar{\mu}^k}{\rho_k} \right\|^2, \forall k \geq k_0.$$

Dessa forma,

$$f(x^k) + \frac{\rho_k}{2} \left[\left\| \frac{\bar{\lambda}^k}{\rho_k} \right\|^2 + \left\| \left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k} \right)_+ \right\|^2 \right] \leq f(z) + \frac{\rho_k}{2} \left[\left\| \frac{\bar{\lambda}^k}{\rho_k} \right\|^2 + \left\| \frac{\bar{\mu}^k}{\rho_k} \right\|^2 \right], \forall k \geq k_0.$$

Observando que $\rho_k \rightarrow +\infty$ e $\{\bar{\lambda}^k\}$, $\{\bar{\mu}^k\}$, $\{x^k\}$ são limitadas segue, tomando o limite, que:

$$f(x^*) \leq f(z).$$

Como z é um ponto admissível arbitrário, resulta que x^* é solução global.

Do caso I e do caso II segue a tese do teorema. \square

CAPÍTULO 4

Implementação e testes

No presente capítulo apresentaremos comparações detalhadas entre a implementação Algencan e a implementação do novo algoritmo, a qual denominamos Algencan-OTR. Alguns detalhes da implementação deste novo método são expostos. Apresentaremos também uma versão de Algencan e uma versão de Algencan-OTR para a minimização global. Finalizando o capítulo vamos expor comparações entre Algencan-OTR e RAlgencan. Esta última é a implementação do método apresentado no capítulo 2.

4.1 Detalhes de implementação

A implementação do Algoritmo 4, Algencan-OTR, foi realizada por algumas modificações na implementação Algencan. Essencialmente, introduzimos restrições de caixa artificiais e adaptativas nos subproblemas do método de Lagrangiano Aumentado. A versão utilizada de Algencan foi a 2.2.1 (veja [28]). Uma observação importante é que os subproblemas de Algencan são resolvidos por usar o robusto Gencan. Gencan é usado para a minimização de funções com restrições de caixa e sua evolução reflete resultados melhores quando incorporada ao método de Lagrangiano Aumentado citado no capítulo 1. Note ainda que, em Algencan-

OTR, não alteramos a complexidade dos subproblemas, isto é, continuamos minimizando em restrições de caixa, logo, o novo método pode aproveitar as boas características de Gencan. Os detalhes expostos nesta seção estão como em [8].

Seja $\varepsilon > 0$ um parâmetro de tolerância para a factibilidade, complementaridade e otimalidade. Em cada iteração k , se $k \neq 1$ e se as condições (3.1-3.4) são satisfeitas substituindo k por $k - 1$, $\varepsilon_k = \sqrt{\varepsilon}$ e $R_k \leq \sqrt{\varepsilon}$ então, definimos $\varepsilon_k = 0.1\varepsilon_{k-1}$. Caso contrário, nós definimos $\varepsilon_k = \sqrt{\varepsilon}$. Paramos o algoritmo em uma iteração k quando (3.1-3.4) é satisfeita substituindo ε_k por ε e $R_k \leq \varepsilon$. Nós definimos os seguintes valores para os parâmetros em questão: $\varepsilon = 10^{-8}$, $\tau = 0.5$, $\eta = 10$, $\lambda_{min} = -10^{20}$, $\lambda_{max} = \mu_{max} = 10^{20}$. Consideramos $\lambda_0 = 0$ e $\mu_0 = 0$.

Em cada iteração k , calculamos $\bar{\lambda}^k \equiv P_{[\lambda_{min}, \lambda_{max}]^m}(\lambda^k)$ e $\bar{\mu}^k \equiv P_{[0, \mu_{max}]^p}(\mu^k)$. Quanto aos novos parâmetros introduzidos, usamos $\beta_1 = \beta_2 = 10^{-8}$, $R_{tol} = 0.1$ e $\Delta_1 = \infty$. No passo 5, se $R_k > 100 \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$, definimos $\bar{\Delta}_{k+1} = 0.5\|x^k - \bar{x}^k\|_\infty$ e $\Delta_{k+1} \equiv \max\{\bar{\Delta}_{k+1}, \beta_1/R_k, \beta_2\rho_k\}$. Caso contrário, definimos $\Delta_{k+1} = \infty$. Note que ao definir Δ_{k+1} desta forma, a condição (3.8) e (3.9) são satisfeitas.

Os algoritmos foram implementados em Fortran 77 usando precisão dupla e compilado com gfortran (GNU Fortran (GCC) 4.2.4). A opção de compilação foi otimizada, a saber, -O4. Todos os experimentos foram executados num Intel Core2 QUAD com 4.0GB de memória RAM e o sistema operacional foi o Linux.

Dois aspectos da implementação Algencan contribuem para o não surgimento do fenômeno de voracidade, o *escalamento* e o *parâmetro de penalidade inicial*.

Escalamento. A implementação Algencan resolve, não o problema PNL citado, mas um problema escalado da forma:

$$\text{Minimizar } \hat{f}(x) \text{ sujeita a } \hat{h}(x) = 0, \hat{g}(x) \leq 0, x \in \Omega = \{x \in \mathbb{R}^n | \hat{\ell} \leq x \leq \hat{u}\}, \quad (4.1)$$

onde:

$$\hat{f}(x) \equiv s_f f(x) \text{ e } s_f = \frac{1}{\max\{1, \|\nabla f(x^0)\|_\infty\}},$$

$$\hat{h}_i(x) \equiv s_{h_i} h_i(x) \text{ e } s_{h_i} = \frac{1}{\max\{1, \|\nabla h_i(x^0)\|_\infty\}}, i = 1, \dots, m,$$

$$\hat{g}_i(x) \equiv s_{g_i} g_i(x) \text{ e } s_{g_i} = \frac{1}{\max\{1, \|\nabla g_i(x^0)\|_\infty\}}, i = 1, \dots, p,$$

e $\hat{\ell}_i \equiv \max\{-10^{20}, \ell_i\}$ e $\hat{u}_i \equiv \min\{10^{20}, u_i\}$ para todo i . O critério de parada está associado ao sucesso considerando a factibilidade e a complementaridade do problema original (sem escalar restrições) e a otimalidade do problema escalado (4.1). No caso particular em que $m = p = 0$, definimos $s_f \equiv 1$.

Parâmetro de penalidade inicial. A função Lagrangiano Aumentado para o problema (4.1) e para o caso particular de $(\lambda, \mu) = 0$ se reduz a

$$L_\rho(x, 0, 0) = \hat{f}(x) + \frac{\rho}{2} C(x),$$

onde

$$C(x) = \sum_{i=1}^m \hat{h}_i(x)^2 + \sum_{i=1}^p \max(0, \hat{g}_i(x))^2.$$

Então, se $C(x) \neq 0$, o valor de ρ que mantém o Lagrangiano Aumentado “bem balanceado” é dado por $\rho = 0.5 \frac{\hat{f}(x)}{C(x)}$. Em Algencan, assumimos que $(\lambda^0, \mu^0) = 0$ e definimos:

$$\rho_1 = \min \left\{ \max \left\{ 10^{-8}, 10 \frac{\max(1, |\hat{f}(x^0)|)}{\max(1, C(x^0))} \right\}, 10^8 \right\}. \quad (4.2)$$

Além disso, tentamos fazer uma escolha do parâmetro de penalidade um pouco mais independente das aproximação inicial x^0 . Neste caso, o que é feito é calcular rapidamente x^1 como solução do primeiro subproblema (limitando a 10 o número de iterações para resolver o subproblema) e calculamos ρ_2 como em (4.2) porém usando o ponto x^1 . Na sequência, as regras para atualizar o parâmetro de penalidade descrita no passo 4 do algoritmo são aplicadas para $k \geq 3$.

Os dois detalhes de implementação descritos acima reduzem a chance de Algencan ser atraído por pontos infactíveis nas primeiras iterações por, basicamente, ignorar as restrições. Entretanto, exemplos específicos mostram que o tal fenômeno pode ocorrer mesmo com as duas considerações citadas.

4.2 Testes numéricos

Na presente seção vamos apresentar exemplos em que ocorrem voracidade e analisar de forma detalhada o que ocorre com Algencan e Algencan-OTR. Em seguida, vamos mostrar uma comparação numa quantidade grande de problemas testes, para verificar que Algencan-OTR mantém o bom desempenho de Algencan, ou seja, a adição das caixas adaptativas não prejudica o desempenho do método em problemas gerais.

4.2.1 Exemplos específicos de voracidade

Para os exemplos numéricos que vamos expor foi usado a versão Algencan 2.2.1.

Problema A:

$$\begin{aligned} \min \quad & -\sum_{i=1}^n x_i^8 + x_i \\ \text{s.a} \quad & \sum_{i=1}^n x_i^2 \leq 1, \end{aligned}$$

Seja $n = 10$ e considere o ponto inicial $x^0 = \frac{1}{n}\bar{x}$, onde \bar{x}_i são números aleatórios distribuídos $[0.9, 1.1]$.

Algencan. Observe que x^0 é um ponto factível, $s_f = s_{g_1} = 1.0D+00$ e $\rho_1 = 1.0D+01$. Na primeira iteração para resolver o primeiro subproblema, o solver interno, Gencan, dá um grande passo na direção do gradiente negativo obtendo o ponto x^1 no qual a função objetivo que está sendo minimizada (o Lagrangiano Aumentado) tem um valor menor que -10^{20} . A função objetivo escalada em x^1 é aproximadamente $-6.5D+21$ e a norma das restrições é aproximadamente $2.0D+06$. O valor de ρ_2 é $1.0d+08$. Nem este valor de ρ_2 nem os posteriores são capazes de remover Algencan deste ponto. Algencan pára após poucas iterações externas em um ponto infactível.

Algencan-OTR. temos que x^0 é um ponto factível. Desta forma, um ponto x^k apenas é aceito como novo ponto de referência se $R_k \leq R_{tol} = 0.1$. Em outras palavras, enquanto $R_k > R_{tol}$ nós teremos $\bar{x}^k = \bar{x}^0 = x^0$. A solução do primeiro subproblema (aquela que conduziu Algencan para um ponto infactível) é rejeitada pelo Algencan-OTR.

Na próxima iteração, Algencan-OTR usa $\Delta_2 = 2.0D+02$ que ainda é grande. Nas sucessivas iterações Algencan-OTR usa $\Delta_3 = 1.0D+02$, $\Delta_4 = 5.0D+01$, $\Delta_5 = 2.5D+01$, $\Delta_6 = 1.2D+01$, $\Delta_7 = 6.2D+00$, $\Delta_8 = 3.1D+00$. Na iteração externa 8, usando $\Delta_8 = 3.1D+00$ Gencan encontra uma solução x^8 do subproblema tal que $R_8 = 2.0D-02$. Este ponto é aceito como novo ponto de referência, os multiplicadores de Lagrange são atualizados e, em três iterações seguintes (usando $\Delta_9 = \Delta_{10} = \Delta_{11} = \infty$), Algencan-OTR encontra a solução x^* tal que $x_i^* \approx -3.16227766016870D-01$, para todo i .

Problema B:

$$\begin{aligned} \min \quad & -\exp\left(\sum_{i=1}^n x_i^2 + 0.01\right)^{-1} \\ \text{s.a} \quad & \sum_{i=1}^n x_i = 1, \end{aligned}$$

Seja $n = 10$ e considere o ponto inicial e $x_i = \frac{1}{10}$, $i = 1, \dots, 10$.

Algencan. O comportamento de Algencan é essencialmente o mesmo do caso anterior. Neste, temos $s_f = 3.9D-06$, $s_{h_1} = 1.0D+00$ e $\rho_1 = 1.0D+01$. A função objetivo escalada vale $-5.11D-02$ e a norma das restrições vale $2.0D-02$. Na primeira iteração ocorre voracidade e no ponto x^1 obtido a função escalada vale $-2.523028D+22$ e a medida de complementariedade factibilidade vale $1.D+00$. O parâmetro de penalidade ρ_2 para a próxima iteração vale $1.0D+08$. As normas das restrições variam entre $9.0D-01$ e $1.0D+00$ por várias iterações seguintes ao passo que o parâmetro de penalidade cresce. No final, $\rho_{14} = 1.0D+21$ e Algencan pára num ponto infactível.

Algencan-OTR. Neste problema, temos que $R_0 = 2.D-02$. Como R_0 é menor que R_{tol} , apenas um ponto x^k tal que $R_k \leq R_{tol}$ poderá ser aceito como ponto de referência. Usando $\rho_1 = 10$ e $\Delta_1 = \infty$, a solução do primeiro subproblema (com $R_1 = 1.D+00$) não é aceita como novo ponto de referência. Agora, como $R_1 \leq 10^2(R_0)$ caixa adaptativa não é reduzida. Com o parâmetro de penalidade $\rho_2 = 1.0D+02$ o iterando obtido x^2 com $R_2 = 6.0D-03$ é aceito como novo ponto de referência. A partir deste ponto e sempre usando $\Delta_k = \infty$, Algencan-OTR encontra a solução em 7 iterações, parando na iteração 9, quando encontra $x_i^* = 0.1$ para todo i e $f(x^*) = -8.8742D+03$.

Problema C:

$$\begin{aligned} \min \quad & -x \exp(-xy) \\ \text{s.a} \quad & -(x+1)^3 + 3(x+1) + y = 1.5 \\ & -10 \leq x, y \leq 10. \end{aligned}$$

Algencan. Vamos considerar o ponto inicial $x^0 = (-1, 1.5)^T$. Para este problema temos $s_f = 8.9\text{D}-02$ e $s_{g_1} = 1.0\text{D}+00$, e $\rho_1 = 1.0\text{D}+01$. O ponto x^0 é factível e o valor da função objetivo escalada neste ponto é $4.0\text{D}-01$. Quando resolve o primeiro subproblema acaba encontrando um ponto x^1 cuja função objetivo escala vale aproximadamente $-7.1325\text{D}+24$ e a norma das restrições valendo $2.0\text{D}+02$. O parâmetro de penalidade ρ_2 vale $1.0\text{D}+08$ porém nem este, nem os seguintes parâmetros conseguem fazer com que o método convirja. Algencan pára depois de 10 iterações num ponto infactível.

Algencan-OTR. Novamente, o ponto inicial é factível e qualquer ponto com $R_k > R_{tol}$ não será aceito para atualizar o ponto de referência. x^1 com $\hat{f}(x^1) = -7.132532\text{D}+24$ e $R_1 = 2.0\text{D}+02$ é rejeitado. Com $\Delta_2 = 5.75\text{D}+00$ Gencan converge para x^2 com $\hat{f}(x^2) = -2.115861\text{D}+00$ e $R_2 = 3.0\text{D}-02$, isto é, $\bar{x}^2 = x^2$ é aceito como novo ponto de referência. A partir deste ponto Algencan-OTR realiza mais 4 iterações usando $\Delta_3 = \Delta_4 = \dots \Delta_7 = \infty$ e converge para $x^* \approx (1.3186\text{D}+00, -2.1632\text{D}+00)$ onde $f(x^*) = -2.2849\text{D}+01$.

Problema D:

$$\begin{aligned} \min \quad & -x^6 \\ \text{s.a} \quad & x - 1 = 0 \end{aligned}$$

Neste problema, consideramos o ponto inicial $x=0.8$. Os parâmetros de escalamento são $s_f=5.1\text{D}-01$ e $s_h=1.0\text{D}+00$.

Em Algencan. Usando um parâmetro de penalidade igual a 10, ocorreu voracidade na primeira iteração. Assim, o método encontrou um ponto cuja medida de complementariedade-factibilidade vale $3.0\text{D}+03$ e o valor da função objetivo escalada vale aproximadamente $-6.3\text{D}+20$. Mesmo aumentando o parâmetro de penalidade nas iterações seguintes Algencan decretou que o problema é, possivelmente, infactível.

Em Algencan-OTR. O ponto de referência inicial é o ponto $\bar{x}_0 = 0.8$ com $R_0 = 2.0D - 01$. Como a primeira iteração de Algencan-OTR coincide com a Algencan, o método rejeitou o primeiro iterando e o raio da região de confiança passou a valer $\Delta_2 = 1638.39$. Com este valor, o ponto obtido tem medida de complementariedade-factibilidade igual a $4.0D-02$ e é aceito como novo ponto de referência. O raio da região de confiança passa a valer $+\infty$ e Algencan-OTR converge para o ponto $x \approx -1$ em mais 9 iterações.

4.2.2 Voracidade e minimização global

Consideraremos agora, uma adaptação de Algencan para minimização global. Essencialmente a idéia desta adaptação é resolver cada subproblema da iteração k considerando várias aproximações iniciais aleatórias em vez de apenas considerar um ponto x^{k-1} . Fazendo isso, esperamos aumentar a probabilidade de obter a solução global.

Em particular, $N_{trials} = 100$ pontos aleatórios são gerados por uma distribuição normal com média x_i^{k-1} e desvio padrão $10 \max\{1, |x_i^{k-1}|\}$, para $i = 1, \dots, n$. Quando geramos um ponto aleatório para uma aproximação inicial z , as componentes z_i tais que z_i que não pertence ao intervalo $[\hat{\ell}_i, \hat{u}_i]$ são descartados. Por esta razão, quando $\max\{\hat{u}_i - x_i^{k-1}, \hat{\ell}_i\} < \max\{1, |x_i^{k-1}|\}$, uma distribuição uniforme dentro do intervalo $[\hat{\ell}_i, \hat{u}_i]$ é usada no lugar da distribuição normal. No que segue, estaremos nos referindo a esta versão de Algencan como Algencan-Global. Dessa forma, uma correspondente versão de Algencan-OTR também foi definida, por considerar os pontos iniciais aleatórios dentro do intervalo $[\tilde{\ell}_i, \tilde{u}_i] \equiv B_k \cap [\hat{\ell}_i, \hat{u}_i]$ em vez de considerar pontos aleatórios dentro de $[\hat{\ell}_i, \hat{u}_i]$. A esta versão de Algencan-OTR estaremos nos referindo por Algencan-OTR-Global.

O exemplo que mostraremos agora ilustra um problema no qual Algencan encontra uma solução e Algencan-Global não encontra. Com este exemplo, verificamos que o problema de voracidade é mais prejudicial no processo de otimização global do que no processo de otimização local.

Problema E:

$$\begin{aligned} \min \quad & c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x^1 + c_0 \\ \text{s.a} \quad & x^2 = 1, \end{aligned}$$

onde $c_0 = 2$, $c_1 = 1.56$, $c_2 = -2$, $c_3 = -1.2916$, $c_4 = 0.5$ e $c_5 = 0.225$. Neste problema, temos um minimizador local $x^* = 1$ tal que $f(x^*) = 1$ e um minimizador global $x^{**} = -1$ tal que $f(x^{**}) = 0$.

Considerando o ponto inicial $x_0 = 2$, Algencan convergiu com 8 iterações para o ponto $x^* \approx 1$ cuja função objetivo vale aproximadamente 0.993, ou seja, Algencan convergiu para o mínimo local. Considerando a implementação Algencan-Global, observamos que esta encontra na primeira iteração um ponto bastante infactível com valores negativos. Logo, o método não consegue em iterações posteriores retirar seus iterados desta área infactível. Em Algencan-OTR-Global temos o seguinte desempenho. Inicialmente, $R_0 = 8.0\text{D}-01$ (este valor corresponde a versão escalada do problema com $s_f = 8.3\text{D}-02$ e $s_{h_1} = 2.5\text{D}-01$) e $\rho_1 = 1.0\text{D}+01$. Gencan começa sua execução com 100 aproximações iniciais distintas, obtendo o ponto x_1 tal que $\hat{f}(x_1) = -2.878904\text{D}+21$ e $R_1 = 5.0\text{D}+08$. Este ponto não é aceito como ponto de referência e uma nova iteração é realizada com $\rho_2 = 1.0\text{D}+02$ e $\Delta_2 = 21710.94$. Neste caso, o ponto obtido foi x_2 com $R_2 = 1.0\text{D}+08$. Novamente, o ponto é rejeitado e o raio da região de confiança passa a valer $\Delta_3 = 10855.47$. O ponto x_3 cuja função escalada vale aproximadamente $4.48\text{D}-03$ e $R_3 = 1.0\text{D}-03$ é aceito como ponto de referência e $\Delta_4 = \infty$. Com 4 iterações seguintes a solução global $x^{**} = -1$ é encontrada.

De certa forma, é natural que a voracidade seja mais eminente neste processo de minimização global do que o local. De fato, suponha que a distribuição dos pontos iniciais seja dada por p_1, \dots, p_6 . Além disso, suponha que o primeiro subproblema a ser resolvido seja dado por, digamos, $L_{10}(x, 0, 0)$. Se a função a ser minimizada ($L_{10}(x, 0, 0)$) tem forte decréscimo, é provável que a encontremos uma direção para este decréscimo em algum destes pontos p_1, \dots, p_6 e daí a voracidade pode ocorrer. A figura a seguir indica este comportamento.

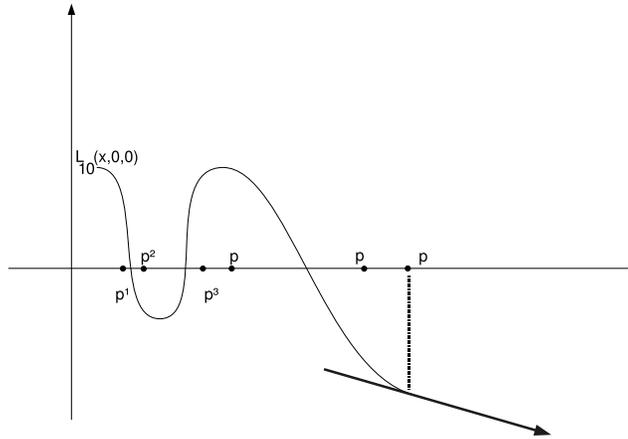


Figura 4.1: Aumento da probabilidade de ocorrer voracidade em minimização global.

4.2.3 Comparação dos resultados

Os problemas escolhidos para testar os métodos são encontrados na Coleção CUTer [15]. Uma parte dos testes é dedicada a minimização local e outra é dedicada a minimização global. Em ambos os casos nós tentamos verificar a influência da restrição de caixa adaptativa comparando o desempenho de Algencan com Algencan-OTR e Algencan-Global com Algencan-OTR-Global.

Para efeito de comparação, vamos fazer as seguintes considerações. Dado um problema fixo, para cada método M , denotamos por x_{final}^M ao ponto final obtido quando o método M é aplicado para resolver o problema dado. Dizemos que x_{final}^M é factível se

$$\max\{\|h(x_{final}^M)\|_\infty, \|g(x_{final}^M)\|_\infty\} \leq 10^{-8}.$$

Observe que, pela estrutura do conjunto simples e a forma pela qual os subproblemas são resolvidos, as restrições de caixa são sempre satisfeitas. Definimos

$$f_{best} = \min_M \{f(x_{final}^M) | x_{final}^M \text{ é factível}\}.$$

Dizemos que o método M encontrou a solução do problema se x_{final}^M é factível e

$$f(x_{final}^M) \leq f_{best} + 10^{-3}|f_{best}| + 10^{-6}.$$

Seja t^M o tempo de CPU que o método M utiliza para encontrar x_{final}^M . Definimos

$$t_{best} = \min_M \{t^M | \text{o método } M \text{ encontra a solução}\}.$$

Dizemos que o método M é *um dos métodos mais rápidos* para o problema se

$$t_M \leq t_{best} + 0.01t_{best}.$$

Estamos interessados em fazer comparações dos métodos com respeito a robustez e eficiência. Dizemos que um método é *robusto* para resolver o problema, quando este encontra a solução do problema. E dizemos este é *eficiente* quando é um dos métodos mais rápidos que resolve o problema.

Os perfis de desempenho [11] para as comparações que veremos, utilizam o tempo de CPU como medida de desempenho. Em termos de perfis de desempenho, eficiência e robustez são, essencialmente, os valores da função do perfil em 1 e ∞ respectivamente.

Para os experimentos, envolvendo minimização local, foram selecionados 736 problemas de programação não linear da Coleção CUTER. A Figura 4.2 mostra uma comparação entre Algencan e Algencan-OTR usando o perfil de desempenho e o tempo de CPU como medida de desempenho. Um tempo de CPU limite foi estabelecido nos testes, a saber, o tempo de CPU de cada problema em cada método não podia exceder 10 minutos. A eficiência de Algencan e Algencan-OTR foi, respectivamente, 78.26 % e 71.60%. Quanto a robustez dos métodos foi, 82.88% e 83.42%, respectivamente. Ambos os métodos encontraram pontos factíveis com valores de função objetivo equivalentes em 588 problemas. Note que, dizemos que f_1 e f_2 são equivalentes quando

$$|f_1 - f_2| \leq \max\{10^{-10}, 10^{-6} \min\{|f_1|, |f_2|\}\}$$

ou

$$f_1 \leq -10^{20} \text{ e } f_2 \leq -10^{20}.$$

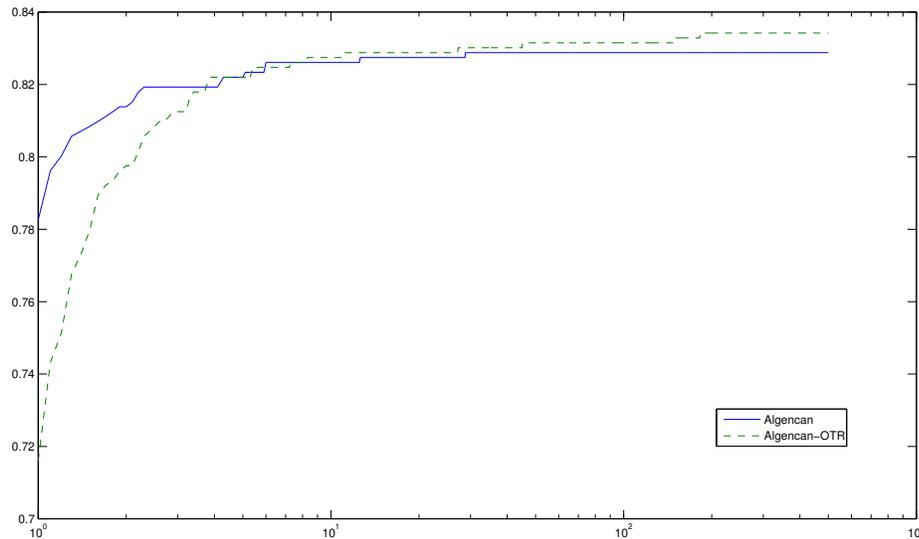


Figura 4.2: Comparação entre Algenca e Algenca-OTR usando o perfil de desempenho (em escala logarítmica).

Ambos os métodos falharam em encontrar um ponto factível em 100 problemas. Ambos os métodos encontraram ponto factível com valor de função objetivo diferente (não equivalente) em 34 problemas. Destes problemas, o valor de função objetivo encontrado por Algenca foi menor em 15 casos, enquanto o valor de função de função objetivo encontrado por Algenca-OTR foi menor nos outros 19 casos. Algenca encontrou um ponto factível em 7 problemas em que Algenca-OTR não encontrou. Contudo, Algenca-OTR também encontrou um ponto factível em 7 problemas em que Algenca não encontrou. Dentre estes 7 problemas em que apenas Algenca-OTR encontrou um ponto factível, somente em um problema Algenca detectou voracidade, a saber, no problema DITTERT. Dessa forma, verificamos que voracidade não é uma grande dificuldade para a versão Algenca 2.2.1 quando aplicada aos problemas da coleção CUTER.

Observação 4.2.1. *Versões mais antigas de Algenca, a saber, versões anteriores a março de 2007 são versões mais sensíveis aos problemas de voracidade, posto que estas não têm os mesmos recursos de escalamento e atualização de parâmetro de penalidade já citados*

anteriormente. Além disso, melhorias no algoritmo de resolução do subproblemas, Gencan, pode refletir, independentemente da questão de voracidade, melhores resultados.

Para os experimentos de minimização global, selecionamos 260 problemas de programação não linear da Coleção CUTEr com, no máximo, 10 variáveis. A Figura 4.3 mostra uma comparação entre Algencan-Global e Algencan-OTR-Global usando perfil de desempenho. O tempo de 30 minutos foi estabelecido como tempo limite para cada problema em cada método testado. A eficiência de Algencan-Global e Algencan-OTR-Global foi de 89.92% e 88.46%, respectivamente. Quanto a robustez, o resultado foi 95.38% para ambos.

Em 244 problemas, ambos os métodos encontraram o mesmo mínimo. Também, em 8 problemas ambos os métodos pararam em pontos inactivéis. Assim, os dois métodos apresentaram comportamentos diferentes em 8 problemas. A tabela a seguir, mostra alguns detalhes destes problemas. Nesta tabela, $f(x^*)$ e $R(x^*)$ representam o valor da função objetivo e a medida de factibilidade-complementariedade, respectivamente. SC é o critério de parada, isto é: C significa convergência, T quer dizer que o tempo limite de CPU foi alcançado e I significa que o parâmetro de penalidade está muito grande. Basicamente, temos o seguinte:

- (i) Ambos os métodos encontraram diferentes minimizadores globais em 5 problemas,
- (ii) Algencan-OTR-Global encontrou um ponto factível em um problema (HS107) em que Algencan-Global não conseguiu.
- (iii) Algencan-OTR-Global encontrou a solução em dois problemas em Algencan-Global apresentou voracidade.

Concluindo a análise de robustez, nós podemos dizer que Algencan-OTR-Global foi bem sucedido ao encontrar a solução em dois problemas que apresentaram voracidade HS24 e HS56 (veja [17] para formulação destes problemas). Entretanto, esta vantagem foi compensada pelo fato de Algencan-Global encontrar 4 minimizadores melhores do Algencan-OTR-Global nos 6 problemas que ambos resolveram. Contudo, nestes problemas, não podemos prever se, de fato, isto é uma consequência relacionada a redução das caixas e dos pontos aleatórios gerados nestas afim de obter as soluções globais dos subproblemas. Nos problemas DIXCHLNG e SNAKE ambos os métodos satisfazem os critérios de parada “C” e

então, claramente, ambos os métodos convergem à diferentes soluções locais (em um caso, Algencan-Global tem melhor solução e em outro Algencan-OTR-Global tem melhor). Nos quatro casos restantes, ambos métodos param por obter um parâmetro de penalidade muito alto ou pelo fato do tempo de CPU ter sido alcançado.

Problema	Algencan-Global			Algencan-OTR-Global		
	$f(x^*)$	$R(x^*)$	SC	$f(x^*)$	$R(x^*)$	SC
CRESC50	5.9339763626815067E-01	0.0E+00	T	5.9357981462855491E-01	1.4E-09	T
DIXCHLNG	1.6288053006804543E-21	8.9E-13	C	4.2749285786956551E+02	7.8E-13	C
EQC	-1.0380294895991835E+03	1.0E-10	T	-1.0403835102461048E+03	1.0E-10	T
HS107	5.0549933321413228E+03	1.0E-08	I	5.0550117605040141E+03	1.2E-09	T
HS24	-1.9245010614395141E+59	1.7E+20	I	-1.0000000826918698E+00	9.6E-12	C
HS56	-9.999999999999995E+59	1.0E+20	I	-3.455999999999844E+00	6.9E-14	C
QC	-1.0778351725254695E+03	0.0E+00	T	-1.0776903884481490E+03	1.0E-10	T
SNAKE	2.9758658959064692E-09	0.0E+00	C	-7.0445051551086390E-06	3.7E-10	C

Tabela 4.1: Informações sobre os 8 problemas em que Algencan-Global e Algencan-OTR-Global mostraram desempenho diferentes.

Os problemas em que ambos os métodos pararam em pontos infactíveis foram: ARGAUSS, CRESC132, CSFI1, ELLATTAR, GROWTH, HS111LNP, TRIGGER e YFITNE.

Observamos ainda que, Algencan (sem a estratégia de Algencan-Global) obteve sucesso ao resolver HS24 e HS56, evidenciando que o problema de voracidade está, nestes dois casos, diretamente relacionado ao a estratégia de globalização (como visto no problema E).

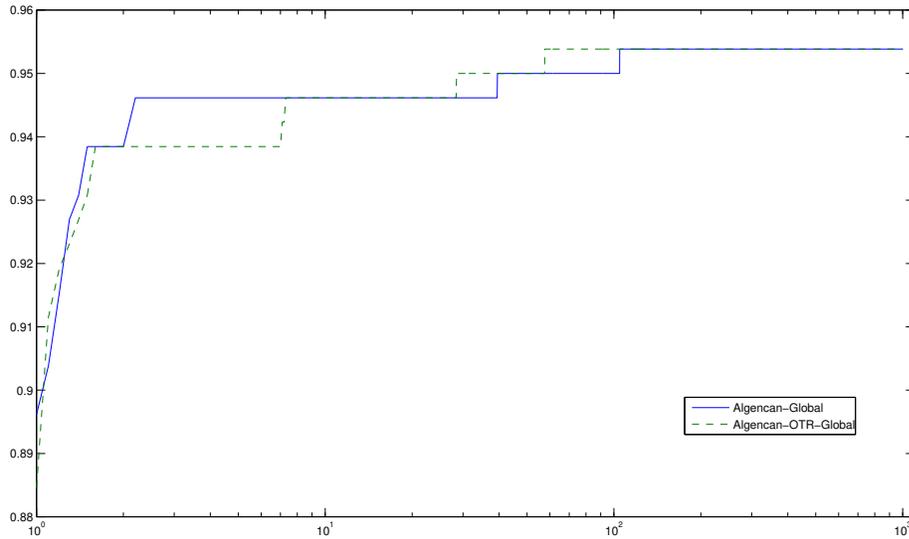


Figura 4.3: Comparação entre Algenca-Global e Algenca-OTR-Global usando o perfil de desempenho (em escala logarítmica).

4.3 Comparação entre Algenca e RAlgenca

Estamos interessados em testar a confiabilidade do método de Lagrangiano Aumentado com Região de Confiança apresentado anteriormente. Considerando o fato de que a versão regularizada deste método visa os mesmos objetivos é interessante compararmos as duas versões, ou seja, compararmos Algenca-OTR com RAlgenca com a finalidade de determinar qual estratégia é melhor na prática. Antes porém, vamos verificar a confiabilidade do método regularizado. Desta forma, esta seção dedica-se a uma comparação da implementação do método apresentado no capítulo 2 (RAlgenca) com a supracitada implementação Algenca. Note que, estaremos apenas comparando eficiência e robustez nesta seção. Uma descrição de desempenho em problemas que apresentam voracidade será dada na próxima seção.

Para a implementação RAlgenca atualizamos γ_{k+1} como em [9], isto é:

- Se $R_k = \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$ definimos $\gamma_{k+1} = 0$.

- Senão, definimos

$$\gamma_{k+1} = \min\{\beta R_k, \gamma_k + 1\}. \quad (4.3)$$

Considerando novamente os 736 problemas de programação não-linear da coleção CUTER temos que a eficiência de Algencan e RAlgencan foram, respectivamente, 81.25 % e 67.25 %. Quanto a robustez dos métodos temos 83.55% para Algencan e 81.25% para RAlgencan.

Ambos os métodos encontraram pontos factíveis com valor de função objetivo equivalentes em 581 problemas e falharam em encontrar pontos factíveis em 102 problemas. Em 33 problemas os métodos encontraram pontos factíveis com valor de função objetivo diferentes. Algencan encontrou pontos factíveis com valor de função objetivo menor em 19 problemas enquanto a versão regularizada encontrou pontos factíveis com valor de função objetivo menor em 14 problemas. Em 15 problemas, Algencan encontrou pontos factíveis e RAlgencan não. RAlgencan encontrou ponto factível em 5 problemas nos quais Algencan não encontrou. São eles: DITTERT, KISSING2, QR3D, ROCKET e SNAKE. Observe que DITTERT apresentou voracidade, indicando o benefício da regularização neste caso.

Estes resultados indicam que a implementação RAlgencan apresenta desempenho um pouco inferior a implementação Algencan. Mesmo assim, é possível verificar que esta estratégia de regularização é confiável pois, sua robustez ficou próxima da robustez de Algencan. Para analisar o perfil de desempenho, veja a Figura 4.4.

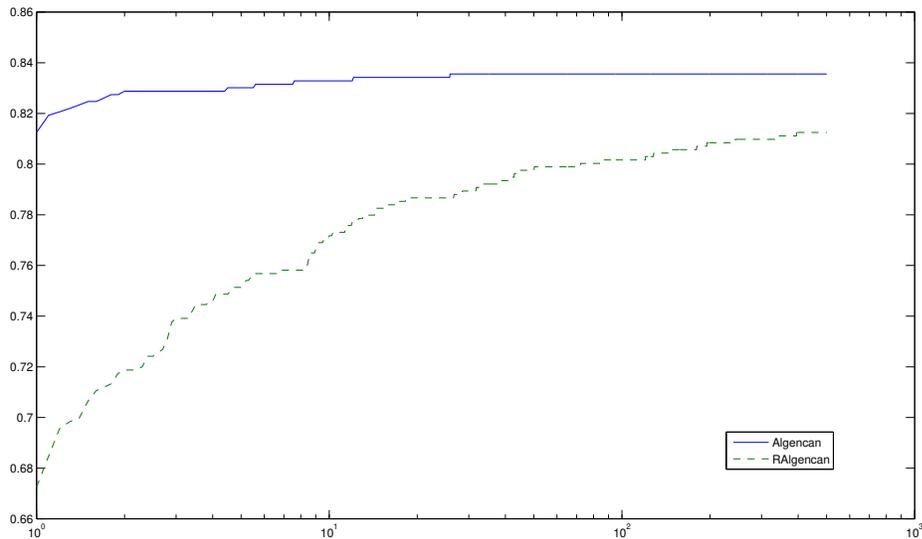


Figura 4.4: Comparação entre Algencan e RAlgencan usando o perfil de desempenho (em escala logarítmica).

Verificaremos agora o comportamento de RAlgencan para problemas de minimização global. Para este fim, vamos considerar os mesmos 260 problemas usados na comparação de Algencan-OTR-Global e Algencan-Global. Estaremos nos referindo ao método RAlgencan com estratégia para minimização global por RAlgencan-Global.

Quanto a robustez, Algencan-Global e RAlgencan-Global obtiveram 95.38%. Com relação a eficiência, Algencan-Global obteve 91.15% e RAlgencan-Global obteve 86.54% (veja o perfil de desempenho na Figura 4.5). As duas implementações encontraram pontos factíveis com valor de função objetivo equivalentes em 244 problemas. RAlgencan-Global encontrou solução factível em 2 problemas que Algencan-Global não encontrou, tais problemas são HS24 e HS56 que apresentam voracidade. Contudo, Algencan-Global encontrou um ponto factível em um problema que RAlgencan não obteve sucesso, HS87. Em 5 problemas os métodos encontram pontos factíveis com valor de função objetivo não equivalentes. Em 3 destes problemas (CRESC50, HS107 e QC) Algencan-Global encontrou ponto com menor valor de função objetivo. Enquanto nos 2 restantes (EQC e SNAKE), RAlgencan-Global

encontrou solução melhor.

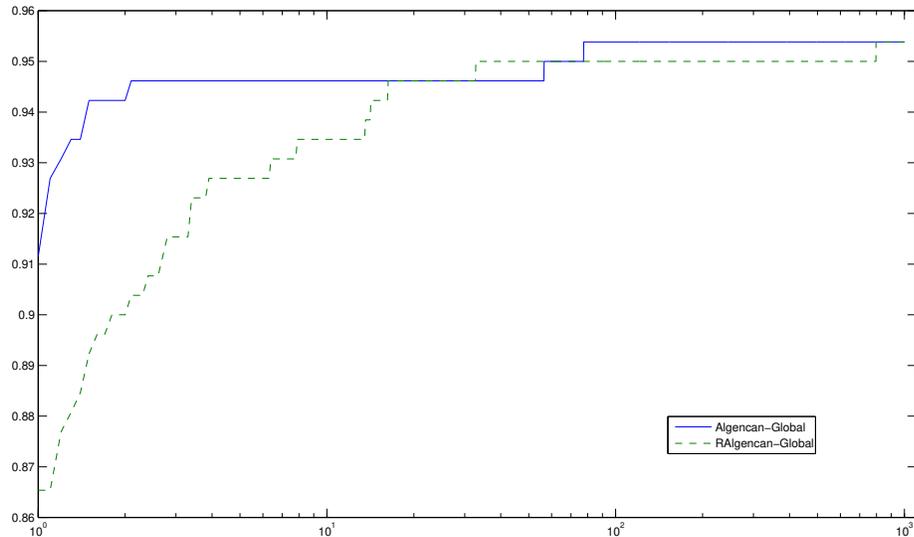


Figura 4.5: Comparação entre Algenca-Global e RAlgenca-Global usando o perfil de desempenho (em escala logarítmica).

4.4 Comparação entre Algenca-OTR e RAlgenca

Nesta seção vamos considerar a citada implementação Algenca-OTR e a implementação do método de Lagrangiano Aumentado Regularizado apresentado no capítulo 2, RAlgenca.

Primeiramente, vamos analisar o comportamento de RAlgenca em problemas que apresentem voracidade, isto é, vamos considerar os problemas A, B, C e D. Estaremos considerando os mesmos pontos iniciais vistos na análise de Algenca-OTR.

Problema A. A primeira iteração de RAlgenca e Algenca-OTR coincidem pois o parâmetro de regularidade inicial (γ_1) vale 0. Assim como em Algenca-OTR um iterando apenas é aceito como novo ponto de referência quando $R_k \leq R_{tol}$. Desta forma, o primeiro iterando é rejeitado por RAlgenca e o parâmetro de regularização passa a valer 1. Com este parâmetro de regularização o iterando obtido na segunda iteração ainda não pode ser aceito como novo

ponto de referência e apenas na terceira iteração, usando $\gamma_3 = 2$ o ponto obtido é aceito como ponto de referência, neste caso, $\hat{f}(x^3) = -1.5267D+00$, $R_3 = 0.0D+00$ e $\rho_3 = 1.0D+02$. Contudo, na iteração seguinte o parâmetro de regularização passa a valer 0 e, novamente, ocorre voracidade. Desta forma, com $\gamma_5 = 1$, o iterando obtido não é aceito como novo ponto de referência. Com $\gamma_6 = 2$ o iterando x^6 obtido é tal $\hat{f}(x^6) \approx -3.164846D+00$ e $R_6 = 1.D-03$. γ_7 passa a valer aproximadamente 1.0669383341384944. Note que, como o iterando x^3 é factível, um iterando seguinte só é aceito como novo ponto de referência quando este também for factível. Desta forma, o método vai penalizando cada vez mais e a sequência dos valores de γ é (1.31, 0.12, 0.16, 1.16, 2.16, 1.03E-004, 1.00, 1.33E-005, 1.00, 2.00, 0.00). Observe que na iteração 17, usando $\gamma_{17} = 2.00$ o método consegue encontrar um ponto factível, contudo, o parâmetro de penalidade neste momento é muito grande, a saber, 1.0D+09 e o método não consegue decretar convergência pois a norma do gradiente projetado vale aproximadamente 4.D-05. Em seguida, com $\gamma_{18} = 0$ ocorre voracidade novamente. O método não consegue convergir.

Problema B. Assim como em Algencan-OTR, o primeiro iterando obtido por RAlgencan, usando $\gamma_1 = 0$ não é aceito como novo ponto de referência. Com $\gamma_0 = 1$ o ponto obtido tem medida de complementaridade-factibilidade de valor 6.0D-03 e é aceito como novo ponto de referência. O valor do parâmetro de regularização passa a valer 0 e com este valor o método consegue decretar a convergência com 9 iterações externas e para o mesmo ponto obtido por Algencan-OTR.

Problema C. Assim como o que aconteceu em Algencan-OTR o ponto x^1 é rejeitado e conseqüentemente o parâmetro de regularização para a próxima iteração passa a valer 1. Com este parâmetro, o método encontra um ponto x^2 com $R_2 = 1.0D-03$ e $\hat{f}(x^2) = 1.606981D-01$. O iterando é aceito como ponto de referência e o parâmetro de regularização para a iteração seguinte passa a valer 0 e $\rho_2 = 1.0D+02$. Ocorre voracidade novamente, o ponto não é aceito como novo ponto de referência e $\gamma_4 = 1$. Novamente, é obtido um iterando x^4 com $R_4 = 4.D-06$. Este ponto é aceito como ponto de referência o parâmetro de regularização passa a valer 0, ocorre voracidade e esta situação se repete por algumas iterações. O método oscila os valores dos parâmetros de regularização entre 0 e 1 e enquanto isso os parâmetros de penalidade

aumentam demasiadamente. O método não consegue convergir.

Problema D. O desempenho neste problema foi muito semelhante com Algencan-OTR. Inicialmente $\gamma_1=0$. A primeira iteração foi rejeitada, pois esta coincide com a primeira iteração de Algencan-OTR e o valor do parâmetro de regularização passou a ser $\gamma_2=1$. Com este valor o método encontra um ponto cuja medida de complementariedade-factibilidade é igual a $3.0D-02$. Este ponto é aceito como novo ponto de referência e o parâmetro de regularização passa a valer 0. Usando este valor para os parâmetros de regularização seguintes o método converge com o mesmo número de iterações que Algencan-OTR.

Novamente, para os experimentos, foram considerados 736 problemas de programação não linear da coleção CUTEr. A eficiência de RAlgencan e Algencan-OTR foram, respectivamente, 67.52% e 81.11%. Quanto a robustez os resultados indicaram 81.65% para RAlgencan e 84.10% para Algencan-OTR.

Em 589 problemas as duas implementações encontraram pontos factíveis com valor de função objetivo equivalentes. Ambos encontraram pontos infactíveis em 105 problemas. Em 12 problemas Algencan-OTR encontrou ponto factível e RAlgencan não. Em outros 2 problemas RAlgencan encontrou ponto factível e Algencan-OTR não. Em 28 problemas os métodos encontraram pontos de mínimos diferentes (não equivalentes). Destes problemas, Algencan-OTR encontrou um ponto factível com valor de função objetivo melhor em 18 problemas e nos 10 problemas restantes RAlgencan encontrou um ponto com menor valor de função objetivo. Observe que, Algencan-OTR apresentou melhor desempenho tanto em eficiência como em robustez (veja a Figura 4.6). A razão principal para esta discrepância deve-se à atualização de γ , isto é, em Algencan-OTR relaxamos a atualização da região de confiança ao fato de que apenas quando $R_k > 100 \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$ nós diminuimos o raio Δ_{k+1} , entretanto, para atualização de γ_{k+1} não impomos este fato, ou seja, se $R_k > \min\{\max\{R_{tol}, R_0\}, R_1, \dots, R_k\}$ então aumentamos o valor de γ_{k+1} . Tal fato, pode ser na prática um pouco restritivo.

Para a minimização global, vamos começar por analisar o problema E. Neste problema RAlgencan se comportou de maneira parecida com Algencan-OTR. O iterando x_1 obtido é rejeitado pois este é muito infactível. Assim, o parâmetro de regularização passa a valer

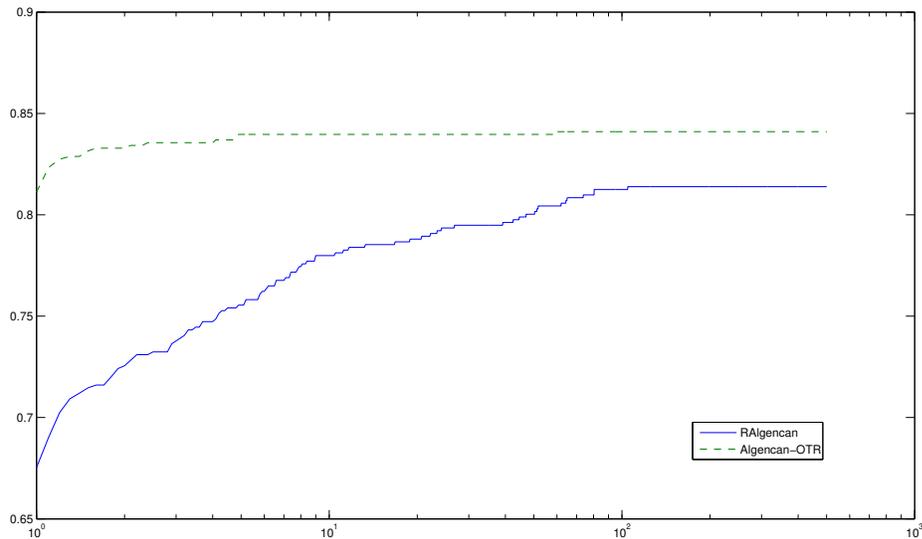


Figura 4.6: Comparação entre RAlgencan e Algencan-OTR usando o perfil de desempenho (em escala logarítmica).

1 e o parâmetro de penalidade $\rho_2 = 1.0\text{D}+02$. O ponto x_2 obtido é tal que $R_2 = 2.0\text{D}-02$. Este ponto é aceito como novo ponto de referência e o parâmetro de regularização passa a valer 0. Na iteração seguinte, novamente o iterando é aceito como novo ponto de referência pois $R_2 = R_3$ e $\rho_4 = 1.0\text{D}+03$. Contudo, uma iteração depois, ocorre voracidade novamente e o iterando x_4 é rejeitado, o parâmetro de penalidade é atualizado ($\rho_5 = 1.0\text{D}+04$) e o termo regularizador passa a valer 1. Com estes parâmetros o iterando x_5 com $R_5 = 1.0\text{D}-05$ é aceito como novo ponto de referência e $\gamma_6 = 0$. Em 2 iterações seguintes o método decreta a convergência para um minimizador global $x^{**} = -1$.

Para fazer uma comparação entre Algencan-OTR e RAlgencan para otimização global vamos, novamente, considerar 260 problemas da coleção CUTER (os mesmos escolhidos anteriormente). Quanto a eficiência de RAlgencan-Global e Algencan-OTR-Global os resultados foram, respectivamente, 84.23% e 93.46%. Com relação a robustez temos 95% para RAlgencan-Global e 96.15% para Algencan-OTR-Global. Os dois métodos encontraram pontos factíveis com valor de função objetivo equivalentes em 245 e em 8 problemas ambos

os métodos não encontraram ponto factível. Algencan-OTR-Global encontrou um ponto factível em um problema em que RAlgencan-Global não encontrou, a saber, HS87. Em 6 problemas os métodos encontraram pontos factíveis com valor de função objetivo diferentes. Destes, o valor encontrado por RAlgencan-Global foi menor em 2 problemas. São eles: DIXCHLNG e EQC. Nos 4 problemas restantes (CRESC50, HS107, QC e SNAKE), Algencan-OTR-Global encontrou pontos com valor de função objetivo menor (estes dados são expressos na Tabela 4.2). Desta forma, concluímos que método Algencan-OTR é menos prejudicial que a estratégia com termo regularizador em problemas de otimização global. O perfil de desempenho é fornecido na Figura 4.7.

Problema	RAlgencan-Global			Algencan-OTR-Global		
	$f(x^*)$	$R(x^*)$	SC	$f(x^*)$	$R(x^*)$	SC
CRESC50	5.9717435587492662E-01	0.0E+00	T	5.9357981462855491E-01	1.4E-09	T
DIXCHLNG	1.0726116603070267E-23	7.8E-14	C	4.2749285786956551E+02	7.8E-13	C
EQC	-1.0428378767648676E+03	2.2E-09	T	-1.0403835102461048E+03	1.0E-10	T
HS107	1.2312842658704827E+04	2.6E-15	T	5.0550117605040141E+03	1.2E-09	T
QC	-1.0749824312633964E+03	1.0E-10	T	-1.0776903884481490E+03	1.0E-10	T
SNAKE	-6.3663488406867719E-10	3.2E-14	C	-7.0445051551086390E-06	3.7E-10	C

Tabela 4.2: Informações sobre os 8 problemas em que Algencan-Global e Algencan-OTR-Global mostraram desempenho diferentes.

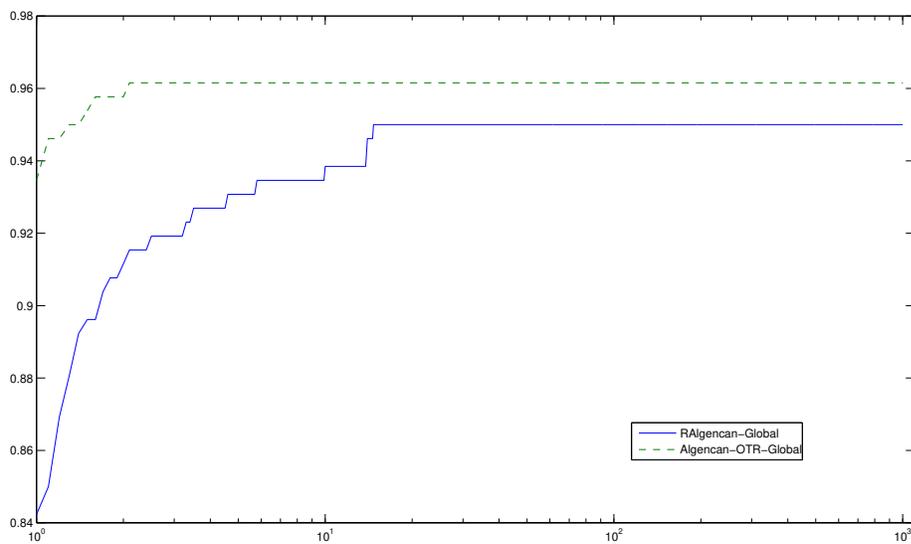


Figura 4.7: Comparação entre Algenca-OTR-Global e RAlgenca-Global usando o perfil de desempenho (em escala logarítmica).

CAPÍTULO 5

Conclusões

Na presente tese, apresentamos um método do tipo Lagrangiano Aumentado. Em verdade, a motivação para o novo método é dada devido a uma dificuldade prática em aplicar métodos deste tipo, a saber, o fenômeno de voracidade.

Naturalmente, na área de otimização, podemos ter diversos algoritmos que, possivelmente, resolvem os problemas que apresentam este fenômeno. Afinal, estes problemas são peculiares do método de Lagrangiano Aumentado. Entretanto, por acreditar nas boas características deste tipo de método e, em especial, pelos bons resultados que a implementação Algencan apresenta, decidimos explorar este método e esta implementação.

Evidentemente, muitas soluções para esta patologia podem ser dadas. Contudo, métodos que apresentam bons resultados teóricos e que, em sua prática, resolvem as patologias sem agredir o bom desempenho numa quantidade razoável de problemas, são complicados de serem obtidos. Um método que almeja estes requisitos é o método de Lagrangiano Aumentado Regularizado. Daí, a relevância das comparações apresentadas.

A nova estratégia, isto é, a inserção de restrições de caixa adaptativas para remediar o problema de voracidade não afeta a rica teoria que existe no método de Lagrangiano Aumentado, ou seja, teoremas de convergência e limitação de parâmetro de penalidade foram

obtidos mantendo hipóteses bastante razoáveis. Assim, tanto o Lagrangiano Aumentado Regularizado como o Lagrangiano Aumentado com Região de Confiança são, do ponto de vista teórico, confiáveis.

Já do ponto de vista prático, conseguimos, usando como base a implementação Algencan, obter uma implementação confiável e eficiente, Algencan-OTR. Tal implementação, é bastante promissora posto que os testes numéricos realizados indicam um bom desempenho, isto é, em problemas gerais de otimização não linear Algencan-OTR se apresentou comparável a implementação Algencan. Assim, verificamos que as restrições de caixa utilizadas, em geral, não comprometem a obtenção de solução e além disso, testamos a confiabilidade da nova estratégia. Evidentemente existem exceções, ou seja, existem problemas que Algencan resolve e Algencan-OTR não (e também o contrário), como foi citado no capítulo anterior. Contudo, Algencan-OTR tem a vantagem de ser menos vulnerável ao efeito de voracidade, gerando assim um pouco mais de robustez. Nos problemas de minimização global, novamente a implementação Algencan-OTR (Algencan-OTR-Global) se mostrou promissora e seus resultados foram equivalentes aos de Algencan (Algencan-Global). Mais do que isso, foi verificado que este problema de voracidade é mais comum quando se usa uma estratégia de minimização global como a apresentada neste trabalho e também nestes casos a implementação Algencan-OTR se mostrou confiável. Dessa forma, verificamos que a nova estratégia é, em sua prática, robusta e eficiente. Além disso, considerando a implementação RAlgencan, verificamos que Algencan-OTR se mostrou superior, indicando que a o uso de região de confiança é melhor que o uso do termo regularizador. Evidentemente, estamos considerando a atualização do termo regularizador como descrito em [9] que é um pouco mais restritiva do que a atualização da região de confiança proposta neste trabalho. Possivelmente, uma nova atualização para o termo regularizador pode fornecer um desempenho superior para RAlgencan.

BIBLIOGRAFIA

- [1] R. Andreani, E. G. Birgin, J. M. Martínez e M. L. Schuverdt, On Augmented Lagrangian Methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286-1309, (2007).
- [2] R. Andreani, J. M. Martínez e M. L. Schuverdt, On the relation between the constant positive linear dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications*, 125, pp. 473-485, (2005).
- [3] M. Bazaraa, H. Sherali e C. M. Shetty, “Nonlinear programming: theory and algorithms”, 2^a edição, Nova Iorque, Wiley, 1993.
- [4] D. P. Bertsekas, “Nonlinear Programming”, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.
- [5] D. P. Bertsekas, “Constrained Optimization and Lagrange Multiplier Methods”, Academic Press, 1982; republished by Athena Scientific, 1996.
- [6] E. G. Birgin, R. Castillo e J. M. Martínez. Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems, *Computational Optimization and Applications* 31, pp. 31-56 (2005).

- [7] E. G. Birgin e J. M. Martínez, Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications*, 23, pp. 101-125, 2002.
- [8] E.G. Birgin, E.V. Castelani, A.L. Martinez e J.M. Martínez, “Outer Trust-Region method for Constrained Optimization”, *submetido*, 2009.
- [9] E.V. Castelani, A.L. Martinez, J.M. Martínez e B. F. Svaiter, Addressing the greediness phenomenon in nonlinear programming by means of proximal augmented Lagrangians. Por aparecer em *Computational Optimization and Applications*.
- [10] A. R. Conn, N. I. M. Gould e Ph. L. Toint, Trust Region Methods, *MPS/SIAM Series on Optimization*, SIAM, Philadelphia, 2000.
- [11] E. D. Dolan e J. J. Moré. Benchmarking optimization software with performance profiles, *Mathematical Programming* 91, pp. 201-213 (2002).
- [12] A. V. Fiacco e G.P. McCormick, “Nonlinear Programming: Sequential Unconstrained Minimization Techniques”, John Wiley & Sons, New York, 1968.
- [13] R. Fletcher, “Practical Methods of Optimization”, Academic Press, London, 1987.
- [14] R. Fletcher, “Practical methods of optimization”, 2ª edição, Nova Iorque, Wiley, 1987.
- [15] N. I. M. Gould, D. Orban e Ph. L. Toint, CUTeR e SifDec: A Constrained and Unconstrained Testing Environment, revisited, *ACM Transactions on Mathematical Software* 29, pp. 373-394, 2003.
- [16] M. R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications*, 4, pp. 303-320, 1969.
- [17] W. Hock e K. Schittkowski, Test examples for nonlinear programming codes, *Lecture Notes in Economics and Mathematical Systems* 187, Springer-Verlag, Berlin, Heidelberg, New York, 1981.

- [18] D. G. Luenberger, “Linear and nonlinear programming”, 2^a edição, Reading, Addison-Wesley, 1984.
- [19] J. M. Martínez, “Otimização Prática usando o Lagrangiano Aumentado”, Departamento de Matemática Aplicada, Universidade Estadual de Campinas, Campinas, Brasil, 2006.
- [20] J. Nocedal e S. J. Wright, “Numerical optimization ”, Nova Iorque, Springer, 1999.
- [21] M. J. D. Powell, A method for nonlinear constraints in minimization problems, in Optimization, R. Fletcher (ed.), Academic Press, New York, NY, pp. 283-298, 1969.
- [22] L. Qi e Z. Wei, On the constant positive linear dependence condition and its application to SQP methods, *SIAM Journal on Optimization*, 10, pp. 963-981, 2000.
- [23] R. T. Rockafellar, A dual approach to solving nonlinear programming problems by unconstrained optimization, *Mathematical Programming*, 5 , pp. 354-373, 1973.
- [24] R. T. Rockafellar, Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control and Optimization*, 12, pp. 268-285, 1974.
- [25] R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Mathematics of Operations Research* 1, 97-116, 1976.
- [26] R. T. Rockafellar, Lagrange multipliers and optimality, *SIAM Review*, 35, pp. 183-238, 1993.
- [27] M. L. Schuverdt, Métodos de Lagrangiano aumentado com convergência utilizando a condição de dependência linear positiva constante, *Tese de Doutorado em Matemática Aplicada*, UNICAMP.
- [28] <http://www.ime.usp.br/~egbirgin/tango/>.