UM NOVO ALGORÍTMO DE PENALIZAÇÃO HIPERBÓLICA PARA RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO NÃO-LINEAR COM RESTRIÇÕES DE IGUALDADES

Turíbio José Gomes dos Santos

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO .

Aprovada por :

Prof. Adilson Elias Xavier , D. Sc.

Prof. Nelson Maculan Filho , D. Habil.

Prof. Luis Alfredo V. de Carvalho , D. Sc.

Prof. Roberto Quirino do Nascimento , D. Sc.

Prof. Sérgio Granville , Ph. D.

Prof. Luiz Satoru Ochi , D. Sc.

RIO DE JANEIRO - RJ - BRASIL

ABRIL DE 1998 .

SANTOS , TURÍBIO JOSÉ GOMES DOS

Um novo algorítmo de penalização hiperbólica para resolver o problema de programação não-linear sujeito a restrições de igualdades . [Rio de Janeiro] 1998

XIII , p. 29,7 cm (COPPE / UFRJ, D.Sc. , Engenharia de Sistemas e Computação 1998)

Tese - Universidade Federal do Rio de Janeiro , COPPE

1. Otimização , 2 . Programação Não-Linear 3 . Método de Penalidade , 4 . Lagrangeano.

I. COPPE / UFRJ II. Título (série)

A minha esposa

Ana Elvira e a

meus filhos ,

Cristhiano

Andréa e

Raíssa

AGRADECIMENTOS

Ao Programa de Engenharia de Sistemas e Computação da COPPE / UFRJ pelo apoio na realização deste doutorado .

Aos professores Paulo Roberto , Susana Scheimberg , Clovis Gonzaga , Adilson Xavier, pelos cursos teóricos básicos que deram sustentação à elaboração deste trabalho.

Ao professor Adilson Xavier , a quem muito admiro pelo espírito de pesquisa, agradeço profundamente a sua confiança , solidariedade , amizade , e pela riqueza e abrangência de sua orientação.

Ao professor Roberto Quirino , pelo incentivo , apoio , compreensão e amizade , demonstrados em todos os momentos .

A minha esposa Ana Elvira , aos meus filhos Cristhiano , Andréa e Raíssa , pelo incentivo , colaboração , compreensão e amizade . O apoio de todos foi muito importante para o término deste trabalho.

Aos meus pais Avelino e Lôide , muito presentes e solidários me incentivando e dando forças para a elaboração deste trabalho .

Aos meus irmãos e minhas irmãs Ana Clara e Lélia , pelo incentivo , força, compreensão e amizade , desde o início deste trabalho .

Aos meus colegas da UFPb e em particular aos professores do Departamento de Matemática da UFPb-JP, pela confiança , incentivo e apoio em todos os momentos.

Aos meus colegas da COPPE / UFRJ , por estarem sempre presentes e prontos a ajudar em qualquer problema , agradeço por suas amizades .

A CAPES / PICD , pelo apoio financeiro durante todo o período de estudos e realização desta Tese.

Resumo da Tese apresentada à COPPE / UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D. Sc.).

UM NOVO ALGORÍTMO DE PENALIZAÇÃO HIPERBÓLICA PARA RESOLVER O PROBLEMA DE PROGRAMAÇÃO NÃO-LINEAR SUJEITO A RESTRIÇÕES DE IGUALDADES

Turíbio José Gomes dos Santos

Abril de 1998 .

Orientador : Adilson Elias Xavier

Programa : Engenharia de Sistemas e Computação

O presente trabalho mostra um novo algorítmo para resolução do problema de programação não-linear com restrições de igualdades. Para isso , se utiliza do método de penalização hiperbólica , originalmente desenvolvido para resolução de problemas de programação não-linear com restrições de desigualdades.

Através da consolidação de resultados previamente estabelecidos , apresentamos ademais, um outro algorítmo que contempla a resolução dos problemas de programação não-linear simultaneamente com restrições de igualdades e desigualdades.

O desempenho computacional desses algorítmos é ilustrado através da resolução de um conjunto de problemas -teste da bibliografia.

Abstract of Thesis presented to COPPE / UFRJ as partial fulfillment of the requirments for the degree of Doctor of Science (D. Sc.).

A NEW ALGORITHM OF HIPERBOLIC PENALIZATION TO

TO SOLVE THE NONLINEAR PROGRAMMNIG PROBLEMS

SUBJECT TO EQUALITY CONSTRAINTS

Turíbio José Gomes dos Santos

Abril de 1998.

Chairman: Adilson Elias Xavier

Department : Systems Engineering and Computer Sciences

This work presents a new algorithm in order to solve the nonlinear programming problems with equality constraints . To achieve this proposal , it is used the hyperbolic penalty method , which was originally developed to solve the nonlinear programming problems with inequality constraints.

By the consolidation of previously established results , it is presented additionally another algorithm which considers the solution of the nonlinear prograqmming problems with equality and inequality constraints simultaneously.

The computational performance of these algorithms is exhibited through the solution of a set of test-problems extracted from the literature.

ÍNDICE

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

CAPÍTULO 4

CAPÍTULO 5

ÍNDICE DAS FIGURAS

ÍNDICE DAS TABELAS

NOTAÇÕES

$f(x)$	Função objetivo
$F(x, \alpha, \tau)$	Função objetivo aumentada ou modificada
$g_i(x)$	Funções restrições de desigualdades
$h_j(x)$	Funções restrições de igualdades
i	Índice especificando restrição de desigualdade
j	Índice especificando restrição de igualdade
k	Número da iteração
$l(x, .)$	Função Lagrangeana
$L(x, ., .)$	Função Lagrangeana aumentada ou modificada
$L_H(y, \alpha, \tau)$	Função Lagrangeana hiperbólica
m	Número de restrições de desigualdades
n	Dimensão do espaço Euclidiano
p	Número de restrições de igualdades
$P(y, \alpha, \tau)$	Função de penalidade hiperbólica
R^n	Espaço Euclidiano n - dimensional
S	Conjunto solução do problema com restrições
$S\varepsilon$	Conjunto solução do problema relaxado
x	Variável no espaço R^n
x^k	Ponto ótimo da função objetivo modificada na iteração k
x^o	Ponto inicial

x^*	Ponto ótimo ou solução do problema original
α	Parâmetro ângulo da penalização hiperbólica
ε_j^+	Relaxação ou folga superior
ε_j^-	Relaxação ou folga inferior
λ	Multiplicador de Lagrange relativo às restrições de desigualdades
μ	Multiplicador de Lagrange relativo às restrições de igualdades
τ	Parâmetro distância da penalização hiperbólica

CAPÍTULO 1

INTRODUÇÃO

Consideraremos neste trabalho a resolução do problema geral de programação não-linear sujeito a restrições de igualdades e desigualdades da forma:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito à:} && g_i(x) \geq 0 \quad , \ i = 1, \dots, m \\ & && h_j(x) = 0 \quad , \ j = 1, \dots, p \end{aligned} \tag{1}$$

onde $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$ e $h : R^n \rightarrow R^p$.

Em particular, consideraremos a resolução do problema de programação não-linear sujeito a restrições de igualdades da forma :

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito à:} && h_j(x) = 0 \quad , \ j = 1, \dots, p \end{aligned} \tag{2}$$

onde $f : R^n \rightarrow R$ e $h : R^n \rightarrow R^p$.

Primeiramente apresentaremos um novo método para sua resolução . Esse método se fundamenta na utilização da penalização hiperbólica , que na sua abordagem inicial trata da resolução de problemas unicamente com restrições de desigualdades e com ademais , a hipótese de possuir região viável com interior não-vazio.

O novo método basicamente se constitui na relaxação das retrições de igualdades, transformadas em desigualdades , através de folgas superiores ε^+ e inferiores ε^-, criando assim interior não-vazio , e permitindo desse modo , a utilização da penalização hiperbólica .

Basicamente , seguindo essas idéias e incorporando a imprescindível manipulação conveniente das folgas , apresentamos o algorítmo I para resolução do problema (2).

Para estabelecermos o estudo teórico de convergência do método , relacionamos primeiramente um conjunto de resultados subsidiários a esse estudo . Além disso , especificamos um conjunto de condições que o problema (2) necessita satisfazer. A seguir são desenvolvidos resultados que mostram a convergência do algorítmo I.

Com o objetivo de validar o desempenho do algorítmo I , efetuamos um conjunto de experimentos computacionais se utilizando da referência de Hock e Schittkowiski [34] , seguindo as recomendações de JACKSON et al. [36] e consentâneo a outros trabalhos publicados na literatura.

Os resultados obtidos se comportaram conforme a teoria previamente desenvolvida , demonstrando robustez e eficiência frente à totalidade dos problemas testes efetuados em nossos experimentos.

Através da consolidação dos resultados previamente estabelecidos e daqueles descritos em Xavier [62] , estendemos nossos estudos à resolução do problema (1).

Nesse sentido , apresentamos o algorítmo II para avaliação de seu desempenho e o submetemos a um conjunto de problemas testes relacionados em Hock e Schittkowiski [34]. Registramos igualmente robustez e eficiência frente a esse conjunto de problemas testados.

A apresentação do presente trabalho é feita com mais detalhes em descrição dos capítulos , visto logo a seguir.

DESCRIÇÃO DOS CAPÍTULOS

Este trabalho de tese consta de cinco capítulos assim distribuídos :

No Capítulo 1 , apresentamos algumas definições e condições de otimalidade para problemas de programação não-linear irrestritos como também para problemas restritos.

No Capítulo 2 , apresentamos uma revisão bibliográfica envolvendo basicamente alguns métodos de resolução dos problemas de programação não-linear com restrições tais como : método de direções viáveis , método de gradiente projetado, método de gradiente reduzido , método de gradiente reduzido generalizado , método de penalidades tais como : pontos exteriores , pontos interiores , mista , exata , método dos multiplicadores , método de programação quadrática sequencial e finalmente o método de penalização hiperbólica .

Do método de penalização hiperbólica , nos emulamos para chegarmos ao objetivo do trabalho de tese , que trata de resolver o problema de programação não-linear com restrições de igualdades , que será apresentado no capítulo seguinte .

No Capítulo 3 , apresentamos o problema de programação não-linear com restrições de igualdades , a ser resolvido , utilizando a penalização hiperbólica , o algorítmo I e seu funcionamento . Fundamentado sob um conjunto de condições que o problema deve satisfazer , são desenvolvidos os resultados teóricos que nos levam até a convergência .

Com o objetivo de mostrarmos o funcionamento do algorítmo e a validade do método , apresentamos os resultados computacionais , obtidos a partir de um exemplo simples de programação não-linear restrito , contendo duas variáveis e somente uma restrição de igualdade ,

Apresentamos ainda , uma relação de alguns problemas testes do livro de Hock e Schittkowski [34] , envolvendo somente restrições de igualdades , inclusive alguns problemas considerados degenerados , face a presença de multiplicadores de Lagrange iguais a zero , contrariando a condição de $\lambda_j^k \neq 0$, $j = 1 , \dots , p$ estabelecidos anteriormente.

No capítulo 4 , desenvolve-se um novo algorítmo II para o problema geral de programação não-linear com restrições simultaneamente de igualdades e desigualdades .

Com o objetivo de mostrarmos o funcionamento do algorítmo e a validade do método , apresentamos os resultados computacionais , obtidos a partir de um exemplo simples de programação não-linear contendo duas variáveis , uma restrição de igualdade e três restrições de desigualdades ,

Apresentamos ainda , uma relação de alguns problemas testes do livro de Hock e Schittkowski [34] , envolvendo restrições de igualdades e desigualdades , onde todos foram resolvidos com sucesso utilizando-se o algorítmo II .

No capítulo 5 , apresentamos as conclusões e propostas para estudos futuros.

CLASSIFICAÇÃO DOS PROBLEMAS

Nosso objetivo é mostrar uma introdução no campo da programação não-linear, com a perspectiva de abordar em síntese, mas com a máxima amplitude, os resultados-padrão mais importantes atualmente disponíveis nessa área, tais como algumas definições e condições de otimalidade para os problemas de programação não-linear.

Basicamente abordaremos dois tipos de problemas : os denominados irrestritos, e os denominados restritos. Os problemas restritos consistem em otimizar (minimizar ou maximizar) uma função , denominada função objetivo, respeitando um conjunto de restrições.

Estas restrições determinam um conjunto de soluções viáveis, e a solução viável que otimiza a função objetivo, que é a melhor solução encontrada, denomina-se solução ótima do problema.

O objetivo ideal da otimização seria encontrar o ótimo global. Infelizmente, os ótimos globais podem ser caracterizados somente em alguns casos particulares, como nos problemas de programação convexa. Portanto, quando falarmos em otimalidade estamos nos referindo aos ótimos locais.

A representação de um problema de minimização restrito (PR) é dado por :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad x \in S, \end{aligned} \tag{1.1}$$

onde $x = (x_1, x_2, \dots, x_n)$ designa as variáveis, S o conjunto de restrições ou conjunto solução, e $f(.) : S \subset R^n \rightarrow R$ é uma função objetivo qualquer.

Por outro lado, quando $S = R^n$, temos os problemas ditos irrestritos (PI), que consistem em minimizar uma função objetivo sem restrições, e cuja representação é dada por :

$$\min_{x \in R^n} \quad f(x) \tag{1.2}$$

DEFINIÇÃO (1.1) : (Mínimo Local)

Um ponto $x^* \in S$ é dito uma solução local ou relativa do problema (PR), se existe uma vizinhança $\Omega(x^*, \Omega) = \{ z / | z - x^* | < \delta ; \delta > 0 \}$ tal que $f(x^*) \leq f(x)$, para todo $x \in \Omega(x^*, \delta) \cap S$.

DEFINIÇÃO (1.2) : (Mínimo global)

Um ponto $x^* \in S$ é dito um mínimo global ou absoluto do problema (PR) se $f(x^*) \leq f(x)$, para qualquer $x \in S$.

Usualmente, os métodos de programação não-linear são iterativos, no sentido de que, partindo-se de um ponto inicial x^o, uma sequência de pontos $\{ x^k \}$ é obtida através de repetidas aplicações de uma regra algorítmica. Essa sequência deve convergir para uma solução x^* do problema.

DEFINIÇÃO (1.3) : (Convergência Local)

Um algorítmo é localmente convergente , se existe um escalar σ positivo, tal que para qualquer ponto inicial $x^o \in R^n$ (ou $x^o \in S$) , verificando $| x^o - x^* | \leq \sigma$, o algorítmo gera uma seqüência de pontos convergindo para a solução x^* do problema original .

DEFINIÇÃO (1.4) : (Convergência Global)

Um algorítmo iterativo é dito globalmente convergente, se para qualquer ponto inicial $x^o \in R^n$ (ou $x^o \in S$), o algorítmo gera uma sequência de pontos convergindo para uma solução do problema original .

A convergência é dita assintótica , quando a solução não é encontrada após um número finito de iterações . Exceto em alguns casos particulares , como problemas de programação linear e também de programação quadrática , um caso de programação não-linear , posteriormente descrito , podendo ter convergência finita.

Em geral , os algorítmos globalmente convergentes definem a cada ponto uma direção de busca à procura de um novo ponto . Daí teremos :

DEFINIÇÃO (1.5) : (Direção de Descida)

Um vetor $d \in R^n$ é uma direção de descida de uma função real f em $x \in R^n$, se existe $\delta > 0$, tal que $f(x + td) < f(x)$ para qualquer $t \in (0 , \delta]$.

DEFINIÇÃO (1.6) : (Direção Viável)

Um vetor $d \in R^n$ é uma direção viável para o problema geral em $x \in S$, se para algum $\theta > 0$ temos $x + td \in S$ para todo $t \in [\, 0 , \theta \,]$

Os resultados mais importantes na programação não-linear são as condições de otimalidade , porque elas devem ser satisfeitas em qualquer ponto ótimo restrito, local ou global , de qualquer problema de programação linear , e na maioria dos problemas de programação não-linear . De outro lado , essas condições formam a base para o desenvolvimento de muitos algorítmos . Além disso , os critérios de parada de muitos algorítmos , isto é , de reconhecer quando um ponto ótimo local restrito é atingido , são diretamente derivados dessas condições , as quais apresentaremos a seguir .

CONDIÇÕES DE OTIMALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO IRRESTRITO

Considere o problema de programação não-linear irrestrito dado por :

$$\underset{x \in R^n}{\text{minimizar}} \quad f(x) \qquad (1.2)$$

CONDIÇÃO NECESSÁRIA DE PRIMEIRA ORDEM :

Se a função $f(\,.\,)$ é de classe C^1 , então uma condição necessária para $x^* \in R^n$ ser uma solução do problema (1.2), é que :

$$\nabla f(x^*) = 0 \qquad (1.3)$$

CONDIÇÃO NECESSÁRIA DE SEGUNDA ORDEM :

Se a função $f(\,.\,)$ for de classe C^2, então uma condição necessária para $x^* \in R^n$ ser uma solução do problema (1.2) é que $\nabla^2 f(x^*)$, a Hessiana de f em x^*, seja semidefinida positiva, isto é, para toda direção $d \in R^n$,

$$d^t \,\nabla^2 f(x^*)\, d \geq 0 . \tag{1.4}$$

CONDIÇÕES DE SUFICIÊNCIA :

As condições de suficiência para $x^* \in R^n$ ser uma solução local para o problema irrestrito (1.2) são :

$$\nabla f(x^*) = 0 \tag{1.5}$$

e

$$z^t \nabla^2 f(x^*)\, z > 0 , \tag{1.6}$$

para todos os vetores não-nulos $z \in R^n$.

DEFINIÇÃO DOS PROBLEMAS DE OTIMIZAÇÃO COM RESTRIÇÕES

Primeiramente dizemos que , se o conjunto viável S for definido por um conjunto de restrições de igualdades e desigualdades , então o problema (1.1) será chamado de problema geral de programação não-linear restrito (PGR) , e sua representação é dada por :

$$\begin{aligned} &\text{minimizar} \quad f(x) && \qquad (1.7)\\ &\text{sujeito à :} \quad g_i(x) \geq 0 \quad , \; i = 1, \dots, m \\ &\qquad\qquad\quad h_j(x) = 0 \quad , \; j = 1, \dots, p \end{aligned}$$

onde $f(\,.\,) : R^n \to R$, $g_i(\,.\,) : R^n \to R^m$ e $h_j(\,.\,) : R^n \to R^p$ são funções não-lineares e diferenciáveis .

Como um caso especial do problema (1.3) , podemos pensar em minimizar uma função $f(\,.\,)$ sobre uma superfície S , definida somente pelas equações $h_j(x) = 0$ para $j = 1, \ldots, p < n$. Como estamos tratando unicamente com restrições de igualdades , então , nos referimos a este problema como um problema de otimização restrito a igualdades (PRI) , e pode ser escrito na forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad h_j(x) = 0 \quad ; \quad j = 1,\ldots,p \end{aligned} \tag{1.8}$$

Por outro lado , nos referimos a problemas em que somente as restições de desigualdades são envolvidas , nos proporcionando os chamados problemas de otimização restrito a desigualdades , que podem ser expressos na forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad g_i(x) \geq 0 \quad ; \quad i = 1,\ldots,m \end{aligned} \tag{1.9}$$

CONDIÇÕES DE OTIMALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO COM RESTRIÇÕES DE IGUALDADE

Primeiramente introduzimos as variáveis auxiliares $\mu \in R^p$ e $\lambda \in R^m$, chamadas de Variáveis Duais ou de Multiplicadores de Lagrange para definirmos as seguintes funções Lagrangeanas .

DEFINIÇÃO (1.7) :

A função Lagrangeana $l(\,.\,) : R^n \times R^p \rightarrow R$ associada ao problema restrito a igualdades (1.8) é definida como :

$$l(x, \mu) = f(x) + \mu^t h(x) \qquad (1.10)$$

onde $\mu \in R^p$ é o vetor multiplicador de Lagrange .

Restrições não-lineares podem ser extremamente complicadas , de maneira que as condições de otimalidade serão definidas fazendo-se hipóteses favoráveis a respeito dessas restrições , que são conhecidas na literatura como condições de qualificação das restrições .

Uma condição normalmente usada , conhecida como hipótese de regularidade, é apresentada a seguir :

DEFINIÇÃO (1.8) : (Regularidade)

Um ponto $x \in S$ é um ponto regular das restrições se os vetores $\nabla h_j(x^*)$, para $j = 1, \dots, p$ são linearmente independentes .

CONDIÇÕES NECESSÁRIAS DE PRIMEIRA ORDEM :

As condições necessárias de primeira ordem para que x^* seja um ponto de mínimo local para o problema (1.8) , ou condições de Karush - Kuhn - Tucker (KKT) , são que x^* seja um ponto viável , $h(x^*) = 0$, e que exista um multiplicador de Lagrange $\mu^* \in R^p$ tal que :

$$\nabla f(x^*) + \mu^{*t} \nabla h(x^*) = 0 \qquad (1.11)$$

Fazendo uso da função Lagrangeana, esta equação é equivalente a:

$$\nabla l(x^*, \mu^*) = 0 \qquad (1.12)$$

CONDIÇÕES NECESSÁRIAS DE SEGUNDA ORDEM:

A condição necessária de segunda ordem para que x^* seja um ponto de mínimo local para o problema (1.8) é que exista $\mu^* \in R^p$ tal que sejam satisfeitas as condições de primeira ordem (1.11), e que a Hessiana da função Lagrangeana seja semidefinida positiva para todos os vetores sobre o espaço nulo associado aos vetores gradientes $\nabla h(x^*)^t$, isto é, para todo vetor v que satisfaz

$$\nabla h(x^*)^t v = 0,$$

deve ser observado

$$v^t \nabla^2_x l(x^*, \mu^*) v \geq 0 \qquad (1.13)$$

CONDIÇÕES DE SUFICIÊNCIA:

As condições suficientes para x^* ser um minimizador local isolado do problema restrito (1.8) são que x^* seja um ponto KKT, isto é, x^* e μ^* satisfaçam as condições necessárias de primeira ordem, e que

$$v^t \nabla^2_x l(x^*, \mu^*) v > 0, \qquad (1.14)$$

para todo vetor $v \neq 0$ que satisfaça

$$\nabla h(x^*)^t . v = 0.$$

CONDIÇÕES DE OTIMALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO COM RESTRIÇÕES DE DESIGUALDADE

Considerando o problema de programação não-linear com restrições de desigualdade dado em (1.9), definimos sua função Lagrangeana como :

DEFINIÇÃO (1.9) :

A função Lagrangeana $l(\,.\,) : R^n \times R^m \to R$ associada ao problema restrito a desigualdades (1.9) é definida como :

$$l(x, \lambda) = f(x) - \lambda^t g(x) , \qquad (1.15)$$

onde $\lambda \in R^m$, $\lambda \geq 0$, é o vetor multiplicador de Lagrange .

DEFINIÇÃO (1.10) : (Regularidade) .

Um ponto $x \in S$ é um ponto regular das restrições , se os vetores $\nabla g_i(x)$ para $i \in I(x)$, são linearmente independentes , onde $I(x) = \{ i / g_i(x) = 0 \}$ é o conjunto de restrições ativas em x .

Suponhamos agora que x^* , um ponto regular , seja um mínimo local para o problema (1.9) . É claro que x^* é também um mínimo local de $f(x)$, sujeito à $g_i(x) = 0$, para $i \in I(x^*)$. Então , segue das condicões necessárias de primeira ordem que existe um vetor $\lambda^* \in R^m$ tal que :

$$\nabla f(x^*) - \lambda^{*t} \nabla g(x^*) = 0, \qquad (1.16)$$

onde $\lambda_i^* = 0$ para $i \notin I(x^*)$.

A condição $\lambda_i^* = 0$ para $i \notin I(x^*)$ é chamada de Condição de Complementaridade, e pode ser representada por meio de as seguintes igualdades:

$$\lambda_i^* g_i(x^*) = 0 \; ; \qquad i \;\; 1, \ldots, m \qquad (1.17)$$

Ademais, se x^* é um ponto de mínimo, devemos ter necessariamente $\lambda_i^* \geq 0$, $i \in I(x^*)$, pois em caso contrário, haveria uma direção de decréscimo de $f(x)$, que seria viável.

CONDIÇÕES NECESSÁRIAS DE PRIMEIRA ORDEM:

As condições necessárias de primeira ordem para que x^*, um ponto regular em relação às restrições, seja um ponto de mínimo do problema (1.9), é que exista um vetor $\lambda^* \in R^m$ tal que:

$$\begin{aligned} \nabla f(x^*) - \lambda^{*t} \nabla g(x^*) &= 0 \\ \lambda^{*t} g(x^*) &= 0 \\ \lambda^* &\geq 0 \\ g(x^*) &\geq 0 \end{aligned} \qquad (1.18)$$

CONDIÇÕES DE OTIMALIDADE PARA O PROBLEMA GERAL DE OTIMIZAÇÃO

Considerando o problema geral de programação não-linear (1.7) , temos :

DEFINIÇÃO (1.11) :

A função Lagrangeana $l(\,.\,) : R^n \times R^m \times R^p \to R$ associada ao problema geral de programação (1.7) é definida como :

$$l(x,\mu,\lambda) = f(x) + \mu^t h(x) - \lambda^t g(x) \qquad (1.19)$$

DEFINIÇÃO (1.12) : (Regularidade) .

Um ponto $x \in S$ é um ponto regular em relaçao às restrições , se os vetores $\nabla h_j(x)$, para $j = 1, \ldots, p$ e $\nabla g_i(x)$, para $i \in I(x)$ são linearmente independentes , onde $I(x) = \{\, i \,/\, g_i(x) = 0 \,\}$.

CONDIÇÕES NECESSÁRIAS DE PRIMEIRA ORDEM OU KKT :

As condições necessárias de primeira ordem ou condições KKT para um ponto $x^* \in S$ ser um ponto de mínimo local para o problema (1.7) . Supondo que x^* seja um ponto regular para as restrições dadas na definição de S , é que exista um vetor multiplicador $\mu^* \in R^p$ e um vetor multiplicador $\lambda^* \in R^m$, $\lambda^* \geq 0$ tais que :

$$\begin{aligned} \nabla f(x^*) + \mu^{*t}.\nabla h(x^*) - \lambda^{*t}\nabla g(x^*) &= 0 \\ \lambda^{*t} g(x^*) &= 0 \\ h(x^*) &= 0 \\ g(x^*) &\geq 0 \end{aligned} \qquad (1.20)$$

CONDIÇÕES NECESSÁRIAS DE SEGUNDA ORDEM :

Nas condições necessárias de segunda ordem supomos que as funções $f(\,.\,)$, $g(\,.\,)$, e $h(\,.\,)$ sejam de classe C^2, e que x^* seja um ponto regular em relação às restrições $h(x^*) = 0$ e $g(x^*) \geq 0$. Se x^* é um ponto de mínimo local para o problema (1.7), então existe um vetor $\mu^* \in R^p$ e um vetor $\lambda^* \in R^m$, $\lambda^* \geq 0$ tais que as condições de primeira ordem sejam obedecidas, e que a matriz

$$\nabla^2 l(x^*, \mu^*, \lambda^*) = \nabla^2 f(x^*) + \mu^{*t} \nabla^2 h(x^*) - \lambda^{*t} \nabla^2 g(x^*) \qquad (1.21)$$

seja semi-definida positiva sobre o espaço tangente associado às restrições ativas em x^*. Denotamos $\nabla^2 l(x^*, \mu^*, \lambda^*)$ como sendo a Hessiana da função Lagrangeana relativa ao problema (1.6), e $\nabla^2 f(x^*)$, $\nabla^2 h(x^*)$ e $\nabla^2 g(x^*)$ como sendo as Hessianas das funções f, h e g respectivamente.

CONDIÇÕES DE SUFICIÊNCIA :

Agora, nas condições de suficiência de segunda ordem, sendo as funções f, g e h de classe C^2, e um ponto x^* satisfazendo as restrições $h(x^*) = 0$ e $g(x^*) \geq 0$ para ser um ponto de mínimo local para o problema (1.7) é que exista um vetor $\mu^* \in R^p$ e um vetor $\lambda^* \in R^m$ tais que :

$$\begin{aligned} \nabla f(x^*) + \mu^{*t} \nabla h(x^*) - \lambda^{*t} \nabla g(x^*) &= 0 \\ \lambda^{*t} g(x^*) &= 0 \\ \lambda^* &\geq 0 \,, \end{aligned} \qquad (1.22)$$

e a matriz Hessiana

$$\nabla^2 l(x^*, \mu^*, \lambda^*) = \nabla^2 f(x^*) + \mu^{*t} \nabla^2 h(x^*) - \lambda^{*t} \nabla^2 g(x^*) \qquad (1.23)$$

seja definida positiva sobre o espaço tangente M , onde

$$M = \{ y / \nabla h(x^*).y = 0 \ ; \ \nabla g(x^*).y = 0 \ \text{ para todo } i \in I \}$$

com $\qquad I = \{ i / g_i(x^*) = 0 \ ; \ \lambda_i > 0 \}$

CAPÍTULO 2

MÉTODOS UTILIZADOS PARA RESOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO NÃO-LINEAR COM RESTRIÇÕES

O objetivo fundamental do presente trabalho, como já referido na introdução, é o desenvolvimento de um algorítmo para a resolução do problema de programação não-linear com restrições de igualdade, e como uma extensão, o problema geral. Face a essa proposta, julgamos de suma importância efetuar uma apresentação sintética de alguns dos principais métodos destinados à resolução dos problemas de otimização com restrições.

MÉTODO DE DIREÇÕES VIÁVEIS

A idéia central desse método é adaptar os princípios dos métodos de minimização irrestrita, levando em consideração as limitações impostas pela presença de restrições.

Considere o problema:

$$\text{minimizar } f(x) \qquad (2.1)$$

$$\text{sujeito à: } x \in X \subset R^n,$$

onde X é um subconjunto fechado do R^n e $f(.)$ uma função continuamente diferenciável, que tem conjunto de nível limitado.

Um algorítmo para resolver o problema (2.1) segundo o enfoque de direções viáveis segue basicamente os seguintes passos :

1 - Para $k = 0, 1, \ldots$, e dado um ponto $x^k \in X$, encontrar um vetor direção d^{k+1} que seja de decréscimo, ou seja :

$$(d^{k+1})^t \nabla f(x^k) < 0$$

e também que seja uma direção viável, ou seja, que exista um $\bar{\theta}_{k+1} > 0$ satisfazendo $x^k + \theta_{k+1} d^{k+1} \in X$ para todo $0 \leq \theta_{k+1} \leq \bar{\theta}_{k+1}$. Em síntese, d^{k+1} é simultaneamente uma direção de descida e uma direção viável.

2 - Determinar um passo θ_{k+1}^* por algum critério de forma que :

$$x^{k+1} = x^k + \theta_{k+1}^* d^{k+1} \in X$$

e que $f(x^{k+1}) < f(x^k)$.

3 - Continue dessa forma até uma condição de parada ser atendida, e a convergência será atingida.

A despeito da convergência desse método ser aparentemente segura, ela não pode ser garantida para um problema geral, conforme descrito detalhadamente em Bazaraa [6] e Minoux [48].

MÉTODO DO GRADIENTE PROJETADO

Um dos métodos mais conhecidos da classe dos métodos de direções viáveis é o método de Rosen [59] , que essencialmente é uma adaptação do método do gradiente , que leva em consideração as restrições do problema . Essa adaptação se constitue basicamente na projeção ortogonal do gradiente, na variedade linear das restrições ativas. O método de Rosen [58] é melhor aplicável nos problemas onde as restrições são lineares da forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad a_i x \leq b_i \quad ; \ i \in I_1 \\ &\qquad\qquad\qquad a_i x = b_i \quad ; \ i \in I_2 \end{aligned} \tag{2.2}$$

Para o caso mais geral , onde as restrições são não-lineares , segue um esquema semelhante , isto é , em um ponto viável x^k , determina-se as restrições ativas , e projeta-se o gradiente naquele subespaço , que é tangente à variedade gerada pelas restrições . Esse vetor , se for diferente de zero , determinará a direção para a obtenção do próximo ponto . Face a não-linearidade das restrições , geralmente esse vetor não é uma direção viável , devido a curvatura das superfícies. Neste caso , ao se mover ao longo do gradiente projetado , afasta-se da região viável . Face a esse fenômeno , o método deve incorporar uma estratégia de retorno à região viável .

MÉTODO DE GRADIENTE REDUZIDO

O método de gradiente reduzido, devido a Wolfe [61], é uma extensão direta do método simplex, da programação linear, aplicado ao caso da função objetivo não-linear.

Para elucidar sua conexão com o método simplex, considere o problema:

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à:} \quad Ax = b \\ &\qquad\qquad\quad x \geq 0 \end{aligned} \qquad (2.3)$$

O vetor das variáveis define-se como $x = (x_B, x_N)$, a matriz A escreve-se na forma $A = [B, N]$, onde B é uma base, isto é, uma submatriz quadrada de ordem m, não-singular extraída da matriz A.

As restrições de igualdade podem ser escritas como:

$$Bx_B + Nx_N = b \qquad (i)$$

o que nos permite escrever:

$$x_B = B^{-1} b - B^{-1} Nx_N \qquad (ii)$$

com $B^{-1} = \underline{b}$ e $B^{-1}N = \underline{N}$, tal como se faz no método simplex.

Assim sendo, para que um deslocamento infinitesimal $dx = (dx_B , dx_N)$ seja compatível com as equações lineares, devemos ter :

$$dx_B = B^{-1} N \, dx_N \qquad (iii)$$

A idéia básica dos métodos de gradiente reduzido, consiste em eliminar x_B (como uma função de x_N) através de (ii), e considerar o problema de otimização em termos de x_N apenas. Essa idéia tem sido usada em outros algorítmos de programação não-linear, como por exemplo, no método simplex convexo de Zangwill [68].

A variação da função objetivo f para um pequeno deslocamento dx compatível com as restrições é dada por :

$$df = (\partial f / \partial x_B)^t dx_B + (\partial f / \partial x_N)^t dx_N \qquad (iv)$$

ou

$$df = (\partial f / \partial x_B)^t (- B^{-1} N \, dx_N) + (\partial f / \partial x_N)^t dx_N , \qquad (v)$$

onde, pode-se escrever :

$$df = u_n^t \, dx_N , \qquad (vi)$$

com

$$u_n^t = (\partial f / \partial x_N)^t - (\partial f / \partial x_B)^t B^{-1} N , \qquad (vii)$$

que por definição, é o gradiente reduzido da função f em relação à base B.

Wolfe, para a caso de restrições não-lineares, simplesmente toma como a matriz A do problema (2.3), a matriz jacobiana do sistema $h(x^k) = 0$, do problema (1.8), ou seja, é efetuada a linearização dessas restrições.

MÉTODO DE GRADIENTE REDUZIDO GENERALIZADO

O método de gradiente reduzido foi generalizado por Abadie e Carpentier [1] em 1969, para o caso de restrições não-lineares de igualdades, e variáveis limitadas inferiormente, ou seja, para o problema :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à:} \quad h(x) = 0 \\ &\qquad\qquad\qquad x \geq 0 \end{aligned} \tag{2.4}$$

O método combina as idéias de gradiente reduzido de Wolfe, com as estratégias do método de linearização, e tem sido considerado um dos mais eficientes para tratar o caso mais geral em que as restrições, assim como a função objetivo, são não-lineares. Veja Colville [19], Abadie & Gulgou [2] e Schittkowski [34].

O método tem total analogia com o método de gradiente de Wolfe, e portanto, com o simplex, logo a idéia básica é que um conjunto de restrições de igualdades não-lineares é um sistema de equações, onde, de maneira ímplícita é possível colocar algumas variáveis em função das outras.

Assim , minimizar com esse conjunto de restrições passa a ser um problema irrestrito , cujas variáveis são exatamente as variáveis escolhidas como independentes . Esta fundamentação se dá pelo teorema da função implícita .

MINIMIZAÇÃO RESTRITA VIA RESOLUÇÃO DAS EQUAÇÕES KARUSH-KUHN -TUCKER

As técnicas para otimização irrestrita podem ser estendidas para o problema

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad h(x) = 0 \end{aligned} \tag{2.5}$$

O par (x^* , μ^*) , satisfazendo as condições de otimalidade de primeira ordem Karush - Kuhn - Tucker (1.11) , é solução do sistema de equações não-lineares :

$$\begin{aligned} \nabla f(x) + \mu^t \nabla h(x) &= 0 \\ h(x) &= 0 \ , \end{aligned} \tag{2.6}$$

onde as variáveis x e μ sào chamadas de variáveis primais e duais , respectivamente.

Para termos uma solução μ única em (2.6) num ponto ótimo x^*, este deve ser um ponto regular do problema (2.5) , que corresponde ao Jacobiano do segundo conjunto de igualdades do sistema (2.6) , ser não-singular no ponto x^* .

Devido a esse fato , os algorítmos que resolvem as condições de otimalidade de primeira ordem , requerem a hipótese de que os iterados x^k sejam pontos regulares dos problemas .

Fazendo-se :

$$y = (x, \mu) \ , \qquad (2.7)$$

e

$$\phi(y) = \begin{pmatrix} \nabla f(x) + \mu \nabla h(x) \\ h(x) \end{pmatrix} , \qquad (2.8)$$

temos :

$$\nabla\phi(y) = \begin{bmatrix} H(x, \mu) & \nabla h(x)^t \\ \nabla h(x) & 0 \end{bmatrix} , \qquad (2.9)$$

onde :

$$H(x, \mu) = \nabla^2 f(x) + \sum_{j=1}^{p} \mu_j \nabla^2 h_j(x)$$

é a Hessiana da função Lagrangeana definida em (1.10) .

Os métodos baseados na resolução das equações Karush- Kuhn - Tucker , basicamente se utilizam do método de Newton para resolução do sistema de equações não-lineares (2.6).

A iteração de Newton que começa em (x^k , μ_k) , tem como próximo ponto (x^{k+1} , μ_{k+1}) , a solução do sistema linearizado :

$$\begin{bmatrix} H(x, \mu_k) & \nabla h(x^k)^t \\ \nabla h(x^k) & 0 \end{bmatrix} \cdot \begin{bmatrix} x^{k+1} - x^k \\ \mu_{k+1} - \mu_k \end{bmatrix} = \begin{bmatrix} \nabla l(x^k, \mu_k) \\ h(x^k) \end{bmatrix} \qquad (2.10)$$

Sob condições especiais , a sequência de pares gerada converge para um par ótimo (x^* , μ^*) .

PROGRAMAÇÃO QUADRÁTICA SEQUENCIAL

A programação quadrática sequencial é um método bastante empregado em problemas de otimização não-linear com restrições. É uma técnica Quasi-Newton baseada na primeira idéia proposta por Wilson [60], e interpretada por Beale [7]. Pesquisadores tais como Bard e Greenstadt [5], Biggs [13] e outros mais, também desenvolveram esse método.

Um programa quadrático é uma classe de problemas de otimização com restrições, tais que a função objetivo é uma função quadrática, e as restrições são necessariamente lineares. Existem técnicas eficientes para resolver estes problemas, mesmo quando restrições de desigualdades são incluídas. A solução exata é obtida após um número finito de iterações.

Para explicar a idéia de Wilson, consideremos o seguinte problema:

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeito à : } h(x) = 0, \end{aligned} \tag{2.11}$$

onde temos a função objetivo e um conjunto de restrições, geralmente não-lineares todas diferenciáveis.

Conforme exposto em Martinez [43], o método de Wilson consiste em substituir, em cada passo, a função objetivo por uma aproximação quadrática, e as restrições por equações ou inequações lineares.

Dessa maneira , o subproblema a ser considerado em cada iteração é consideravelmente mais simples em comparação ao problema original , ou seja , o subproblema se constitui em um problema de programação quadrática (PQ) da forma :

$$\text{minimizar} \quad \nabla f(x)^t d + \tfrac{1}{2} d^t B d \tag{2.12}$$

$$\text{sujeito à :} \quad \nabla h(x)^t d + h(x) = 0 \,,$$

onde d é a direção de deslocamento , e B é uma matriz definida positiva , basicamente tomada como uma aproximação Quasi-Newton da Hessiana.

Se adota B como uma matriz definida positiva com o objetivo de garantir a existência de ponto de mínimo global do problema (2.12) (o que não necessariamente acontece , quando se toma B como a própria matriz Hessiana do Lagrangeano) .

Como o problema (2.12) é convexo , o mínimo global satisfaz as condições de otimalidade de KKT , isto é :

$$B d + \nabla f(x) + \nabla f(x)\,\mu = 0 \tag{2.13}$$

$$\nabla h(x)^t d + h(x) = 0 \,,$$

onde μ é o vetor multiplicador de Lagrange . Então , (d , μ) pode ser obtido , resolvendo-se o problema (2.12).

Baseado nessa idéia, para resolver o problema geral de programação não-linear simultaneamente com retrições de igualdades e desigualdades, Wilson na sua proposta definiu a direção de busca d, e novas estimativas dos multiplicadores de Lagrange μ e λ para a resolução em cada iteração do subproblema:

$$\begin{array}{ll} \text{minimizar} & \nabla f(x)^t d + \tfrac{1}{2} d^t B d \\ \text{sujeito à :} & \nabla g(x)^t d + g(x) \geq 0 \\ & \nabla h(x)^t d + h(x) = 0 \end{array} \qquad (2.14)$$

O algorítmo de Wilson de programação quadrática sequencial é assim, uma generalização dos métodos de Newton aplicado ao sistema de condições KKT. De uma maneira genérica, nos algorítmos de programação quadrática sequencial, a matriz B é definida como uma aproximação Quasi-Newton da Hessiana da Lagrangeana. Ademais a função de mérito utilizada na busca linear é normalmente uma função de penalidade exata que contempla simultaneamente a função objetivo e a observância das restrições do problema, como por exemplo:

$$P(x,r,s) = f(x) + \sum_{i=1}^{m} s_i \{ \sup [0, g_i(x)] \} + \sum_{j=1}^{p} r_j \mid h_j(x) \mid \qquad (2.15)$$

MÉTODOS DE PENALIDADES

Dado o problema de programação não-linear com restrições da forma :

$$\text{minimizar} \quad f(x) \qquad (2.16)$$

$$\text{sujeito à:} \quad x \in S$$

A idéia comum aos métodos de penalidades é a de resolver o problema acima, transformando-os na resolução de problemas irrestritos sob a seguinte forma:

$$\text{minimizar} \quad f(x) + P(x) \qquad (2.17)$$

$$x \in R^n$$

A função $P(x)$, denominada função penalidade, deve incorporar de uma forma conveniente o espaço viável S.

Os métodos de penalidades podem ser basicamente classificados em métodos de penalização exterior e penalização interior.

MÉTODOS DE PENALIDADE EXTERIOR

Esses métodos transformam o problema restrito em um problema irrestrito, introduzindo as restrições na função objetivo, penalizando qualquer violação das mesmas. Nessa família de métodos, a função de penalidade gera uma penalidade positiva para pontos inviáveis, e em contrapartida não penaliza os pontos viáveis.

Esses métodos obtém a solução do problema original através de uma sequência de minimizações irrestritas . A sequência de pontos de ótimos gerada nesse processo tem como limite um ponto , que para esse caso , a função de penalidade P (x) deve satisfazer :

i) P (x) é contínua

ii) P (x) $\geq$ 0

iii) P (x) = 0 $\Leftrightarrow$ x $\in$ S

O efeito da função de penalidade exterior é visto na figura 1 :

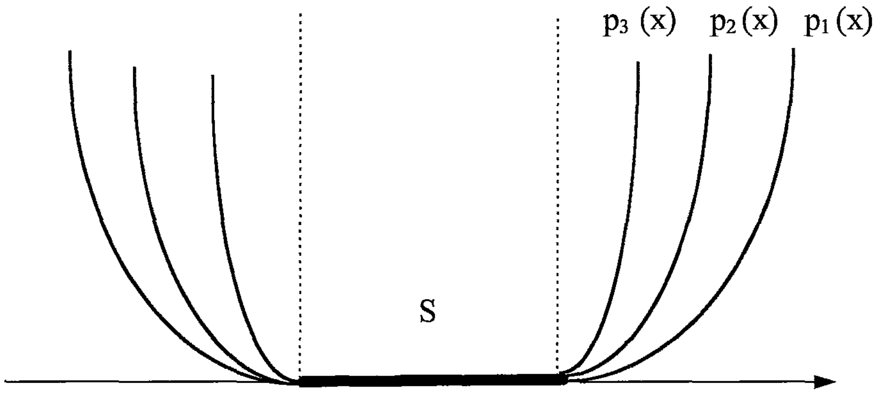

Figura 1 : Penalização Exterior

O uso das funções de penalidades exterior em problemas restritos foi proposto originalmente por Courant [20] . Subsequentemente Camp [15] discutiu esse procedimento para resolver problemas não-lineares . No entanto , o estudo detalhado de tais métodos para resolver problemas práticos foi apresentado por Fiacco & McCormick primeiramente em [24] , e mais detalhadamente em [25]. Outros autores como : Avriel [4] , Polak [51] , Luemberger [42] , Zangwill [67 , 68] , Himmelblau [33] , Lootsma [40] e Osborne e Ryan [49] também estudaram esses métodos.

A primeira função de penalidade para o problema restrito , com restrições de igualdades :

$$\text{minimizar} \quad f(x) \tag{2.18}$$
$$\text{sujeito à :} \quad h(x) = 0 \ ,$$

sugerida por Courant [20] em 1943 , foi definida de tal forma a produzir o problema:

$$\underset{x \in R^n}{\text{minimizar}} \quad f(x) + \gamma [h(x)]^2 \quad ; \quad \gamma \geq 0 \tag{2.19}$$

Assim , nesta proposta , a resolução de problemas com restrições de igualdades , é obtida através de uma seqüência de minimizações da forma :

$$\underset{x \in R^n}{\text{minimizar}} \quad f(x) + \tfrac{1}{2} c_k \, | h(x) |^2 \quad , \tag{2.20}$$

onde $\{ c_k \}$ é uma seqüência de escalares positivos com $c_k < c_{k+1}$ para todo k , e $c_k \to +\infty$, as funções $f(.) : R^n \to R$, $h(.) : R^n \to R^p$ são contínuas , e $| \, . \, |$ é a norma Euclideana, conforme apresentado com detalhes em Fiacco & McCormick [25].

Considerando o problema de programação restrito a desigualdades :

$$\text{minimizar} \quad f(x) \tag{2.21}$$
$$\text{sujeito à :} \quad g_i(x) \leq 0 \quad ; \quad i = 1, \ldots, m \quad ,$$

Zangwill [67 , 68] usa duas alternativas de funções de penalidades que produzem problemas irrestritos :

a) $$\underset{x \in R^n}{\text{minimizar}} \quad f(x) + \gamma \sum_{i=1}^{m} \max [\, 0 , g_i(x) \,] \tag{2.22}$$

b) $$\underset{x \in R^n}{\text{minimizar}} \quad f(x) + \gamma . \sum_{i=1}^{m} [\max \, (0 , g_i(x))]^2 \tag{2.23}$$

Agora , considerando o problema geral de programação , onde as funções f , g e h são contínuas em R^n e $S = \{ x / g_i(x) \geq 0 , i = 1,...,m , h_j(x) = 0 , j = 1,...,p \}$ é o conjunto viável . Zangwill em [68] , generaliza as penalidades definidas anteriormente através das funções φ e ϕ de variáveis $\xi \in R$ por :

$$\varphi(\xi) = | \min (0, \xi) |^{\alpha} \tag{2.24}$$

$$\phi(\xi) = | \xi |^{\beta} \tag{2.25}$$

onde $\alpha \geq 0$ e $\beta \geq 0$ são constantes dadas usualmente iguais a 1 e 2 respectivamente, e toma :

$$P(x) = \sum_{i=1}^{m} \varphi(g_i(x)) + \sum_{j=1}^{p} \phi(h_j(x)) \tag{2.26}$$

ou seja :

$$P(x) = \sum_{i=1}^{m} | \min (0, g_i(x)) |^{\alpha} + \sum_{j=1}^{p} | h_j(x) |^{\beta} \tag{2.27}$$

Analogamente a (2.19),o problema original é resolvido através da resolução da seqüência de problemas :

$$\underset{x \in R^n}{\text{minimizar}} \quad f(x) + 1/\gamma^k P(x) \quad ; \quad k = 0, 1, \ldots\ldots \qquad (2.28)$$

Uma outra classe de técnicas de minimização sequencial bastante utilizada para resolver problemas de programação não-linear sujeito a restrições de desigualdades são conhecidas como :

MÉTODOS DE PENALIDADE INTERIOR OU BARREIRA

Estes métodos , tais como os métodos de penalidade exterior , substituem a resolução de um problema restrito pela resolução de uma seqüência de problemas irrestritos, em que a função objetivo é igual à função objetivo original, adicionada do termo de penalidade , no qual são consideradas as restrições .

Essa família de métodos trabalha unicamente com pontos viáveis , ou seja , com pontos interiores à região viável.

Assim , nesses métodos obtém-se novos pontos , também viáveis que gradativamente vão se aproximando da fronteira da região viável a medida que se diminue o termo de penalidade .

O termo de penalidade tem o papel de desestimular a aproximação da fronteira da região viável . Esse desestímulo é efetuado por uma " barreira " resultante da contribuição do termo penalidade na função objetivo .. Entre as vantagens , está o fato de trabalharmos na região viável, obtendo-se pelo menos uma solução viável, caso ocorra uma parada prematura.

Prova-se que, se tivermos uma sequência $\{ x^k \}$ convergente, então o limite é uma solução ótima para o problema original. Ademais é provado que $f(x^k) \to f^*$, onde $\{ x^k \}$ é a seqüência de pontos gerados pelo algorítmo, e f^* é solução ótima do problema original.

Para esse caso, a função de penalidade $P(x)$ deve satisfazer:

i) $P(x)$ é contínua

ii) $P(x) \geq 0$

iii) $P(x) \to \infty$ quando x se aproxima da fronteira de S.

O efeito da função de penalidade interior é visto na figura 2:

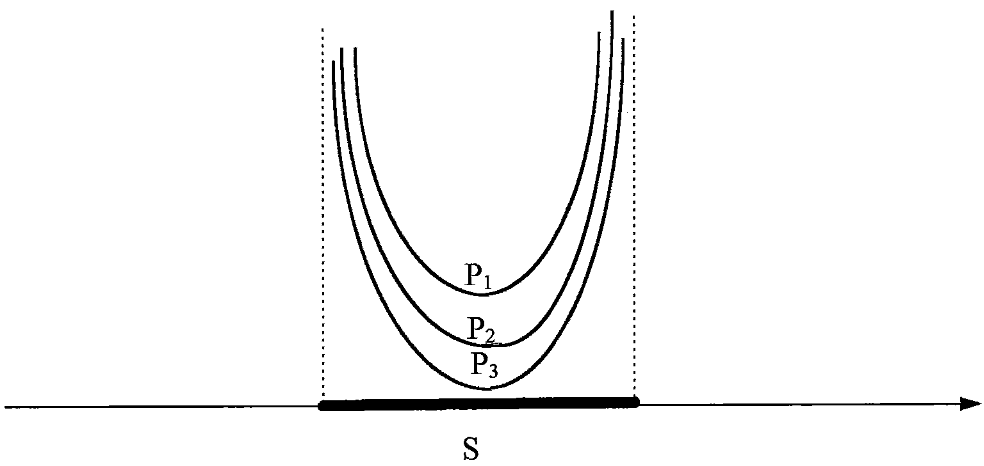

Figura 2 : Penalização Interior

O uso da função barreira foi inicialmente proposto por Frish [29] em 1955, e mais tarde por Carrol [16] em 1959 , com o nome de CRST - Created Response Surface Technique . O procedimento foi usado para resolver problemas com restrições de desigualdades por Box , Davies e Swann [14] e Kowalik [38] .

O problema numérico de como mudar os parâmetros da penalidade exterior e da penalidade interior têm sido investigado por vários autores , em particular por Fiaccco & McCormick , que estabeleceram os resultados teóricos fundamentais desses métodos em seu premiado livro [25] para uma discussão mais detalhada .

Várias extensões para os conceitos das funções de penalidade exterior e interior tem sido feitas . Primeiro a fim de evitar as dificuldades associadas com o mal-condicionamento quando os parâmetros da penalidade exterior vai para infinito , e os parâmetros da penalidade interior vai para zero , por conseguinte , vários métodos de parâmetro livre têm sido propostos . Para maiores detalhes , veja igualmente em Fiacco & McCormick [25] , o método dos centros de Huard [35] e Lootsma [40]. Uma descrição integrada desses métodos pode também ser vista em Lootsma [41] .

Considere o problema de programação restrito a desigualdades :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à:} \quad g_i(x) \geq 0 \quad ; \; i = 1, \ldots, m \end{aligned} \tag{2.29}$$

onde $f(.)$ e $g_i(.)$ são funções contínuas no R^n.

O conjunto $S^o = \{ x \; ; \; g_i(x) > 0 \; ; \; i = 1, \ldots, m \}$ deve ter interior não-vazio , isto porque , o método trabalha no interior da região .

O problema irrestrito tem a forma básica :

$$\text{minimizar} \quad f(x) + \rho P(x) , \qquad (2.30)$$
$$x \in S^o$$

A função de penalidade P (x) pode assumir diferentes formas ; por exemplo

$$P(x) = \sum_{i=1}^{m} 1/g_i(x) \quad , \qquad (2.31)$$

que é a função barreira inversa de Carrol [17] , ou

$$P(x) = -\rho \sum_{i=1}^{m} \log[g_i(x)] \quad , \qquad (2.32)$$

que é a função barreira logarítmica de Frisch [29] .

MÉTODO DE PENALIDADE MISTA

Para resolver um problema geral de programação não-linear da forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad g_i(x) \geq 0 \quad ; \; i = 1, \ldots, m \qquad (2.33) \\ &\qquad\qquad\quad h_j(x) = 0 \quad ; \; j = 1, \ldots, p \end{aligned}$$

pode-se usar um método de penalidade mista , obtida pelo uso simultâneo da penalização exterior e penalização interior . Para ilustração , a função objetivo aumentada do problema irrestrito associada ao problema (2.41),por exemplo , assume a forma :

$$F(x, \rho, \eta) = f(x) - \rho \sum_{i=1}^{m} \log g_i(x) + 1/\eta \sum_{j=1}^{p} [h_j(x)]^2 \qquad (2.34)$$

Sob certas condições , a seqüência $\{ F(x^{k*}, \rho^k, \eta^k) \}$ converge para $f(x^*)$, o valor ótimo do problema original , e uma subseqüência de $\{ x^{k*} \}$ converge para x^* . Veja Fiacco & McCormick [24 , 25] e Lootsma [41] .

Observa-se que em todos estes métodos algumas preocupações devem ser ressaltadas . A primeira delas é saber quão bem os problemas irrestritos aproximam o restrito e , se as soluções destes problemas convergem à solução ótima do problema original . Outra preocupação é quanto à dificuldades impostas ao problema irrestrito com o acréscimo da função de penalidade , e como consequência destas dificuldades , a escolha do método de minimização irrestrita deve ser pesquisada .

MÉTODO DE PENALIZAÇÃO EXATA

A idéia central no método de penalização exata é construir funções de penalidade que são exatas , no sentido de que , a solução dos problemas de penalidade, produzem a solução exata para o problema original , para um valor finito do parâmetro de penalidade.

Com estas funções não é necessário resolver uma sequência infinita de problemas de penalidade para obter a solução correta.

Um dos primeiros desses métodos é devido a Zangwill [67] usando uma função de penalidade exterior linear. Posteriormente Evans, Gould e Tolle [22], conforme exposto em Avriel [4], desenvolvem uma teoria geral sobre a penalização exata para uma classe de funções de penalidades diferenciáveis por partes.

Por exemplo, para o problema geral de programação da forma:

$$\begin{array}{lll} \text{minimizar} & f(x) & \\ \text{sujeito à:} & g_i(x) \geq 0 & ;\ i = 1, \ldots, m \\ & h_j(x) = 0 & ;\ j = 1, \ldots, p \end{array} \tag{2.35}$$

a função de penalidade valor-absoluto proposta em Zangwill [66] é:

$$P(x) = \sum_{i=1}^{m} \max[0, g_i(x)] + \sum_{j=1}^{p} |h_j(x)| , \tag{2.36}$$

onde o problema penalizado se escreve como:

$$\begin{array}{ll} \text{minimizar} & f(x) + c.P(x) , \\ x \in R^n & \end{array} \tag{2.37}$$

para alguma constante positiva c .

Zangwill [67], em 1967 prova a existência de um valor $\underline{c}$ tal que para todo $c > \underline{c}$, a solução do problema (2.37) é solução do problema original (2.35).

Todavia , essa aparentemente extraordinária vantagem desse método desaparece face à presença das não-diferenciabilidades . Resulta por essa razão , que essa abordagem tem quase unicamente um interesse de natureza teórica .

PENALIZAÇÃO EXATA DIFERENCIÁVEL

Seguindo as linhas apresentadas em Bertsekas [12] , mostraremos que é possivel construir um problema de minimização irrestrita diferenciável envolvendo minimização em x e μ , e tendo soluções ótimas que são chamadas de par KKT para o problema :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad h(x) = 0 \end{aligned} \tag{2.38}$$

onde as funções $f , h \in C^2$ no R^n . Para tal , precisamos considerar a função Lagrangeana

$$l(x,\mu) = f(x) + \mu^t h(x) ,$$

as condições necessárias para otimalidade

$$\nabla_x l(x,\mu) = 0 \quad , \quad \nabla_\mu l(x,\mu) = h(x) = 0 ,$$

e o problema de minimização irrestrito :

$$\begin{aligned} &\text{minimizar} \quad ½ \,| h(x) |^2 + ½ \,| \nabla_x l(x,\mu) |^2 \\ &(x,\mu) \in R^n \times R^p \end{aligned} \tag{2.39}$$

Observa-se que (x^*, μ^*) é um par KKT para o problema original se, e somente se, (x^*, μ^*) é um mínimo global do problema irrestrito. Logo é possível encontrar uma solução para o problema restrito a igualdades, resolvendo-se o problema irrestrito dado acima.

Uma dificuldade dessa abordagem, é que a distinção entre mínimo local e máximo local do problema (2.38) é completamente perdida, quando da passagem para o problema irrestrito (2.39).

Como uma alternativa para o problema (2.39), podemos considerar o problema:

$$\underset{(x,\mu) \in R^n \times R^p}{\text{minimizar}} \quad l(x,\mu) + \tfrac{1}{2}\mathbf{r}\,|h(x)|^2 + \tfrac{1}{2}\mathbf{s}\,|\nabla_x l(x,\mu)|^2 \qquad (2.40)$$

onde **r** e **s** são parâmetros positivos.

Como motivação para isso, mencionamos o fato de que, para todo $\mathbf{r} > 0$ e $\mathbf{s} > 0$ se (x^*, μ^*) é qualquer par KKT para o problema (2.38), então o par (x^*, μ^*) é um ponto crítico da função objetivo do problema (2.40).

Nossa esperança, no entanto, é que introduzindo na função objetivo a função Lagrangeana $l(x, \mu)$, e escolhendo-se apropriadamente **r** e **s**, podemos construir no problema irrestrito (2.40) uma preferência para mínimo local em vez de máximo local.

A relação entre os pontos críticos da função objetivo do problema (2.40) com os pares KKT do problema (2.38) é dado no teorema seguinte :

TEOREMA 1 :

Seja $X \times \Lambda$ um subconjunto compacto de $R^n \times R^p$. Suponha que $\nabla h(x)$ tenha posto **p** , para todo $x \in X$. Existe um escalar $\bar{\alpha} > 0$, e para cada $\alpha \in (0, \bar{\alpha}]$ um escalar $\bar{c} > 0$ tal que , para todo c e α com $\alpha \in (0, \bar{\alpha}]$ e $c \geq \bar{c}(\alpha)$, todo ponto crítico da função objetivo de (2.40) pertencendo ao sub-conjunto $X \times \Lambda$ é um par KKT para o problema (2.38) .

Se $\nabla^2_{xx} l(x, \mu)$ é semidefinida positiva para todo par (x, μ) pertencente à $X \times \Lambda$, então $\bar{\alpha}$ pode ser tomado como qualquer escalar positivo .

Demostração : Ver Bertsekas [12] .

Essa abordagem apresentada por Bertsekas é um dos caminhos que nos leva aos métodos Lagrangeanos aumentados .

LAGRANGEANO AUMENTADO OU DOS MULTIPLICADORES

Os métodos Lagrangeanos Aumentados também conhecidos como métodos dos multiplicadores , podem ser vistos como uma extensão da idéia da função de penalidade, com a perspectiva de evitar a necessidade do parâmetro ρ de penalidade definido abaixo na equação (2.42) se tornar muito grande . Consideremos o problema de programação não-linear com restrições de igualdades:

$$\text{minimizar } f(x) \tag{2.41}$$

$$\text{sujeito à : } h_j(x) = 0 \quad ; j = 1, \dots, p$$

Para este problema a função objetivo modificada do problema irrestrito correspondente à penalização exterior de Courant [20] se apresenta na forma :

$$\text{minimizar} \quad f(x) + \rho \sum_{j=1}^{p} [h_j(x)]^2 \quad , \tag{2.42}$$

cuja solução $x(\rho) \to x^*$ quando $\rho \to \infty$.

Por outro lado , as condições de primeira ordem estabelecem :

$$\nabla f(x^*) + \mu^{*t} \nabla h(x^*) = 0$$

em qualquer ponto ótimo x^* . Segue-se que , o gradiente da função

$$f(x) + \sum_{j=1}^{p} \mu_j^* h_j(x) + \tfrac{1}{2} \rho \sum_{j=1}^{p} [h_j(x)]^2 \tag{2.43}$$

é zero no par (x^*, μ^*) , para todos os valores de ρ . Portanto , parece ser melhor substituir a função (2.42) pela função :

$$L(x, \mu, \rho) = f(x) + \sum_{j=1}^{p} \mu_j h_j(x) + \frac{1}{2} \rho \sum_{j=1}^{p} [h_j(x)]^2 \qquad (2.44)$$

onde $\rho > 0$ é o parâmetro de penalidade, e μ é o multiplicador de Lagrange .

A função (2.44) definida acima, é a função Lagrangeana Aumentada proposta por Hestenes [32].

O método de penalidade quadrática à função Lagrangeana proposto por Hestenes, consiste em resolver uma seqüência de problemas da forma :

$$\begin{array}{l} \text{minimizar} \quad L(x, \mu_k, \rho^k) \\ x \in R^n \end{array} \qquad (2.45)$$

onde $\{ \mu_k \}$ é uma sequência limitada do R^n, e $\{ \rho^k \}$ é uma seqüência de parâmetros satisfazendo $0 < \rho^k < \rho^{k+1} \quad \forall k$, $\rho^k \to \infty$.

Observa-se na versão original dos métodos de penalidades, os multiplicadores μ_k são tomados iguais a zero, isto é, $\mu_k = 0 \quad \forall\ k = 0, 1, \ldots$

O resultado que μ^* é a escolha ótima do vetor parâmetro de controle na função proposta por Hestenes é expresso no seguinte :

TEOREMA 2 :

Se as condições de suficiência de segunda ordem valem num par ótimo (x^* , μ^*) , então existe $\rho' \geq 0$ tal que , para qualquer $\rho \geq \rho'$, x^* é um minimizador local da função $L(x, \mu^*, \rho)$, isto é , $x^* = x(\mu^*)$.

Deonstração : Fletcher [27] .

Já Powell [51] , de uma forma independente , propôs a seguinte função :

$$L(x, \theta, \tau) = f(x) + \sum_{j=1}^{p} \tau.(h_j(x) + \theta_j)^2, \tag{2.46}$$

onde θ_j é uma tolerância à violação a cada restrição e τ é o parâmetro de penalidade , que é absolutamente equivalente à formulação de Hestenes (eq . 2.8) . Da mesma forma , Haarhoff e Buys [31] propuseram uma outra equação equivalente .

O método Lagrangeano Aumentado pode ser também aplicado a problemas restritos a desigualdades . Por exemplo : Rockafellar [55 , 56] sugeriu a seguinte função :

$$L(x, \lambda, \rho) = f(x) + \sum \begin{cases} \lambda_i g_i(x) + \frac{\rho}{2} g_i(x)^2 & ; g_i(x) \leq \frac{\lambda_i}{2} \\ -\frac{\rho}{2} \lambda_i^2 & ; caso.\, contrario \end{cases} \tag{2.47}$$

Em outro trabalho , Rockafellar [54] , para o caso do problema de programação convexa com restrições de desigualdades mostra que , um ponto de sela irrestrito da função (2.11) corresponde a uma solução do problema de programação convexo original .

Como vimos , o Método dos Multiplicadores ou do Lagrangeano Aumentado para problemas de programação não-linear restrito a igualdades, foi independentemente introduzido por Hestenes [32] , Powell [52] e Haarhoff & Buys [31] .Eles propuseram um método dual de solução no qual quadrados das funções restrições são adicionados à função Lagrangeana.

A característica associada a esse método é a minimização irrestrita de uma função objetivo L (.) contínua , que envolve multiplicadores de Lagrange e um termo de penalidade . O termo de penalidade tem por finalidade fazer x* um ponto de mínimo irrestrito .

A expressão da função Lagrangeana Aumentada é dada por :

$$L(\,.\,) = l(\,.\,) + P(\,.\,) \qquad (2.48)$$

onde l (.) é a função Lagrangeana simples e P (.) é a função de Penalidade .

Uma série de minimizações irrestritas sobre a função Lagrangeana aumentada ou penalizada é seguida de uma atualização do vetor de multiplicadores de acordo com uma regra simples. Powell [52] mostra que , se as condições de suficiência de segunda ordem para otimalidade são satisfeitas, o algorítmo converge localmente numa taxa linear.

A vantagem do algorítmo é que ele produz uma estabilidade numérica que não é encontrada nos métodos de penalidades usuais . Por outro lado, Miele et al [44 , 45 , 46 , 47] modificou e melhorou os métodos de Hestenes e Powell e , tem obtido resultados computacionais muito bons para o caso restrito a igualdades .

Fletcher [27 , 28] desenvolveu uma outra técnica relativa às de Hestenes e Powell , que em vez de atualizar o vetor dos multiplicadores após uma sequência de minimizações , o vetor dos multiplicadores é ajustado simultaneamente ao processo de minimizações sobre as Lagrangeanas penalizadas .

Já , Kort & Bertsekas [37] consideraram o caso mais geral em que o problema apresenta simultaneamente restrições de igualdades e desigualdades em seus métodos de multiplicadores para programação convexa .

Arrow , Gould & Howe [3] estudaram a função Lagrangeana aumentada de Rockafellar , e a incluiu numa classe geral para problemas de programação não-convexa, onde estabeleceram propriedades locais de ponto de sela . Já Pierre & Lowe [50] sugeriram algorítmos de multiplicadores, considerando restrições de igualdades e desigualdades , utilizando função Lagrangeana aumentada , e mostra propriedades de convergência local .

VANTAGENS DO MÉTODO DOS MULTIPLICADORES COM RELAÇÃO AOS MÉTODOS DE PENALIDADES

Uma vantagem do método dos multiplicadores é que não é necessário fazer os parâmetros de penalidades ρ^k ir para o infinito a fim de obtermos convergência , assim evitando ou moderando a conhecida dificuldade do mal-condicionamento do problema penalizado .

Uma segunda vantagem do método dos multiplicadores é que sua taxa de convergência é consideravelmente melhor do que a dos métodos de penalidades , isto é , nos métodos dos multiplicadores , a taxa de convergência é linear ou superlinear , enquanto que , nos métodos de penalidades sua taxa de convergência é muito inferior e depende essencialmente da taxa que o parâmetro de penalidade é aumentado.

Esta vantagem em velocidade de convergência tem sido verificada em muitos estudos computacionais , onde uma redução consistente em tempo de computação passou de 80% para 30% em certos problemas de programação envolvendo restrições de igualdades , quando são atualizados via

$$\lambda_{k+1} = \lambda_k + \rho^{k} h(x^k)$$

com relação ao método de penalidade , onde λ_k é mantido constante em todas as iterações .Por outro lado , uma desvantagem do método Lagrangeano é que ele requer um bom ponto inicial a fim de obtermos a convergência para uma solução ótima . Para aumentarmos a região de convergência é necessário fazermos combinações com outros métodos , basicamente do tipo de região de confiança , para assim obtermos propriedade de convergência global .

PENALIZAÇÃO HIPERBÓLICA

Na medida que este trabalho se fundamenta numa extensão da Penalização Hiperbólica , apresentaremos como preâmbulo este método com um particular detalhamento.

O método de penalização hiperbólica apresentado por Xavier [62] trata de resolver o problema de programação não-linear com restrições de desigualdades da forma :

$$\begin{aligned} &\text{minimizar} \;\; f(x) \\ &\text{sujeito à}: \;\; g_i(x) \geq 0 \quad ; \qquad i = 1, \ldots, m \end{aligned} \tag{2.49}$$

Sua solução é obtida através da resolução de uma sequência de problemas de minimizações irrestrito da função objetivo $f(x)$ acrescida de um termo de penalidade $P(x, \alpha, \tau)$ para cada restrição, isto é:

$$\underset{x \in R^n}{\text{minimizar}} \quad F(x, \alpha^k, \tau^k) = f(x) + \sum_{i=1}^{m} P(g_i(x), \alpha^k, \tau^k) \, , \tag{2.50}$$

onde a função de penalidade $P(.)$ é definida por :

$$P(g_i, \alpha_i, \tau_i) = -\tfrac{1}{2} \sum_{i=1}^{m} \operatorname{tg} \alpha_i . g_i + \sum_{i=1}^{m} \sqrt{\left(\tfrac{1}{2} \operatorname{tg}\alpha_{i} . g_i\right)^2 + \tau_i^2} \tag{2.51}$$

com $\qquad \alpha_i \in [\, 0, \pi/2 \,)$ e $\tau_i \geq 0$.

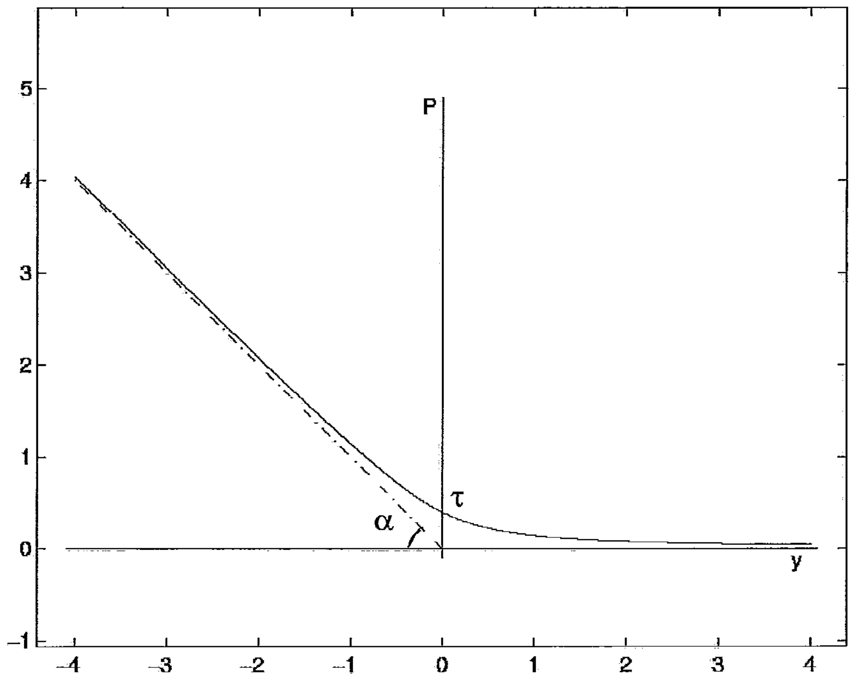

Figura 3 : Função Penalidade Hiperbólica

IDÉIA GEOMÉTRICA DO ALGORÍTMO

Aumentando-se o ângulo α da assíntota da função penalidade , provoca-se um significativo aumento da penalização fora da região viável, enquanto que, simultaneamente, reduz-se a penalização para pontos dentro da região viável . O processo continua até que se consiga um ponto viável. Veja figura 4 .

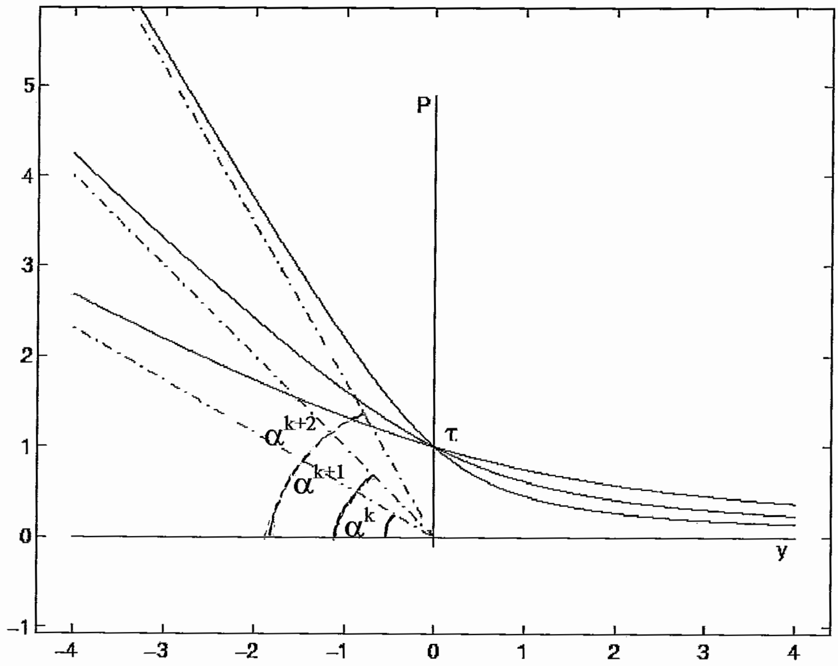

Figura 4 : Aumento do parâmetro α

A partir daí, mantém-se o ângulo α constante e diminue-se sequencialmente o valor da distância τ . Dessa maneira consegue-se que a penalização interior torne-se cada vez mais irrelevante , mantendo-se o mesmo nível de proibitividade fora da região viável. Este fato pode ser visto na figura 5 .

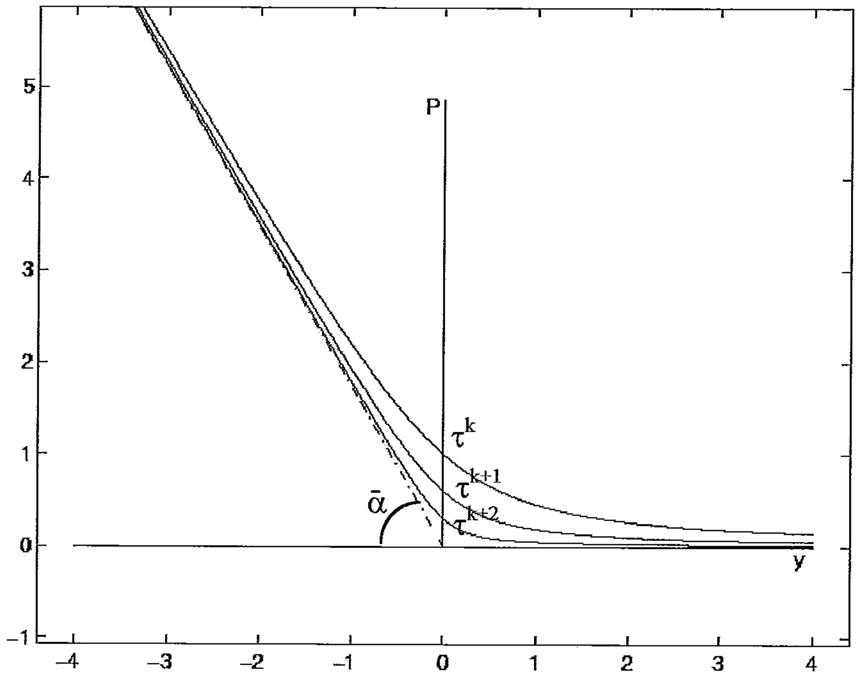

Figura 5 : Diminuição do parâmetro τ

Algumas características fundamentais deste método , por exemplo , mostram que quando se manipula o parâmetro α , a penalização hiperbólica se comporta simultaneamenta aos métodos de penalização exterior , e quando se manipula o parâmetro τ o seu comportamento torna-se similar aos métodos de penalização interior .

Uma outra forma para a função de penalidade P (.) se dá quando definimos :

$$\lambda_i = \tfrac{1}{2} \operatorname{tg} \alpha_i \quad , \qquad (2.52)$$

obtendo-se a expressão :

$$P(g_i, \lambda_i, \tau_i) = - \sum_{i=1}^{m} \lambda_i . g_i + \sum_{i=1}^{m} \sqrt{(\lambda_i . g_i)^2 + \tau_i^2} \qquad (2.53)$$

com $\lambda_i \geq 0$ e $\tau_i \geq 0$.

De maneira simplificada apresentamos o algorítmo **0** de penalização hiperbólica proposto por Xavier [64] , para resolver o problema de programação não-linear com restrições de desigualdades .

ALGORÍTMO 0

1) Faça $k = 0$. Tome valores iniciais x^o , $\lambda^1 > 0$ e $\tau^1 > 0$

2) Faça $k := k+1$.

Resolva o problema de minimização irrestrito

$$\text{minimizar}_x \; L_H (x , \lambda^k , \tau^k)$$

a partir do ponto inicial x^{k-1} obtendo um ponto ótimo intermediário x^k.

3) Atualizar os multiplicadores de Lagrange :

$$\lambda_i^{k+1} = \lambda_i^k \left[1 - \frac{\lambda_i^k g_i(x^k)}{\sqrt{\left(\lambda_i^k g_i(x^k)\right)^2 + (\tau_i^k)^2}} \right]$$

4) Atualizar os parâmetros de penalidades τ

$$\tau_i^{k+1} = q\tau_i^k \quad ; \quad 0 < q \leq 1$$

Vá para o passo 2 .

□

Pode-se observar que existe uma ligação deste método com a função Lagrangeana, e por consequência, com as condições de otimalidade. Temos a função Lagrangeana relativa ao problema (2.49) definida como :

$$l(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i . g_i(x) , \qquad (2.54)$$

e com alguns cálculos chega-se à forma :

$$F(x, \lambda, \tau) = L_H(x, \lambda, \tau) = l(x, \lambda) + \sum_{i=1}^{m} \sqrt{(\lambda_i . g_i)^2 + \tau_i^2} \quad , \qquad (2.55)$$

onde $L_H(.)$ é a função Lagrangeana Hiperbólica .

Dessa forma podemos obsevar que a função objetivo modificada corresponde a um função Lagrangeana aumentada ou função Lagrangeana modificada , como preferem outros autores .

Vejamos agora uma propriedade da função de penalização hiperbólica que relaciona a variação de sua inclinação com a variação do parâmetro λ desta função.

PROPOSIÇÃO 1 :

Para valores $y < 0$, o aumento do parâmetro λ sempre corresponde a um decréscimo da derivada da função de penalização $P(y, \lambda, \tau)$ em relaçào a y , (ou equivalentemente , um aumento da inclinação da penalidade .

Enquanto para um certo valor $y > 0$ fixo , e dado $\tau > 0$, existe um valor $\bar{\lambda}$ tal que , para $\lambda \geq \bar{\lambda}$, a derivada da função penalidade $P(y, \lambda, \tau)$ em relação a y é uma função crescente com λ .

Demonstração :

Primeiramente façamos um estudo do comportamento da inclinação da função penalidade em relação a modificação em seu parâmetro λ . Para facilitar a notação, vamos escolher a função penalidade dada por:

$$P(y, \lambda, \tau) = -\lambda y + \sqrt{(\lambda . y)^2 + \tau^2} \qquad (2.56)$$

Calculando-se a derivada da função penalidade em relação a y , teremos :

$$\frac{dP(y,\lambda,\tau)}{dy} = \lambda \left[-1 + \frac{\lambda . y}{\sqrt{(\lambda . y)^2 + \tau^2}} \right] \qquad (2.57)$$

Pelo simples exame da expressão (2.57) acima podemos observar que esta derivada varia no intervalo (- 2λ , 0) . Agora, vamos estudar o comportamento dessa derivada com relação ao parâmetro λ . Calculando-se a derivada da expressão (2.57) em relação a λ teremos :

$$\frac{d^2 P(y,\lambda,\tau)}{dy d\lambda} = -1 + \frac{\lambda^3 . y^3 + 2\lambda . y . \tau^2}{\left[(\lambda . y)^2 + \tau^2 \right]^{3/2}} \qquad (2.58)$$

Pela análise da expressão (2.58) acima , e considerando que λ ≥ 0 , podemos ver que , para valores de y < 0 , ou seja , para pontos inviáveis , a expressão (2.58) é negativa . Vale dizer : a derivada da função penalização em relação a y diminui crescentemente com λ para pontos inviáveis , ou seja , a inclinação da curva se torna crescentemente mais escarpada com o aumento de λ .

Por outro lado , para valores de y > 0 , que correspondem a pontos viáveis , temos numa faixa inicial de y , a expressão (2.58) negativa , o que corresponde a um comportamento da penalização análogo ao descrito anteriormente. Todavia , para valores maiores que y será observada que a expressão (2.58) é positiva , e o comportamento será exatamente o contrário , ou seja , a inclinação se reduz com o aumento de λ .

Calculamos agora , para um ponto y > 0 fixo , qual é exatamente o valor de λ em que ocorre a transição entre esses dois comportamentos acima descritos . Devemos assim , achar as raízes da equação :

$$\frac{d^2 P(y,\lambda,\tau)}{dy d\lambda} = 0 \qquad (2.59)$$

Da expressão (2.58) obtém-se a equação :

$$y^4 \lambda^4 \tau^2 + y^2 \lambda^2 \tau^4 - \tau^6 = 0 , \qquad (2.60)$$

que tem uma única solução com significado ($\lambda \geq 0$), a raiz :

$$\bar{\lambda} = \frac{\tau}{y} \sqrt{\frac{-1+\sqrt{5}}{2}} = \frac{\tau}{y} \beta \quad , \qquad (2.61)$$

onde

$$\beta = \sqrt{\frac{-1+\sqrt{5}}{2}} \qquad (2.62)$$

CAPÍTULO 3

RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO NÃO-LINEAR COM RESTRIÇÕES DE IGUALDADES VIA PENALIZAÇÃO HIPERBÓLICA

O objetivo deste trabalho é resolver o problema de programação não-linear com restrições de igualdades da forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad h_j(x) = 0 \quad ; \quad j = 1,\dots,p \end{aligned} \tag{3.1}$$

onde as funções $f(.) : R^n \to R$ e $h_j(.) : R^n \to R$ são contínuas .

A estratégia a ser utilizada para resolver o problema acima é baseada em um novo método derivado do algorítmo de Penalização Hiperbólica , Xavier [62] .

Este algorítmo se destina à resolução de problemas de programação não -linear restrito somente a desigualdades, cuja solução é obtida através da resolução de uma seqüência de problemas de minimizações irrestritas da forma (2.50) como descrito anteriormente.

Sendo assim , a primeira providência é transformar cada restrição de igualdade do problema (3.1) em duas restrições de desigualdades produzindo um problema equivalente da forma :

$$\text{minimizar} \quad f(x) \tag{3.2}$$

$$\text{sujeito à :} \quad h_j(x) \leq 0 \quad ; \quad j = 1, \ldots, p \tag{3.2.a}$$

$$h_j(x) \geq 0 \quad ; \quad j = 1, \ldots, p \tag{3.2.b}$$

Supondo-se por hipótese que todos os multiplicadores são positivos, então o problema definido abaixo por :

$$\text{minimizar} \quad f(x) \tag{3.3}$$

$$\text{sujeito à :} \quad h_j(x) \leq 0 \quad ; \quad j = 1, \ldots, p$$

é equivalente ao problema anterior (3.2). Nesta hipótese , equivalentemente , estamos supondo que as restrições (3.2.b) são redundantes , isto é , os multiplicadores correspondentes são nulos . Dessa maneira , podemos acrescentar ao problema (3.3) restrições relaxadas não-ativas obtendo-se assim o problema equivalente :

$$\text{minimizar} \quad f(x) \tag{3.4}$$

$$\text{sujeito à :} \quad h_j(x) \leq 0 \quad ; \quad j = 1, \ldots, p$$

$$h_j(x) \geq \varepsilon_j^- \quad ; \quad j = 1, \ldots, p \quad ,$$

sendo $\varepsilon_j^- \leq 0$, $j = 1, \ldots, p$.

Como o algorítmo de penalização hiperbólica assume a hipótese de que a região viável tenha interior não-vazio , para a sua utilização , no problema (3.2) devemos relaxar as restrições colocando folgas superiores ε_j^+ e inferiores ε_j^- . Assim obtemos o problema :

$$\text{minimizar} \quad f(x) \tag{3.5}$$

$$\text{sujeito à :} \quad h_j(x) \geq \varepsilon_j^- \quad ; \quad j = 1, \ldots, p \tag{3.5.a}$$

$$h_j(x) \leq \varepsilon_j^+ \quad ; \quad j = 1, \ldots, p \ , \tag{3.5.b}$$

com $\varepsilon_j^+ \geq 0$ e $\varepsilon_j^- \leq 0$, que nos dá condições necessárias para utilizarmos o algorítmo da penalidade hiperbólica.

A figura abaixo mostra o efeito provocado pela relaxação quando se trata do problema (3.5)

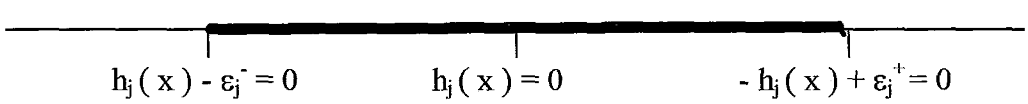

Figura 6 : Relaxação das restrições de igualdades

ou seja , provoca uma região viável com interior não-vazio , associada a cada restrição de igualdade .

Assim fazendo , vamos resolver uma sequência de problemas irrestritos da forma :

$$\underset{x \in R^n}{\text{minimizar}} \quad F(x, \alpha^k, \tau^k) = f(x) + \sum_{j=1}^{m} P(g_j(x), \alpha^k, \tau^k) \tag{3.6}$$

onde :

$$g_j(x) = h_j(x) - \varepsilon_j^- \geq 0 \quad ; \quad j = 1, \dots, p \tag{3.6.a}$$

e

$$g_{p+j}(x) = -h_j(x) + \varepsilon_j^+ \geq 0 \quad ; \quad j = 1, \dots, p \tag{3.6.b}$$

com m = 2p . A função de penalidade P(.) definida de acordo com a expressão (2.51) acima citada , tem a forma :

$$P(g_i(x), \alpha, \tau) = -½ \sum_{j=1}^{m} tg\, \alpha_i . g_i(x) + \sum_{j=1}^{m} \sqrt{\left(\tfrac{1}{2} tg\alpha_i . g_i(x)\right)^2 + \tau_i^2} \tag{3.7}$$

Alternativamente a função objetivo do problema irrestrito pode ser escrita como :

$$F(x, \alpha, \tau) = f(x) - ½ \sum_{j=1}^{p} tg\alpha_j . (h_j(x) - \varepsilon_j^-) - ½ \sum_{j=p+1}^{m} tg\alpha_j . (-h_j(x) + \varepsilon_j^+) +$$
$$\sum_{j=1}^{p} \sqrt{\left[\frac{1}{2} tg\alpha_j . (h_j(x) - \varepsilon_j^-)\right]^2 + \tau_j^2} + \sum_{j=p+1}^{m} \sqrt{\left[\frac{1}{2} tg\alpha_j . (-h_j(x) + \varepsilon_j^+)\right]^2 + \tau_j^2} \tag{3.8}$$

onde o gradiente da função $F(x, \alpha, \tau)$ é dado por :

$$\nabla_x F(x, \alpha, \tau) = \nabla f(x) - \tfrac{1}{2} \sum_{j=1}^{2p} tg\alpha_j . \left\{ \left[1 - \frac{\frac{1}{2}\, tg\alpha_j (h_j(x) - \varepsilon_j^-)}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j (h_j(x) - \varepsilon_j^-)\right]^2 + \tau_j^2}} \right] - \right.$$

$$\left. \left[1 - \frac{\frac{1}{2}\, tg\alpha_j (-h_j(x) + \varepsilon_j^+)}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j (-h_j(x) + \varepsilon_j^+)\right]^2 + \tau_j^2}} \right] \right\} . \nabla h_j(x) \qquad (3.9)$$

No algorítmo de penalização hiperbólica , como descrito em Xavier [64] , na resolução da sequência de problemas irrestritos simultaneamente à correspondente sequência de pontos de mínimos { x^k } também é gerada uma sequência de estimativas dos multiplicadores de Lagrange .

Assim , denominando respectivamente μ_j^k e η_j^k os multiplicadores associados às restrições (3.6a) e (3.6b) .

Na iteração k teremos :

$$\mu_j^k = \tfrac{1}{2}\, tg\alpha_j^{\,h} . \left[1 - \frac{\frac{1}{2}\, tg\alpha_j^k (h_j(x^k) - \varepsilon_j^{k-})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k (h_j(x^k) - \varepsilon_j^{k-})\right]^2 + \left(\tau_j^k\right)^2}} \right] \qquad (3.9.a)$$

$$\eta_j^k = \tfrac{1}{2}\, tg\alpha_j^{\,k} . \left[1 - \frac{\frac{1}{2}\, tg\alpha_j^k (-h_j(x^k) + \varepsilon_j^{k+})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k (-h_j(x^k) + \varepsilon_j^{k+})\right]^2 + \left(\tau_j^k\right)^2}} \right] \qquad (3.9.b)$$

Pela análise das expressões (3.9a) e (3.9b) podemos ver que sempre serão observadas as relações $\mu_j^k > 0$ e $\eta_j^k > 0$.

Na expressão (3.9) , vamos definir :

$$\lambda_j^k = \mu_j^k - \eta_j^k \quad , \qquad (3.10)$$

que denominamos multiplicador de Lagrange equivalente .

Veremos uma interessante associação entre o sinal dos multiplicadores de Lagrange equivalentes e os valores das restrições .

PROPOSIÇÃO 2 :

$$\text{Se} \quad \lambda_j^k > 0 \quad , \quad \text{então} \qquad - h_j(x^k) + \varepsilon_j^+ > h_j(x^k) - \varepsilon_j^-$$

Demonstração :

Por hipótese temos : $\lambda_j^k > 0$, logo da expressão (3.10) obtemos :

$$\mu_j^k - \eta_j^k > 0 \quad \Rightarrow$$

$$\tfrac{1}{2}\, tg\alpha_j{}^h . \left[1 - \frac{\frac{1}{2}\, tg\alpha_j (h_j(x^k) - \varepsilon_j^{k-})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k (h_j(x^k) - \varepsilon_j^{k-})\right]^2 + \left(\tau_j^k\right)^2}} \right] \quad -$$

$$\tfrac{1}{2}\, tg\alpha_j{}^k . \left[1 - \frac{\frac{1}{2}\, tg\alpha_j^k (-h_j(x^k) + \varepsilon_j^{k+})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k (-h_j(x^k) + \varepsilon_j^{k+})\right]^2 + \left(\tau_j^k\right)^2}} \right] \quad > \quad 0 \qquad \Rightarrow$$

$$\frac{\frac{1}{2}\, tg\alpha_j^k(-h_j(x^k) + \varepsilon_j^{k+})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k.\,(-h_j(x^k) + \varepsilon_j^{k+})\right]^2 + \left(\tau_j^k\right)^2}} > \frac{\frac{1}{2}\, tg\alpha_j^k(h_j(x^k) - \varepsilon_j^{k-})}{\sqrt{\left[\frac{1}{2}\, tg\alpha_j^k.\,(h_j(x^k) - \varepsilon_j^{k-})\right]^2 + \left(\tau_j^k\right)^2}}$$

Fazendo-se :

$$u = \tfrac{1}{2}\, tg\, \alpha_j^k\, [\, - h_j\, (x^k) + \varepsilon_j^{k+}\,] > 0$$

e

$$v = \tfrac{1}{2}\, tg\, \alpha_j^k\, [\, h_j\, (x^k) - \varepsilon_j^{k-}\,] > 0$$

podemos escrever a desigualdade anterior como :

$$\frac{u}{\sqrt{u^2 + \left(\tau_j^k\right)^2}} > \frac{v}{\sqrt{v^2 + \left(\tau_j^k\right)^2}} \Rightarrow$$

$$u^2.\,[\, v^2 + (\tau_j^k)^2\,] > v^2.\,[\, u^2 + (\tau_j^k)^2\,] \Rightarrow$$

$$u^2 > v^2 \Rightarrow u^2 - v^2 > 0 \Rightarrow$$

$$(u + v).(u - v) > 0$$

Como $u + v > 0$, então a desigualdade anterior nos dá :

$$u - v > 0 \Rightarrow u > v ,$$

ou seja :

$$\tfrac{1}{2}\, tg\, \alpha_j^k\, [\, - h_j\, (x^k) + \varepsilon_j^{k+}\,] > \tfrac{1}{2}\, tg\, \alpha_j^k\, [\, h_j\, (x^k) - \varepsilon_j^{k-}\,] \Rightarrow$$

$$- h_j\, (x^k) + \varepsilon_j^{k+} > h_j\, (x^k) - \varepsilon_j^{k-} \qquad \square$$

Como vimos , a relaxação de cada restrição de igualdade implica porém numa duplicação de restrições a serem consideradas, $m = 2p$. Numa primeira vista, este aumento do número de restrições poderia comprometer seriamente essa estratégia, podendo até mesmo inviabilizá-la . Todavia, devemos esclarecer, que em métodos do tipo penalidades ou lagrangeanos , normalmente a maior parte dos custos computacionais decorrem dos custos de avaliação das funções intervenientes no problema e de seus gradientes . Como no problema (3.5) , as restrições dadas por (3.5.a) e (3.5.b) envolvem as mesmas funções, podemos concluir que essa proposta não implica num aumento do custo computacional proporcional ao aumento do número de restrições.

Durante a resolução da seqüência de minimizações (3.6), pretende-se efetuar uma conveniente manipulação das folgas ε_j^+ e ε_j^- de modo a buscar uma crescente aproximação do problema relaxado (3.5) ao problema original (3.1), no limite, obtendo-se a equivalência . Além do mais , espera-se que nesse processo iterativo , a partir de um certo ponto somente um dos lados das retrições (3.5.a) ou (3.5.b) seja violado , permitindo o reajuste repetitivo da correspondente folga , de sorte a provocar sua convergência a zero enquanto que a outra folga permanece constante.

Na verdade, esse mecanismo acima idealizado equivale ao uso de uma estratégia de definição de conjunto ativo (correspondente às folgas que convergem a zero) mantendo todavia salvaguardas como proteção a uma definição equivocada desse conjunto ativo. Essas salvaguardas são desempenhadas pelas penalizações correspondentes ao lado oposto (lado em que a folga permanece constante) .

Inspirado nas idéias acima expostas surge de uma forma natural o Algorítmo I, destinado a resolver o problema de programação não-linear com restrições de igualdades (3.1), que apresentamos a seguir :

ALGORÍTMO I

PENALIZAÇÃO HIPERBÓLICA PARA RESTRIÇÕES DE IGUALDADES

1 - "INICIALIZAÇÃO"

Faça $k = 0$, $\alpha_j^1 = \alpha^o$, $j = 1, \ldots, p$, $\tau^1 = \tau^o$, sendo $0 < \alpha^o < \pi/2$ e $\tau^o > 0$.

Tome ponto inicial x^o

$$\text{Calcule : } \begin{cases} \varepsilon_j^- = -100.\,\tau^1 & , \quad j = 1, \ldots, p \\ \varepsilon_j^+ = +100.\,\tau^1 & , \quad j = 1, \ldots, p \end{cases}$$

2 - "CONTAGEM DA ITERAÇÃO"

Faça $k := k + 1$

3 - "RESOLUÇÃO DO SUBPROBLEMA"

Resolver o problema de minimização irrestrita

$$\underset{x \in R^n}{\text{minimizar}} \; F(x, \alpha^k, \tau^k) \quad ,$$

conforme definido em (2.50) a partir de um ponto inicial x^{k-1} , achando um ponto ótimo intermediário x^k .

4 - "TESTE DE VIABILIDADE"

Teste se o ponto x^k obtido é viável para o problema relaxado (3.5)

Se for viável vá para o passo 6.

5 - "AUMENTO DO PARÂMETRO ÂNGULO α DA PENALIZAÇÃO "

$$\text{Faça :} \begin{cases} \alpha_j^k := \rho\alpha_j^k + (1-\rho)\dfrac{\pi}{2} \quad , \quad j = 1,\dots,p \quad , \quad 0 < \rho < 1 \\ \\ \textit{Va para o passo } 3 \end{cases}$$

6 - "DIMINUIÇÃO DO PARÂMETRO DISTÂNCIA τ DA PENALIZAÇÃO"

$$\text{Faça :} \begin{cases} \tau^{k+1} = q\,\tau^k \\ \alpha^{k+1} = \alpha^k \end{cases} \quad , \; 0 < q < 1 \quad , \quad q = 1/10$$

7 - "CÁLCULO DAS DISTÂNCIAS SIGNIFICATIVAS"

Calcule as distâncias à esquerda e à direita do valor $h_j(x^k)$ às folgas.

$$d_j^- = + h_j(x^k) - \varepsilon_j^{k-} \qquad ; \; j = 1, \dots, p$$

$$d_j^+ = - h_j(x^k) + \varepsilon_j^{k+} \qquad ; \; j = 1, \dots, p$$

8 - "TESTES DE VIOLAÇÃO PARA CADA RESTRIÇÃO"

Para cada restrição $j = 1, \dots, p$ faça :

8A - : (" Se o ponto x^k for tal que a distância à direita for significativamente maior do que a distância à esquerda , processa a diminuição da folga inferior ").

Se [$d_j^+ > d_j^-$] e [$h_j(x^k) < 2\beta.\tau^k$] , com $\beta = \sqrt{\dfrac{-1+\sqrt{5}}{2}}$

definido em (2.62)

Então faça : $$\begin{cases} \varepsilon_j^{k+1^-} = \dfrac{\varepsilon_j^{k\,-}}{10} & (8A.1) \\ \varepsilon_j^{k+1\,+} = \varepsilon_j^{k\,+} & (8A.2) \end{cases}$$

8B : ("Se o ponto x^k for tal que a distância à esquerda for significativamente maior do que a distância à direita , processa a diminuição da folga superior ") .

Se [$d_j^- > d_j^+$] e [$h_j(x^k) > -2\beta.\tau^k$] , com $\beta = \sqrt{\dfrac{-1+\sqrt{5}}{2}}$

Então faça : $$\begin{cases} \varepsilon_j^{k+1^+} = \dfrac{\varepsilon_j^{k\,+}}{10} & (8B.1) \\ \varepsilon_j^{k+1\,-} = \varepsilon_j^{k\,-} & (8B.2) \end{cases}$$

8C - : "Diminuição de ambas as folgas se não houver proximidade significativa"

Em caso contrário,

$$
\text{Faça :}\quad
\begin{cases}
\varepsilon_j^{k+1\,-} = -100\,\tau^k & (8C.1)\\
\varepsilon_j^{k+1\,+} = +100\,\tau^k & (8C.2)\\
\alpha_j^{k+1} = \rho\alpha_j^k + (1-\rho)\dfrac{\pi}{2} \quad , \; 0 < \rho < 1 & (8C.3)
\end{cases}
$$

Vá para o passo 2 .

□

DISCUSSÃO INICIAL DO ALGORÍTMO

A estrutura do algorítmo é constituída de tal forma que , para cada restrição o intervalo entre as folgas (ε_j^+ - ε_j^-) seja reduzido monotonamente .

Este procedimento tem como perspectiva forçar uma aproximação gradativa dos pontos de mínimos intermediários x^k ao ponto ótimo x*

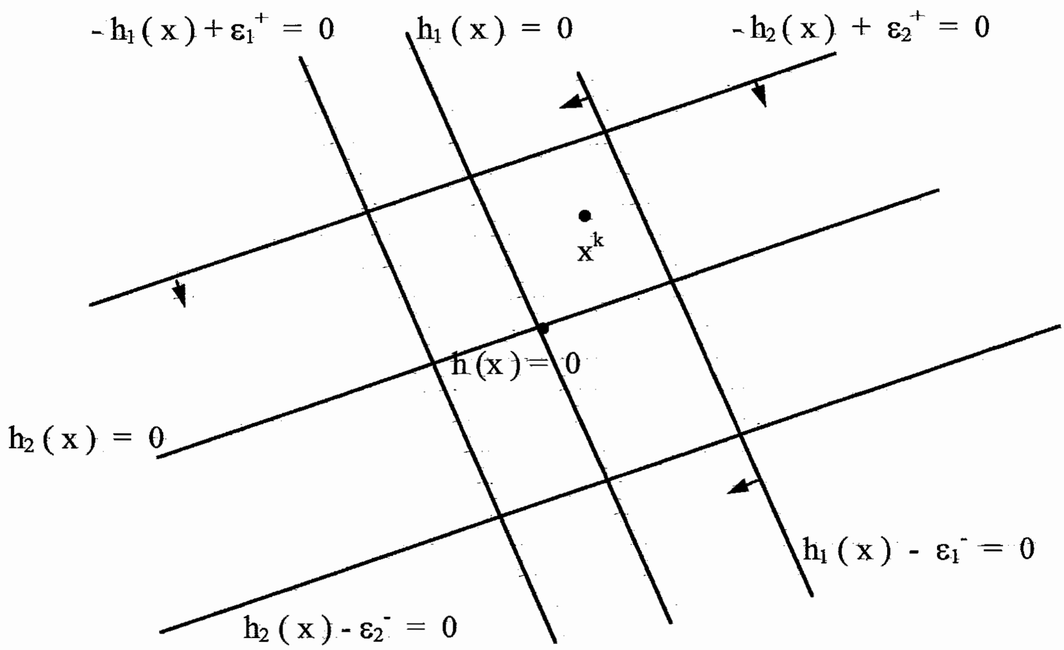

Figura 7 : Redução do intervalo entre as folgas

A figura 7 dá uma idéia de como a região viável relaxada vai gradativamente se aproximando do espaço viável (variedade de dimensão (n - p) , definido por $S = \{ x \in R^n \ / \ h(x) = 0 \ ; j = 1, \ldots, p \}$, quando as folgas são diminuidas provocando , dessa maneira os deslocamentos indicados pelas setas.

Outra componente fundamental do algorítmo é o processo de aumento do ângulo α (quando necessário) com o objetivo de obter pontos de ótimo intermediários x^k viáveis .

A última componente do algorítmo , também fundamental , é a diminuição do parâmetro τ , com o objetivo de aproximar o problema penalizado gradativamente do problema original .

CONDIÇÕES DO PROBLEMA

A fim provarmos a convergência do Algorítmo I, primeiramente é necessário que estabeleçamos algumas condições sobre o problema a ser resolvido .

(C1) : As funções $f(.)$ e $h_j(.)$; $j = 1, \ldots, m$ são duas vezes continuamente diferenciáveis

(C2) : Os gradientes $\nabla h_j(x^*)$; $j = 1, \ldots, m$ são linearmente independentes, dessa forma existe um único multiplicador de Lagrange dado por $\lambda^* = (\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*)$ tal que :

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

(C3) : A matriz Hessiana da função Lagrangeana $l(x, \lambda)$ dada por

$$\nabla^2 l(x^*, \lambda^*) = \nabla^2 f(x^*) + \sum_{j=1}^{m} \lambda_j^* . \nabla^2 h_j(x^*)$$

é positiva definida sobre o plano tangente correspondente às restrições, isto é :

$$z^t . \nabla^2 l(x^*, \lambda^*) . z > 0$$

para $\forall \; z \in R^n$, $z \neq 0$ tal que $z^t . \nabla h_j(x^*) = 0$, $j = 1, \ldots, p$

(C4) : Existe um $\varepsilon > 0$ tal que o conjunto definido por :

$$S_\varepsilon = \{ x \; ; \; h_j(x) \leq \varepsilon_j^+ \; , \; h_j(x) \geq \varepsilon_j^- \; ; \; j = 1, \ldots, p \} \quad \text{é limitado} .$$

(C5) : Não-degenerescência ou complementaridade estrita , isto é, $\lambda_j^* \neq 0$, $j = 1, \ldots, p$.

(C6) : O problema penalizado (3.6) tem solução x^k .

(C7) : O problema original (3.1) tem solução ótima x^* .

Vamos também apresentar alguns resultados que serão utilizados nos nossos desenvolvimentos teóricos de forma a obter a convergência do algorítmo .

RESULTADO 1 : (Mínimo viável)

Se as condições (C1) , ... , (C6) forem obedecidas , existe $\bar{\alpha}(\tau^o)$ tal que , para todo α no intervalo $\bar{\alpha} \leq \alpha < \pi/2$ e para todo τ no intervalo $0 \leq \tau \leq \tau^o$, qualquer ponto de mínimo $x(\alpha, \tau)$ da função objetivo modificada $F(x, \alpha, \tau)$ é viável .

Demonstração : Ver Xavier [62]

RESULTADO 2 : (Teorema da função implícita) , Ver Avriel [4]

Suponha que as funções φ_j a valores reais definidas sobre um conjunto D , e continuamente diferenciáveis num conjunto aberto $D_1 \subset D \subset R^{m+p}$, onde $p > 0$ e $\varphi_j(x^o, y^o) = 0$ para $j = 1, \ldots, m$ e $(x^o, y^o) \in D_1$. Suponha que a matriz Jacobiana $\dfrac{\partial . \varphi_i(x^0, y^0)}{\partial . x_j}$ tenha posto m . Então, existe uma vizinhança $\Omega_\delta(x^o,y^o) \subset D_1$, um conjunto aberto $D_2 \subset R^p$ contendo y^o , e funções ϕ_k ; $k = 1, \ldots, m$ a valores reais continuamente diferenciáveis sobre D_2 , tais que as seguintes condições são satisfeitas :

$$x_k^o = \phi_k(y^o) \quad ; \quad k = 1, \ldots, m \qquad (3.11)$$

Para todo $y \in D_2$, temos :

$$\varphi_j(\phi(y), y) = 0 \quad , \quad j = 1, \ldots, m \qquad (3.12)$$

onde $\phi(y) = (\phi_1(y), \ldots, \phi_m(y))$, e para todo $(x, y) \in \Omega_\delta(x^o, y^o)$ a matriz Jacobiana $\dfrac{\partial.\varphi_i(x,y)}{\partial.x_j}$ tem posto **m** . Além do mais, para $y \in D_2$ as derivadas parciais de $\phi_k(y)$ são as soluções do conjunto das equações lineares.

$$\sum_{k=1}^{m} \frac{\partial.\varphi_i.(\phi(y),y)}{\partial.x_k} \cdot \frac{\partial.\phi_k(y)}{\partial.y_j} = - \frac{\partial.\varphi_i(\phi(y),y)}{\partial.y_j} \quad , \quad j = 1, \ldots, m \tag{3.13}$$

RESULTADO 3 :

Seja x* um ponto de mínimo local para a problema (3.1), este ponto x* seja um ponto regular e ademais o par (x^*, λ^*) satisfaça à condição (C3) . Então , a matriz

$$\mathbf{J} = \begin{bmatrix} \nabla^2 L(x^*,\lambda^*) & \nabla h(x^*) \\ \nabla h(x^*)^t & 0 \end{bmatrix} \text{ é não-singular .} \tag{3.14}$$

Demonstração :

Suponhamos que a matriz **J** seja singular , então existem vetores $y \in R^n$, $z \in R^m$, ambos não-nulos tais que , o par (y , z) esteja no espaço nulo de **J** , ou equivalentemente:

$$\begin{bmatrix} \nabla^2 L(x^*,\lambda^*) & \nabla h(x^*) \\ \nabla h(x^*)^t & 0 \end{bmatrix} \cdot \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.15}$$

ou

$$\nabla^2_x L(x^*, \lambda^*).y + \nabla h(x^*).z = 0 \tag{3.16}$$

$$\nabla h(x^*)^t.y = 0 \tag{3.17}$$

Multiplicando a expressão (3.16) por y^t , e usando (3.17),necessariamente devemos ter :

$$y^t \nabla^2_x L(x^*,\lambda^*).y = 0$$

Agora, usando a condição (C3) , que trata da suficiência de segunda ordem , podemos concluir que $y = 0$. Assim , o sistema acima se reduz a $\nabla h(x^*).z = 0$. Finalmente , usando a condição (C2) , devemos ter $z = 0$. Isto contradiz o fato de y e z não poderem ser ambos iguais a zero, portanto , a matriz **J** é não-singular.

□

RESULTADO 4 :

Tomando as hipóteses do Resultado 3 verdadeiras , então existe um escalar $\delta > 0$ e funções continuamente diferenciáveis $x(\,.\,) : \Omega(0 , \delta) \rightarrow R^n$ e $\lambda(\,.\,) : \Omega(0, \delta) \rightarrow R^m$ tais que $x(0) = x^*$, $\lambda (0) = \lambda^*$, e para todo $\varepsilon \in \Omega(0, \delta)$ o par $\{ x(\varepsilon) , \lambda (\varepsilon) \}$ são mínimo local e multiplicador de Lagrange para o problema :

$$\text{minimizar} \quad f(x) \qquad (3.18)$$
$$\text{sujeito à :} \quad h(x) = \varepsilon$$

Além do mais , $\nabla_\varepsilon f[x(\varepsilon)] = - \lambda (\varepsilon)$, $\forall\ \varepsilon \in \Omega(0 , \delta)$. (3.19)

Este teorema é um resultado da aplicação direta do teorema da função implícita. Abaixo reproduzimos a demonstração dada em Bertsekas [12].

Demonstração :

Considere o sistema de equações em $(x, \lambda, \varepsilon)$ dado por :

$$\nabla f(x) + \nabla h(x)^t . \lambda = 0 \qquad (3.20)$$

$$h(x) - \varepsilon = 0 \qquad (3.21)$$

que tem $(x^*, \lambda^*, 0)$ como solução . Além do mais , o Jacobiano deste sistema com respeito à (x, λ) nesta solução é a matriz invertível **J** do Resultado 3 . Consequentemente pelo Resultado 2 , teorema da função implícita, existe um $\delta > 0$ e funções $x(.) \in C^1$, $\lambda(.) \in C^1$ sobre $\Omega(0, \delta)$ tais que :

$$\nabla f[x(\varepsilon)] + \nabla h[x(\varepsilon)].\lambda(\varepsilon) = 0 \qquad (3.22)$$

$$h[x(\varepsilon)] = \varepsilon \quad , \quad \forall \; \varepsilon \in \Omega(0, \delta)$$

Para ε suficientemente pequeno , os vetores $x(\varepsilon)$ e $\lambda(\varepsilon)$ satisfazem as condições de suficiência para o problema (3.18) , em vista do fato de que eles o satisfazem, por hipótese para $\varepsilon = 0$. Consequentemente , δ pode ser escolhido de modo que o par $\{ x(\varepsilon), \lambda(\varepsilon) \}$ seja um mínimo local e o correspondente multiplicador de Lagrange para o problema (3.18) .

Agora, do sistema (3.22) temos :

$$\nabla_\varepsilon x(\varepsilon).\nabla f[x(\varepsilon)] + \nabla_\varepsilon x(\varepsilon).\nabla h[x(\varepsilon)].\lambda(\varepsilon) = 0$$

ou

$$\nabla_\varepsilon f[x(\varepsilon)] = - \nabla_\varepsilon x(\varepsilon).\nabla h[x(\varepsilon)].\lambda(\varepsilon) \qquad (3.23)$$

Por outro lado , diferenciando a relação $\varepsilon = h[x(\varepsilon)]$, em relação a ε obtemos :

$$I = \nabla_\varepsilon h[x(\varepsilon)] = \nabla_\varepsilon x(\varepsilon).\nabla h[x(\varepsilon)] \qquad (3.24)$$

Combinando (3.23) e (3.24) , temos :

$$\nabla_\varepsilon f[x(\varepsilon)] = -\lambda(\varepsilon) . \qquad \square$$

FUNCIONAMENTO E CONVERGÊNCIA DO ALGORÍTMO

Estabelecidos os resultados iniciais , vamos agora nos deter à análise do funcionamento do algorítmo até a sua convergência .

A cada iteração k , no passo 3 , é feita a minimização irrestrita do problema (3.6), obtendo-se um ponto ótimo intermediário x^k , ou seja :

$$x^k \in \operatorname{argmin}\{ F(x, \alpha^k, \tau^k) \}, \qquad (3.25)$$

Dessa forma , o ponto x^k satisfaz ao sistema :

$$\nabla F(x, \alpha^k, \tau^k) = 0 . \qquad (3.26)$$

No passo 4 testa-se a viabilidade do ponto x^k . Se o ponto for inviável é executado o passo 5 do algorítmo , onde há um acréscimo monótono do parâmentro α da função hiperbólica .

Em vista do Resultado 1 , Xavier [62] , haverá um certo valor $\bar{\alpha}(\varepsilon)$ que certamente o ponto de mínimo x^k da função objetivo modificada $F(x, \alpha^k, \tau^k)$ é viável para o problema relaxado (3.5) . Assim , o algorítmo sempre trabalha com pontos x^k viáveis , isto vale dizer que :

$$h_j(x^k) = \theta_j \quad ; \quad \varepsilon_j^- \leq \theta_j \leq \varepsilon_j^+ \qquad (3.27)$$

Devemos ainda considerar que a partir do momento em que $\alpha^k > \bar{\alpha}(\varepsilon)$, o algorítmo executará o passo 6 , em que há uma redução do parâmentro τ de penalização .

O passo 7 é um passo intermediário destinado unicamente a calcular os valores das restrições decorrentes das folgas (distâncias à esquerda d_j^- e à direita d_j^+).

Vamos agora analizar o passo 8 , que é o mais crítico do algorítmo no que concerne a sua convergência .

Nesse ponto vamos por facilidade de apresentação , desenvolvimento e sem qualquer perda de generalidade supor que os multiplicadores de Lagrange do problema original (3.1) , sejam todos positivos , isto é , $\lambda_j > 0$, $j = 1, \ldots, p$.

Analisando a construção do algorítmo , para cada restrição $j = 1, \ldots, p$ podemos particionar a seqüência de iterações no passo 8 segundo as três alternativas mutuamente exclusivas abaixo apresentadas :

Alternativa 1 :

A partir de uma dada iteração , sempre o teste do passo 8A é satisfeito , ou seja , sempre são executadas as instruções 8A.1 e 8A.2 .

Alternativa 2 :

A partir de uma dada iteração , sempre o teste do passo 8B é satisfeito , ou seja , sempre são executadas as instruções 8B.1 e 8B.2 .

Alternativa 3 :

O passo 8C é executado infinitas vezes , não necessariamente consecutivas.

Analisaremos , abaixo , cada uma dessas sequências alternativas :

Discussão inicial da alternativa 3 :

Nessa situação , pela construção do algorítmo , as instruções (8C.1) e (8C.2) são executadas infinitas vezes, (não necessariamente consecutivas). Como $\tau^k \to 0$, conforme o passo 6 , necessariamente devemos ter :

$$\begin{cases} \varepsilon_j^- \to 0 \\ \varepsilon_j^+ \to 0 \end{cases} \tag{3.28}$$

Por (3.28) e pela viabilidade $\varepsilon_j^- \leq h_j(x^k) \leq \varepsilon_j^+$, dada no passo 4 , podemos concluir que :

$$h_j(x^k) \to 0 \tag{3.28a}$$

Discussão inicial da alternativa 2 :

Nessa situação , certamente existe uma iteração K_3 tal que para toda iteração $k > K_3$, o teste do passo 8B é sempre satisfeito .

Nesse caso , pela viabilidade do ponto x^k temos :

$$h_j(x^k) \leq \varepsilon_j^+$$

e pela satisfação do teste 8B temos :

$$h_j(x^k) > -2\beta.\tau^k .$$

Considerando as duas desigualdades imediatamente acima temos :

$$-2\beta.\tau^k < h_j(x^k) \leq \varepsilon_j^+ \qquad (3.29)$$

Como $\varepsilon_j^+ \to 0$, pela instrução (8B.1) e $\tau^k \to 0$, conforme o passo 6 , necessariamente devemos ter :

$$h_j(x^k) \to 0 , \qquad (3.30)$$

Discussão inicial da alternativa 1 :

Nessa última situação , suponhamos que para toda iteração $k > K_4$, sempre o teste do passo 8A é satisfeito .

Nesse caso , pela viabilidade do ponto x^k temos :

$$h_j(x^k) \geq \varepsilon_j^- ,$$

e pela satisfação do teste 8A temos :

$$h_j(x^k) < 2\beta.\tau^k$$

Considerando as duas desigualdades imediatamente acima , temos :

$$\varepsilon_j^- \leq h_j(x^k) < 2\beta.\tau^k \qquad (3.31)$$

Como $\varepsilon_j^- \to 0$, pela instrução (8A.1) e $\tau^k \to 0$, conforme o passo 6 , necessariamente devemos ter :

$$h_j(x^k) \to 0 \ , \qquad (3.32)$$

Síntese das alternativas 1 , 2 e 3

Considerando a validade das expressões (3.28a) ou (3.30) ou (3.32) , podemos concluir que :

$$h_j(x^k) \to 0 \quad ; \quad j = 1 , \ldots , p \qquad (3.33)$$

ou seja , a seqüência de pontos ótimos intermediários se aproximam da região viável do problema original .

Pela condição C4 , o conjunto S_ε é compacto , assim a seqüência $\{ x^k \}$ é limitada e dessa forma possui uma subseqüência convergente . Tomando-se sem perda de generalidade a subseqüência como a própria seqüência , temos

$$\lim_{k\to\infty} x^k = \bar{x} \ .$$

Usando o fato das funções $h_j(\,.\,)$ serem contínuas , (condição C1), temos :

$$h(\lim_{k\to\infty} x^k) = h(\bar{x}) \quad \Rightarrow \quad \lim_{k\to\infty} h_j(x^k) = h_j(\bar{x})$$

Por outro lado , em virtude da expressão (3.33) , temos :

$$\lim_{k\to\infty} h_j(x^k) = 0 \quad ; \quad j = 1 , \ldots , p$$

implicando que $h_j(\bar{x}) = 0$, $j = 1 , \ldots , p$, ou seja , que o ponto $\bar{x}$ é viável para o problema original (3.1).

Para provar que $\bar{x}$ é solução ótima do problema original (3.1), devemos ter :

$$f(x^*) = f(\bar{x}).$$

Vejamos :

Como x^k é solução ótima do problema relaxado (3.5) em qualquer iteração k , temos :

$$F(x^*, \alpha^k, \tau^k) \geq F(x^k, \alpha^k, \tau^k)$$

Tomando o limite temos :

$$\lim_{k\to\infty} F(x^*, \alpha^k, \tau^k) \geq \lim_{k\to\infty} F(x^k, \alpha^k, \tau^k)$$

Considerando que $\bar{x}$ e x^* são viáveis e o fato de que o algorítmo gera uma seqüência $\{\tau^k\}$ tal que $0 < \tau^{k+1} \leq \tau^k$, $\lim_{k\to\infty} \tau^k = 0$ e usando as propriedades da função penalidade hiperbólica em [63] , temos :

$$F(x^*, \bar{\alpha}, 0) \geq F(\bar{x}, \bar{\alpha}, 0) \qquad \Rightarrow$$

$$f(x^*) \geq f(\bar{x}) \qquad \textbf{(A)}$$

Por outro lado , considerando que x^* é solução ótima do problema original (3.1) e , como $\bar{x}$ é viável para este mesmo problema , temos :

$$f(x^*) \leq f(\bar{x}) \qquad \textbf{(B)}$$

Por **(A)** e **(B)** , podemos concluir que :

$$f(x^*) = f(\bar{x}). \qquad \square$$

Pelo exposto acima , foi demonstrado o seguinte teorema :

TEOREMA DA CONVERGÊNCIA PRIMAL

Sendo x^k, como definido anteriormente, um ponto tal que :

$$F(x^k, \alpha^k, \tau^k) = min_x\, F(x, \alpha^k, \tau^k),$$

então existe uma subseqüência convergente $x^k \rightarrow \bar{x}$, e o limite $\bar{x}$ é um ponto ótimo para o problema original (3.1) .

Provado o teorema da convergência , vamos nos aprofundar nas análises das alternativas consideradas anteriormente .

Será visto abaixo que , somente a alternativa 1 é válida . Esse resultado tem o importante significado : A região viável do problema relaxado sempre tem um amplo interior .

Analisaremos , outrossim , o comportamento da seqüência dos multiplicadores de Lagrange equivalentes λ_j^k , gerada pelo algorítmo .

Análise da viabilidade da alternativa 3 :

Como x^k é um ponto ótimo para o problema relaxado , ele satisfaz o sistema de equações :

$$\nabla f(x^k) + \sum_{j=1}^{m} \lambda_j^k \nabla h_j (x^k) = 0$$

No ponto ótimo do problema original , as condições de KKT de primeira ordem nos fornece :

$$\nabla f(\bar{x}) + \sum_{j=1}^{m} \overline{\lambda_j} \nabla h_j (\bar{x}) = 0$$

De outro lado, considerando $x^k \to \bar{x}$, ponto ótimo do problema original, visto no teorema de convergência, usando a continuidade das funções $\nabla f(\,.\,)$ e $\nabla h_j(\,.\,)$, $j = 1, \dots, p$ dada pela condição C1 e a independência linear dos gradientes $\nabla h_j(\bar{x})$, $j = 1, \dots, p$ dada pela condição C2, podemos concluir, conforme estabelecido pelo Resultado 4, que existe uma vizinhança $\Omega_1(x^*, \delta_1)$ em que os multiplicadores equivalentes λ_j^k, $j = 1, \dots, p$, correspondentes aos pontos x^k serão positivos, pois terão o mesmo sinal dos multiplicadores ótimos, por hipótese, supomos serem todos positivos.

No processo iterativo do algorítmo, se numa dada iteração o passo 8C for executado, então na iteração imediatamente seguinte, certamente o teste 8A será satisfeito, (ou seja, as instruções 8A.1 e 8A.2 serão executadas) pelas razões a seguir apresentadas.

Considerando que os multiplicadores equivalentes são positivos, visto logo acima, e usando a proposição 2, temos :

$$-h_j(x^k) + \varepsilon_j^+ > h_j(x^k) - \varepsilon_j^- \qquad (3.34)$$

Dado que na iteração anterior as instruções 8C.1 e 8C.2 foram executadas, temos :

$$-\varepsilon_j^- = +\varepsilon_j^+ , \qquad (3.35)$$

De (3.34) e (3.35) podemos concluir que $h_j(x^k) < 0$. Assim, a segunda parte do teste 8A [$h_j(x^k) < 2\beta\,\tau^k$] será também satisfeita pois $\beta > 0$ e $\tau > 0$.

Vale dizer, o lado correto da região viável será acionado na iteração imediatamente seguinte à execução de um passo 8C.

Vamos mostrar a seguir, que não somente nesta iteração imediatamente seguinte, o teste 8A será satisfeito, como também em todas as demais iterações seguintes.

De fato, como $\lambda_j^* > 0$, $j = 1, \dots, p$, existirá uma iteração K_1 tal que:

$$\lambda_j^k > \lambda_j^*/2. \quad , \quad j = 1, \dots, p \tag{3.36}$$

para todo $k > K_1$.

Como definimos em (3.10) $\lambda_j^{\,k} = \mu_j^k - \eta_j^k$, temos :

$$\mu_j^{\,k} - \eta_j^k > \lambda_j^*/2 \tag{3.37}$$

Como $\mu_j^k > 0$ e $\eta_j^k > 0$, então será observada a relação :

$$\mu_j^k > \lambda_j^*/2 \tag{3.38}$$

De outro lado, se o passo 8C fosse executado infinitas vezes, teríamos por força da instrução 8C.3 que $\alpha \rightarrow \pi/2$. Ademais, para que o passo 8C seja executado é necessário que o teste de 8A seja falso, ou seja, pelo menos uma das relações

$$d_j^+ > d_j^- \tag{3.39}$$

ou

$$h_j(x^k) < 2\beta.\tau^k \tag{3.40}$$

não seja verdadeira.

Como $\lambda_j^k > 0$, face a Proposição 2, o primeiro teste ($d_j^- > d_j^+$) não falhará.

Se por acaso o segundo teste [$h_j(x^k) < 2\beta.\tau^k$] falhar, ou seja, se tivermos [$h_j(x^k) > 2\beta.\tau^k$], também será verdadeira a relação :

$$h_j(x^k) - \varepsilon_j > 2\beta.\tau^k \quad , \quad \varepsilon_j^- \leq 0. \tag{3.41}$$

Relembrando que μ_j^k definido em (3.9a) corresponde ao negativo da derivada da função penalidade hiperbólica e considerando o fato de que $\alpha \to \pi/2$, visto logo acima, e fazendo uso da Propriedade 1 da penalização hiperbólica , (capítulo 2), podemos concluir que se (3.41) for válida , necessariamente o valor de μ_j^k vai para zero . A figura 8 tem o objetivo de tornar mais clara essa conclusão .

Assim , haverá um valor $\bar{\bar{\alpha}}$ tal que $\mu_j^k < \lambda_j^*/2$, violando a expressão (3.38). Dessa forma podemos concluir que o segundo teste não falhará , ou seja , após $\alpha > \bar{\bar{\alpha}}$ o algorítmo certamente executará o passo 8A .

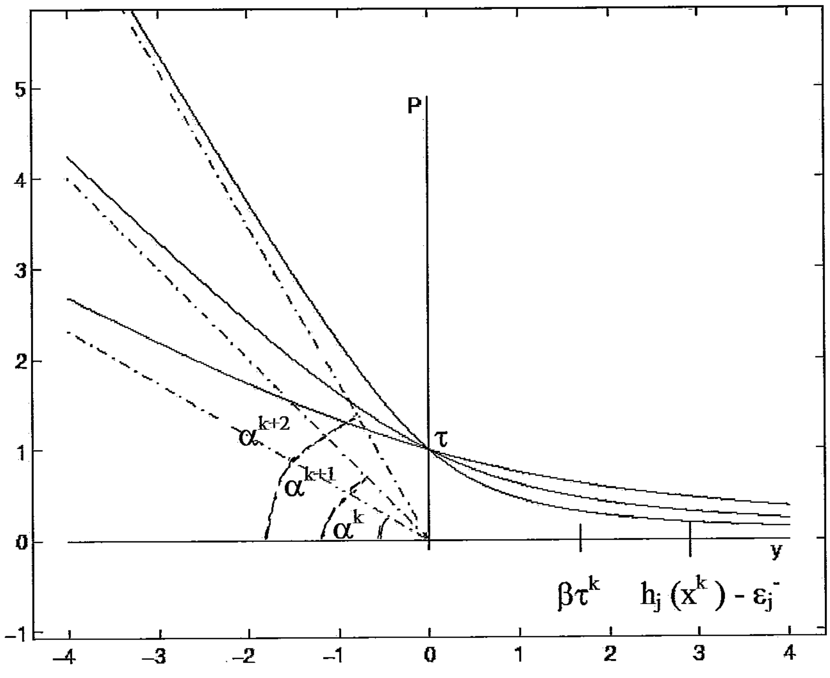

Figura 8 : Acréscimo da inclinação da função penalidade

Análise da viabilidade da alternativa 2 :

Observa-se de outro lado que pela primeira parte do teste 8B , temos :

$$h_j(x^k) - \varepsilon_j^- > - h_j(x^k) + \varepsilon_j^+ \qquad (3.42)$$

Face à desigualdade (3.42) , pela proposição 2 , podemos concluir que os multiplicadores de Lagrange equivalentes λ_j^k são negativos . Dessa forma o passo 8B não é aceito , pois contraria a hipótese dos multiplicadres equivalentes serem positivos .

Análise da viabilidade da alternativa 1 :

Por exclusão , esta é a única alternativa viável e tem uma grande implicação prática : a região viável na seqüência de problemas relaxados mantém-se sempre com um amplo interior , como afirmamos anteriormente.

Se em caso contrário , a alternativa 3 fosse observada , certamente teríamos um fechamento monótono da região viável . Simultaneamente ocorreria o consequente crescimento do mau condicionamento da função objetivo do problema relaxado . Em síntese , tal ocorrência inviabilizaria em termos práticos a validade da presente proposta metodológica .

Vamos agora analisar o comportamento da seqüência dos multiplicadores de Lagrange equivalentes gerados pelo algorítmo .

Considerando a função objetivo penalizada do problema (3.6) , temos num ponto ótimo intermediário x^k observada a seguinte condição :

$$\nabla_x F(x^k, \alpha^k, \tau^k) = 0$$

Tomando -se o limite na expressão acima , temos :

$$\lim_{k\to\infty} \left[\nabla f(x^k) - \sum_{j=1}^{m} \lambda_j^k \nabla h_j(x^k) \right] = 0$$

Considerando que $\lim_{k\to\infty} x^k = x^*$ e fazendo uso da condição (C1) , temos :

$$\lim_{k\to\infty} \left[\nabla f(x^*) - \sum_{j=1}^{m} \lambda_j^* \nabla h_j(x^*) \right] = 0$$

Como os gradientes $\nabla h_j(x^*)$ são L.I. (condição (C2) , logo esse sistema tem solução única , donde podemos concluir que :

$$\lim_{k\to\infty} \lambda_j^k = \lambda_j^* . \qquad \square$$

Pelo acima exposto , demonstramos o seguinte teorema :

TEOREMA DE CONVERGÊNCIA DUAL

A seqüência dos multiplicadores equivalentes λ_j^k gerada pelo algorítmo I converge para o multiplicador ótimo do problema original λ_j^*.

Esse resultado tem de outro lado um outro importante significado prático : o passo 5 do algorítmo é executado um número finito de vezes.

Supondo verdadeira a hipótese contrária , implicaria em $\alpha \to \pi / 2$ concomitantemente a ocorrência cíclica de pontos de mínimos intermediários x^k , no passo 3 do algorítmo, inviáveis , ou seja , pontos em que $h_j(x^k) - \varepsilon_j^- < 0$, para algum índice j .

Pela definição (3.10) , $\lambda_j^k = \mu_j^k - \eta_j^k$, e nesta hipótese pela característica da penalização adotada , teríamos $\mu_j^k \to 0$ e pôr isso $\lambda_j^k \to \infty$, o que é um absurdo.

RESULTADOS COMPUTACIONAIS DOS PROBLEMAS RESTRITOS A IGUALDADES

Esta seção tem como objetivo a apresentação dos resultados computacionais obtidos com a implementação do método de penalização hiperbólica \desenvolvido para solucionar os problemas de programação não-linear, sujeito somente a restrições de igualdades.

O algorítmo I utilizado para resolver esses problemas mencionados acima, foram implementados em FORTRAN e os valores de tempo de CPU foram obtidos usando um Intel 486 de 33MHz.

A minimização irrestrita foi feita utilizando-se uma rotina da Harwell Library que basicamente implementa o método Quasi-Newton com atualização BFGS, (Broyden - Fletcher - Goldfarb - Shanno).

ESCOLHA DOS PARÂMETROS INICIAIS

Os parâmetros do algorítmo I foram escolhidos conforme abaixo explicado:

1 - ESCOLHA DO PARÂMETRO α^o:

Basicamente deve ser um valor tal que a minimização irrestrita da função objetivo penalizada tenha ponto de mínimo, ou seja,

$$\inf_x F(x, \alpha^o, \tau^o) = F^o > -\infty$$

Em particular , pequenos valores de α^o podem produzir divergência da minimização irrestrita . Nesse sentido , uma necessária salvaguarda é tomada no início do algorítmo , a fim de evitar essa dificuldade.

2 - ESCOLHA DAS FOLGAS ε^+ , ε^- :

Em todos os casos vinculamos esses parâmetros ao parâmetro de penalidade τ^o, rigorosamente , como especificamos no algorítmo I .

3 - ESCOLHA DO PARÂMETRO τ^o :

Adotamos em todos esses experimentos computacionais o valor $\tau^o = 0.01$. Alternativamente poderíamos adotar a escolha proposta por Fiacco & McCormick [25]

$$P(x^o, \alpha^o, \tau^o) / F(x^o, \alpha^o, \tau^o) = 0.1$$

Devemos enfatizar que para todos os 12 problemas testados , o algorítmo I obteve a solução ótima especificada na referência do Hock & Schittkowiski [34].

Para ilustrar o mecanismo de funcionamento do algorítmo I , primeiramente são apresentados nas Tabelas 1 , 2 e 3 os resultados computacionais, obtidos na resolução de um problema teste muito simples .

EXEMPLO 1 :

$$\text{minimizar} \quad f(x) = x_1^2 + x_2^2$$

$$\text{sujeito à :} \quad h(x) = 2x_1 + x_2 - 1 = 0$$

e tem como solução ótima o ponto $x^* = (0.4, 0.2)$ e $\lambda^* = 0.4$

Adotamos os seguintes parâmetros iniciais no passo 1 do algorítmo I.

$x^o = (2, -1)$, $\tau^o = 0.1000000\,E-01$ e $\alpha^o = 0.114576\,E+01$.

DESCRIÇÃO DAS TABELAS

As colunas da Tabela 1 representam , na ordem , o número da minimização irrestrita **k** , o valor do parâmetro ângulo da penalização hiperbólica α^k , o valor do parâmetro distância da penalização hiperbólica τ^k , as componentes do ponto solução de cada minimização irrestrita $\mathbf{x}_i^k$; $i = 1 , 2$, e finalmente o número de **Passos** de cada minimização irrestrita .

As colunas da Tabela 2 representam na ordem , o número da minimização irrestrita **k** ,o valor das folgas à esquerda ε_j^- e à direita ε_j^+ , o valor da restrição de igualdade e os multiplicadores de Lagrange λ_i^k associados às restrições de desigualdades .

Finalmente, na Tabela 3, temos , o número da minimização irrestrita **k**, os valores da função objetivo modificada **F (.)** do problema irrestrito , da função penalidade **P (.)** e da função objetivo **f (.)** do problema original .

Após a descrição apresentada acima , podemos observar nas Tabelas 1 e 2 , que os pontos $\mathbf{x}^k$ encontrados na minimização irrestrita até a iteração $k = 7$, são inviáveis em relação ao problema relaxado (3.5) , porque os valores de $h (x^k)$ estão situados fora do intervalo definido pelas folgas da relaxação .

Em função dessa inviabilidade , o algorítmo executa o passo 5 , em que é efetuado o aumento do parâmetro ângulo α , deixando o parâmetro distância τ constante , conforme pode ser visto na segunda e terceira colunas da Tabela 1 .

A partir da iteração $k = 8$, as minimizações irrestritas produzem pontos x^k viáveis para o problema relaxado (3.5) , pois os valores de $h(x^k)$ estão dentro do intervalo definido pelas folgas (Veja Tabela 2) .

Com relação aos parâmetros da função penalidade , podemos ver que o ângulo α permanece constante , e a distância τ decresce até encontrar o ponto ótimo x^* , no limite , (veja Tabela 1) .

Na Tabela 3 , podemos observar a convergência dos valores da função objetivo modificada ao valor da função objetivo do problema original do problema original .

Podemos , sobretudo , observar nas tabelas 1 e 2 , as convergências das seqüências $\{ x^k \}$ e $\{ \lambda^k \}$ aos valores otimais especificados acima na descrição do exemplo 1 .

TABELA 1 : Resultados Computacionais do Problema Teste 1 . Parte 1 / 3

Iteração k	α^k	τ^k	x_1^k	x_2^k	PASSOS
1	0.114576 E+01	0.10 E - 01	0.870692 E - 02	0.435346 E - 02	8
2	0.114576 E+01	0.10 E - 02	0.199184 E - 01	0.995920 E - 02	4
3	0.229061 E+01	0.10 E - 02	0.399541 E - 01	0.199770 E - 01	6
4	0.457392 E+01	0.10 E - 02	0.799706 E - 01	0.399853 E - 01	4
5	0.909027 E+01	0.10 E - 02	0.159972 E+ 00	0.799862 E - 01	5
6	0.177446 E+02	0.10 E - 02	0.319691 E+ 00	0.159845 E+ 00	9
7	0.326192 E+02	0.10 E - 02	0.359843 E+ 00	0.179921 E+ 00	12
8	0.520012 E+02	0.10 E - 02	0.360303 E+ 00	0.180151 E+ 00	15
9	0.520012 E+02	0.10 E - 03	0.396025 E+ 00	0.198012 E+ 00	21
10	0.520012 E+02	0.10 E - 04	0.399602 E+ 00	0.199801 E+ 00	25
11	0.520012 E+02	0.10 E - 05	0.399960 E+ 00	0.199980 E+ 00	24
12	0.520012 E+02	0.10 E - 06	0.399996 E+ 00	0.199998 E+ 00	30
13	0.520012 E+02	0.10 E - 07	0.399999 E+ 00	0.199999 E+ 00	27
14	0.520012 E+02	0.10 E - 08	0.400000 E+ 00	0.200000 E+ 00	29

TABELA 2 : Resultados Computacionais do Problema Teste 1 . Parte 2 / 3

Iteração k	ε_k^-	$h(x^k)$	ε_k^+	λ_1^k
1	- 0.10000 E+01	- 0.978232 E+00	0.10000 E+01	0.97823E - 02
2	- 0.10000 E+00	- 0.950204 E+00	0.10000 E+01	0.19931E - 01
3	- 0.10000 E+00	- 0.900114 E+00	0.10000 E+01	0.39961E -01
4	- 0.10000 E+00	- 0.800073 E+00	0.10000 E+01	0.79974E - 01
5	- 0.10000 E+00	- 0.600068 E+00	0.10000 E+01	0.15997E +00
6	- 0.10000 E+00	- 0.200772 E+00	0.10000 E+01	0.31969E +00
7	- 0.10000 E+00	- 0.100392 E+00	0.10000 E+01	0.35984E +00
8	- 0.10000 E+00	- 0.99240 E -01	0.10000 E+01	0.36030E +00
9	- 0.10000 E -01	- 0.99355 E -02	0.10000 E+01	0.39602E +00
10	- 0.10000 E -02	- 0.99366 E -03	0.10000 E+01	0.39960E +00
11	- 0.10000 E -03	- 0.99367 E -04	0.10000 E+01	0.39996E +00
12	- 0.10000 E -04	- 0.99367 E -05	0.10000 E+01	0.39999E +00
13	- 0.10000 E -05	- 0.99367 E -06	0.10000 E+01	0.39999E +00
14	- 0.10000 E -06	- 0.99367 E -07	0.10000 E+01	0.40000E +00

TABELA 3 : Resultados Computacionais do Problema Teste 1 . Parte 3 / 3

Iteração k	$F(x^k, \alpha^k, \tau^k)$	$P(x^k, \alpha^k, \tau^k)$	$f(x^k)$
1	0.122633E - 01	0.121685E -01	0.947631E - 04
2	0.175842E - 01	0.170883E -01	0.495929E - 03
3	0.340443E - 01	0.320489E -01	0.199541E - 02
4	0.640247E - 01	0.560306E -01	0.799413E - 02
5	0.112016E +00	0.800273E -01	0.319890E - 01
6	0.160033E +00	0.321807E -01	0.127752E +00
7	0.162993E +00	0.113475E -02	0.161858E +00
8	0.162900E +00	0.626621E -03	0.162273E +00
9	0.196112E +00	0.667409E -04	0.196045E +00
10	0.199609E +00	0.673720E -05	0.199602E +00
11	0.199960E +00	0.674151E -06	0.199960E +00
12	0.199996E +00	0.674195E -07	0.199996E +00
13	0.199999E +00	0.674199E -08	0.199999E +00
14	0.200000E +00	0.674199E -09	0.200000E +00

Para uma melhor avaliação do desenvolvimento do algorítmo I , selecionamos alguns problemas testes do livro Hock e Schittkowski [34] seguindo a indicação recomendada pelo " Ad Hoc Committee to Revise Guidelines for Reporting Computational Experiments in Mathematical Programming " da Mathematical Programming Society, vide Jackson et all. [36] .

Os pontos iniciais x^o e , os valores ótimos x^* e λ^* , para todos os problemas selecionados são os mesmos apresentados no livro [34] .

Destaque-se que o algorítmo resolveu eficiente e eficazmente todos os problemas abaixo relacionados .

Problema 7

$$\text{minimizar} \quad f(x) = \ln(1 + x_1^2) - x_2$$

$$\text{sujeito à :} \quad h(x) = (1 + x_1^2)^2 + x_2^2 - 4 = 0$$

Ponto inicial: $x^o = (2 , 2)$

Solução ótima encontrada na 5ª iteração em 0.99 segundos de CPU .

$x^* = (0, 1.7320)$ e $\lambda^* = 0.2887$

Problema 27

$$\text{minimizar} \quad f(x) = 0.01 (x_1 - 1)^2 + (x_2 - x_1^2)^2$$

$$\text{sujeito à :} \quad h(x) = x_1 + x_3^2 + 1 = 0$$

Ponto inicial $x^o = (2 , 2 , 2)$

Solução ótima encontrada na 7ª itração em 1.32 segundos de CPU .

$x^* = (-1 , 1 , 0)$ e $\lambda^* = 0.4$

Problema 39

minimizar $f(x) = -x_1$

sujeito à : $h_1(x) = x_2 - x_1^3 - x_3^2 = 0$

$h_2(x) = x_1^2 - x_2 - x_4^2 = 0$

Ponto inicial $x^o = (2, 2, 2, 2)$

Solução ótima encontrada na 7ª iteração em 1.48 segundos de CPU.

$x^* = (1, 1, 0, 0)$, $\lambda_1{}^* = 1$ e $\lambda_2{}^* = 1$

Problema 42

minimizar $f(x) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$

sujeito à : $h_1(x) = x_1 - 2 = 0$

$h_2(x) = x_3^2 + x_4^2 - 2 = 0$

Ponto inicial $x^o = (1, 1, 1, 1)$

Solução ótima encontrada na 8ª iteração em 1.64 segundos de CPU.

$x^* = (2, 2, 0.6\sqrt{2}, 0.8\sqrt{2})$, $\lambda_1{}^* = 2.5355$ e $\lambda_2{}^* = 2$

Problema 61

minimizar $f(x) = 4x_1^2 + 2x_2^2 + 2x_3^2 - 33x_1 + 16x_2 - 24x_3$

sujeito à : $h_1(x) = 3x_1 - 2x_2^2 - 7 = 0$

$h_2(x) = 4x_1 - x_3^2 - 11 = 0$

Ponto inicial $x^o = (0, 0, 0)$

Solução ótima encontrada na 5ª iteração em 1.1 segundos de CPU.

$x^* = (5.32677015, -2.11899863, 3.21046423)$,

$\lambda_1{}^* = 1.7378$ e $\lambda_2{}^* = 0.8877$

Problema 77

minimizar $(x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$

sujeito à : $x_1^2.x_4 + \text{sen}(x_4 - x_5) - 2\sqrt{2} = 0$

$x_2 + x_3^4.x_4^2 - 8 - \sqrt{2} = 0$

Ponto inicial $x^o = (2, 2, 2, 2, 2)$

Solução ótima encontrada na 6ª iteração em 1.32 segundos de CPU .

$x^* = (1.166172, 1.182111, 1.380257, 1.506036, 0.610920)$

$\lambda_1^* = 0.08554$ e $\lambda_2^* = 0.03188$

Problema 78

minimizar $f(x) = x_1.x_2.x_3.x_4.x_5$

sujeito à : $h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$

$h_2(x) = x_2.x_3 - 5\,x_4.x_5 = 0$

$h_3(x) = x_1^3 + x_2^3 + 1 = 0$

Ponto inicial $x^o = (-2, 1.5, 2, -1, -1)$

Solução ótima encontrada na 6ª iteração em 1.54 segundos de CPU .

$x^* = (-1.717142, 1.595708, 1.827248, -0.763642, -0.763643)$ e

$\lambda_1^* = 0.7035$, $\lambda_2^* = 0.7444$ e $\lambda_3^* = 0.09681$

Problema do Bazaraa [6]

minimizar $f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$

sujeito à : $h(x) = x_1^2 - x_2 = 0$

Ponto inicial $x^o = (2, 1)$

Solução ótima encontrada na 6ª iteração em 1.1 segundos de CPU .

$x^* = (0.94558, 0.89344)$ e $\lambda_1^* = 3.3706$

PROBLEMAS DEGENERADOS

A teoria anteriormente desenvolvida considerava a hipótese de não-degenerescência dos multiplicadores de Lagrange, ou seja, $\lambda_j \neq 0$; $j = 1, \dots, p$.

A despeito da exigência teórica dessa condição, submetemos o Algorítmo I a problemas que não a atendem, ou seja, problemas na literatura denominados duais degenerados.

A seguir estão descritos alguns problemas degenerados do livro [34] resolvidos pelo método de penalização hiperbólica:

Problema Degenerado 50 :

$$\text{minimizar} \quad (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2$$

$$\text{sujeito à :} \quad x_1 + 2x_2 + 3x_3 - 6 = 0$$

$$x_2 + 2x_3 + 3x_4 - 6 = 0$$

$$x_3 + 2x_4 + 3x_5 - 6 = 0$$

Ponto inicial $x^o = (35, -31, 11, 5, -5)$

Solução ótima encontrada na 5ª iteração em 1.21 segundos de CPU.

$x^* = (1, 1, 1, 1, 1)$, $\lambda_1^* = 0$, $\lambda_2^* = 0$ e $\lambda_3^* = 0$

Problema Degenerado 28 :

$$\text{minimizar} \quad (x_1 + x_2)^2 + (x_2 + x_3)^2$$

$$\text{sujeito à :} \quad x_1 + 2x_2 + 3x_3 - 1 = 0$$

Ponto inicial $x^o = (-4, 1, 1)$

Solução ótima encontrada na 3ª iteração em 0.65 segundos de CPU.

$x^* = (-0.5, -0.5, 0.5)$ e $\lambda_1^* = 0$

Problema Degenerado 46 :

$$\text{minimizar} \quad (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$$

$$\text{sujeito à:} \quad x_1^2 x_4 + \text{sen}(x_4 - x_5) - 1 = 0$$

$$x_2 + x_3^4 . x_4^2 - 2 = 0$$

Ponto inicial $x^o = (0.7, 1.75, 0.5, 2, 2)$

Solução ótima encontrada na 10ª iteração em 1.1 segundos de CPU.

$x^* = (1, 1, 1, 1, 1)$, $\lambda_1^* = 0$ e $\lambda_2^* = 0$

Por concisão apresentaremos a seguir, unicamente os resultados computacionais obtidos para o problema degenerado 46.

Destacamos, entretanto, que esses resultados são similares aos outros casos.

TABELA 4 : Resultados Computacionais do Problema Teste 46. Parte 1 / 4

Iteração k	α^k	τ^k	x_1^k	x_2^k	x_3^k	x_4^k	x_5^k
1	0.871E+2	0.1E -1	0.99E+0	0.99E+0	0.10E+1	0.10E+1	0.99E+0
2	0.871E+2	0.1E -2	0.12E+1	0.12E+1	0.10E+1	0.10E+1	0.10E+1
3	0.871E+2	0.1E -3	0.10E+1	0.10E+1	0.10E+1	0.99E+0	0.10E+1
4	0.871E+2	0.1E -4	0.99E+0	0.99E+0	0.10E+1	0.10E+1	0.10E+1
5	0.871E+2	0.1E -5	0.99E+0	0.99E+0	0.10E+1	0.10E+1	0.99E+0
6	0.871E+2	0.1E -6	0.99E+0	0.99E+0	0.10E+1	0.99E+0	0.99E+0
7	0.871E+2	0.1E -7	0.99E+0	0.99E+0	0.10E+1	0.10E+1	0.99E+0
8	0.871E+2	0.1E -8	0.99E+0	0.99E+0	0.10E+1	0.10E+1	0.99E+0
9	0.871E+2	0.1E -9	0.10E+1	0.10E+1	0.10E+1	0.10E+1	0.10E+1
10	0.871E+2	0.1E -10	0.10E+1	0.10E+1	0.10E+1	0.10E+1	0.10E+1

TABELA 5 : Resultados Computacionais do Problema Teste 46 . Parte 2 / 4

Iteração k	ε_k^-	$h(x^k)$	ε_k^+	ε_k^-	$h(x^k)$	ε_k^+
1	- 0.1E+1	- 0.925E-9	0.1E+1	-0.1E+1	- 0.186E-9	0.1E+1
2	- 0.1E+0	0.543E+0	0.1E+1	-0.1E+0	0.258E+0	0.1E+1
3	- 0.1E+0	0.786E-10	0.1E+0	-0.1E-1	0.327E-10	0.1E-1
4	- 0.1E+0	- 0.252E-1	0.1E-1	-0.1E-1	- 0.455E-2	0.1E-2
5	- 0.1E-3	0.319E-12	0.1E-3	-0.1E-2	0.699E-10	0.1E-3
6	- 0.1E-3	- 0.450E-4	0.1E-4	-0.1E-2	- 0.449E-3	0.1E-4
7	- 0.1E-5	- 0.332E-15	0.1E-5	-0.1E-5	0.164E-13	0.1E-5
8	- 0.1E-6	- 0.450E-6	0.1E-5	-0.1E-5	- 0.450E-6	0.1E-6
9	- 0.1E-6	- 0.546E-12	0.1E-6	-0.1E-7	- 0.152E-12	0.1E-7
10	- 0.1E-6	- 0.449E-7	0.1E-7	-0.1E-8	- 0.449E-8	0.1E-8

TABELA 6 : Resultados Computacionais do Problema Teste . Parte 3 / 4

Iteração k	λ_1^k	λ_2^k
1	0.4999E -05	0.4999E -05
2	0.1207E -06	0.9082E -07
3	0.5000E -07	0.4999E -05
4	0.8959E -09	0.1622E -06
5	0.4999E -05	0.5000E -07
6	0.1652E -06	0.1652E -08
7	0.4999E -05	0.4999E -05
8	0.1652E -06	0.4999E -05
9	0.4999E -07	0.4999E -05
10	0.1652E -08	0.1645E -06

TABELA 7 : Resultados Computacionais do Problema Teste 46. Parte 4 / 4

Iteração k	$F(x^k, \alpha^k, \tau^k)$	$P(x^k, \alpha^k, \tau^k)$	$f(x^k)$	Passos
1	0.199E - 04	0.110E -18	0.199E - 04	92
2	0.395E - 06	0.882E -09	0.394E - 06	31
3	0.109E - 06	0.143E -19	0.109E - 06	54
4	0.203E - 08	0.902E -11	0.202E - 08	23
5	0.109E -08	0.289E -21	0.109E - 08	228
6	0.200E -10	0.348E -18	0.200E -10	45
7	0.199E -10	0.239E -25	0.199E -10	169
8	0.363E -12	0.229E -23	0.363E -12	29
9	0.110E -12	0.707E -18	0.109E -12	149
10	0.200E -14	0.339E-17	0.200E -14	79

Devemos enfatizar que para todos os 12 problemas testados , o algorítmo I , obteve a solução ótima da referência do Hock e Sdhittkowiski [34].

Este feito nos permite acreditar que o algorítmo I possa ter similar sucesso face a outros problemas práticos e da literatura .

CAPÍTULO 4

RESOLUÇÃO DO PROBLEMA GERAL DE PROGRAMAÇÃO NÃO - LINEAR VIA PENALIZAÇÃO HIPERBÓLICA

De posse dos resultados obtidos na resolução dos problemas de programação não-linear com restrições de igualdades , via penalização hiperbólica , apresentaremos neste capítulo um algorítmo para resolução de problemas de programação não-linear com restrições de igualdades e desigualdades .

Para resolvermos o problema geral de programação não-linear da forma :

$$\begin{aligned} &\text{minimizar} \quad f(x) \\ &\text{sujeito à :} \quad g_i(x) \geq 0 \quad , \; i = 1 , \ldots , \overline{m} \\ &\qquad\qquad\quad h_j(x) = 0 \quad , \; j = 1 , \ldots , \overline{p} \end{aligned} \tag{4.1}$$

transforma-se cada restrição de igualdade em duas restrições de desigualdades produzindo um problema rigorosamente equivalente ao problema (4.1) , isto é ,

$$\text{minimizar} \quad f(x) \qquad (4.2)$$

$$\text{sujeito à:} \quad g_i(x) \geq 0 \quad , \quad i = 1, \ldots, \bar{m}$$

$$h_j(x) \geq 0 \quad , \quad j = 1, \ldots, \bar{p}$$

$$h_j(x) \leq 0 \quad , \quad j = 1, \ldots, \bar{p}$$

Seguindo o mesmo argumento do método de penalização hiperbólica, descrito no capítulo 2 , relaxa-se as restrições de desigualdades provenientes de cada restrição de igualdade , colocando-se as folgas superiores ε_j^+ e inferiores ε_j^- obtendo-se o problema :

$$\text{minimizar} \quad f(x) \qquad (4.3)$$

$$\text{sujeito à:} \quad g_i(x) \geq 0 \quad , \quad i = 1, \ldots, \bar{m}$$

$$h_j(x) \geq \varepsilon_j^- \quad , \quad j = 1, \ldots, \bar{p}$$

$$h_j(x) \leq \varepsilon_j^+ \quad , \quad j = 1, \ldots, \bar{p} \; ,$$

com $\varepsilon_j^- \leq 0$ e $\varepsilon_j^+ \geq 0$, $j = 1, \ldots, \bar{p}$,

Pelo fato do problema (4.3) se encontrar na forma com somente restrições de desigualdades , aplica-se então o método de penalização hiperbólica , resolvendo-se uma sequência de problemas irrestritos da forma :

$$\text{minimizar} \quad f(x) + \sum_{i=1}^{m} P(\, l_i(x), \alpha_i, \tau_i \,), \qquad (4.4)$$

com $m = \overline{m} + 2\overline{p}$ e onde a função $l(.)$ é definida como :

$$l_i(x) = \begin{cases} g_i(x) & ; \quad i = 1, \ldots, \overline{m} \\ h_{i-m}(x) - \varepsilon_{i-m} & ; \quad i = \overline{m} + 1, \ldots, \overline{m} + \overline{p} \\ -h_{i-m-p}(x) + \varepsilon_{i-m-p} & ; \quad i = \overline{m} + \overline{p} + 1, \ldots, \overline{m} + 2\overline{p} \end{cases} \qquad (4.5)$$

A função de penalidade $P(.)$, define-se como anteriormente em (3.7) por :

$$P(l_i(x), \alpha, \tau) = -½ \sum_{j=1}^{m} tg\, \alpha_i . l_i(x) + \sum_{j=1}^{m} \sqrt{\left[\tfrac{1}{2}\, tg\alpha_i . l_i(x)\right]^2 + \tau_i^2} \qquad (4.6)$$

Inspirado no algorítmo **0** de Xavier [62] e no algorítmo **I** apresentado neste trabalho , no capítulo anterior , desenvolvidos para resolver , respectivamente , problemas de programação não-linear com restrições de desigualdades e problemas de programação não-linear com restrições de igualdades , apresentaremos a seguir o algorítmo **II** com o objetivo de resolver problemas de programação não-linear com restrições simultaneamente de igualdades e desigualdades .

ALGORÍTMO I I

PENALIZAÇÃO HIPERBÓLICA PARA RESTRIÇÕES DE IGUALDADES E DESIGUALDADES

1 - "INICIALIZAÇÃO"

Faça $k = 0$, $\alpha_j^1 = \alpha^o$, $j = 1, \ldots, p$, sendo $0 < \alpha^o < \pi/2$, $\tau^1 = \tau^o$ e $\tau^o > 0$

Tome ponto inicial x^o

$$\text{Calcule : } \begin{cases} \varepsilon_j^- = -\,100\,\tau^1 & , \quad j = 1, \ldots, p \\ \varepsilon_j^+ = +\,100\,\tau^1 & , \quad j = 1, \ldots, p \end{cases}$$

2 - "CONTAGEM DA ITERAÇÃO"

Faça $k := k + 1$

3 - "RESOLUÇÃO DO SUBPROBLEMA"

Resolver o problema de minimização irrestrita

$$\underset{x \in R^n}{\text{minimizar}} \; F(x, \alpha^k, \tau^k) = f(x) + \sum_{i=1}^{m} P(l_i(x), \alpha_i^k, \tau_i^k)$$

conforme definido em (4.4) a partir de um ponto inicial x^{k-1} , achando um ponto ótimo intermediário x^k .

4 - "TESTE DE VIABILIDADE"

Teste se o ponto x^k obtido é viável para o problema relaxado (4.3)

Se for viável, vá para o passo 6.

5 - "AUMENTO DO PARÂMETRO ÂNGULO α DA PENALIZAÇÃO "

Faça : $\begin{cases} \alpha_j^{\ k} := \rho.\alpha_j^{\ k} + (1 - \rho).\pi/2 \ , \ j = 1, \dots, p \quad e \quad 0 < \rho < 1 \\ \text{Vá para o passo 3} \end{cases}$

6 - "DIMINUIÇÃO DO PARÂMETRO DISTÂNCIA τ DA PENALIZAÇÃO"

Faça : $\begin{cases} \tau^{k+1} = q.\tau^k \\ \alpha^{k+1} = \alpha^k \end{cases}$, $0 < q < 1$, $q = 1/10$

7 - "CÁLCULO DAS DISTÂNCIAS SIGNIFICATIVAS "

Calcule as distâncias à esquerda e à direita do valor $h_j(x^k)$ às folgas :

$$d_j^- = +h_j(x^k) - \varepsilon_j^{k-} \quad , \quad j = 1, \dots, p$$

$$d_j^+ = -h_j(x^k) + \varepsilon_j^{k+} \quad , \quad j = 1, \dots, p$$

8 - "TESTE DE VIOLAÇÃO PARA CADA RESTRIÇÃO DE IGUALDADE "

Para cada restrição $j = 1, \dots, p$ faça :

8A - : ("Se o ponto x^k for tal que a distância à direita for significativamente maior do que a distância à esquerda,processa a diminuição da folga inferior")

Se $[\ d_j^+ > d_j^-\ ]$ e $[\ h_j(x^k) < \beta.\tau^k\ ]$; $\beta = 2.\sqrt{\dfrac{-1+\sqrt{5}}{2}}$

$$\text{Então faça :}\quad \begin{cases} \varepsilon_j^{k+1-} = \varepsilon_j^{k-}/10 & , \qquad (8A.1) \\ \varepsilon_j^{k+1+} = \varepsilon_j^{k+} & , \qquad (8A.2) \end{cases}$$

8B - :("Se oponto x^k for tal que a distância à esquerda for significativamente maior do que a distância à direita ,processa a diminuição da folga superior ")

Se $[\ d_j^- > d_j^+\ ]$ e $[\ h_j(x^k) > -\beta.\tau^k\ ]$; $\beta = 2.\sqrt{\dfrac{-1+\sqrt{5}}{2}}$

$$\text{Então faça :}\quad \begin{cases} \varepsilon_j^{k+1+} = \varepsilon_j^{k+}/10 & , \qquad (8B.1) \\ \varepsilon_j^{k+1-} = \varepsilon_j^{k-} & , \qquad (8B.2) \end{cases}$$

8C -:"Diminuição de ambas as folgas se não houver proximidade significativa"

Em caso contrário,

$$\text{Faça :}\quad \begin{cases} \varepsilon_j^{k+1-} = -\ 100\,\tau^k & (8C.1) \\ \varepsilon_j^{k+1+} = +100\,\tau^k & (8C.2) \\ \alpha_j^{k+1} = \rho.\alpha_j^k + (1-\rho).\pi/2 & (8C.3) \end{cases}$$

$$j = 1,\ldots,p \quad \text{e} \quad 0 < \rho < 1$$

Vá para o passo 2.

□

Neste ponto , devemos ressaltar que o algorítmo II é uma modificação do algorítmo I para incorporar as restrições de desigualdades .

Chamamos à atenção , que as únicas diferenças são : a definição da função objetivo modificada , no passo 3 , e o teste de viabillidade incluindo as restrições de desigualdades , no passo 4 .

ESCOLHA DOS PARÂMETROS INICIAIS

Os parâmetros α^{o} , ε^{+} , ε^{-} e τ^{o} do algorítmo II foram os mesmos escolhidos para o algorítmo I , conforme foi apresentado no capítulo 3 .

Como já referido anteriormente , em se tratando do parâmetro α^{o} , pequenos valores podem produzir divergência da minimização irrestrita . No problema-teste HS - 112 este efeito é particularmente suscetível , pois também envolve o parâmetro τ^{o} .

Normalmente os resultados computacionais são um tanto insensíveis a escolha inicial de τ^{o} . Entretanto , existem casos em que tal escolha se torna de fundamental importância pois , conforme especificação do algorítmo II , existe uma vinculação entre as folgas e o parâmetro τ^{o} .

Esse particular exemplo , HS -112 , possui uma função objetivo com uma coercividade intrínsica muito forte .

Assim , grandes folgas aumentam a região viável permitindo valores muito pequenos das componentes do vetor " x " , gerando valores extremamentes negativos do valor da função objetivo do problema HS - 112 , e entrando numa região em que a superioridade da coercividade da função objetivo em relação a penalização se estabelece de forma irreversível . Para o problema HS - 112 foram adotados os seguinte valores $\alpha^{o} = 0.900\,E+2$ e $\tau^{o} = 0.0001$

RESULTADOS COMPUTACIONAIS DOS PROBLEMAS RESTRITOS A IGUALDADES E DESIGUALDADES

Esta seção tem como objetivo a apresentação dos resultados computacionais obtidos com a implementação do método de penalização hiperbólica desenvolvido para solucionar os problemas de programação geral não-linear , sujeito a restrições de igualdades e desigualdades , usando o algorítmo II , acima descrito .

A fim de ilustrar em maiores detalhes o funcionamento do algorítmo II , são apresentados nas Tabelas 8 , 9 , 10 e 11 os resultados computacionais produzidos na resolução somente do problema teste :

EXEMPLO 2 :

$$\begin{aligned} \text{minimizar} \quad f(x) &= x_1^2 + x_2^2 + 2x_2 \\ \text{sujeito à :} \quad h(x) &= x_1^2 + x_2^2 - 1 = 0 \\ g_1(x) &= x_1 + 2x_2 - 0.5 \geq 0 \\ g_2(x) &= x_1 \geq 0 \\ g_3(x) &= x_2 \geq 0 , \end{aligned}$$

cuja solução ótima é o ponto $x^* = (1, 0)$ com $\lambda_1^* = 2$ e $\lambda_2^* = 1$

Para este particular problema , adotamos os seguintes parâmetros iniciais no passo 1 do algorítmo II :

$x^o = (1, 1)$, $\tau^o = 0.100000\,E-01$ e $\alpha^o = 0.114576\,E+01$

A descrição das Tabelas 8 , 9 , 10 e 11 é semelhante às apresentadas anteriormente .

TABELA 8 : Resultados Computacionais do Problema Teste 2 . Parte 1 / 4

Iteração k	α^k	τ^k	x_1^k	x_2^k	PASSOS
1	0.1145 E+01	0.10 E - 01	0.1455 E - 01	- 0.97260 E+ 00	9
2	0.2290 E+01	0.10 E - 01	0.2922 E - 01	- 0.9423 E + 00	7
3	0.4573 E+01	0.10 E - 01	0.5599 E - 01	-0.88197 E + 00	9
4	0.9090 E+01	0.10 E - 01	0.9573 E - 01	-0.7617 E + 00	7
5	0.1774 E+02	0.10 E - 01	0.1656 E+ 00	-0.5224 E + 00	6
6	0.3261 E+02	0.10 E - 01	0.3238 E+ 00	-0.6040 E - 01	11
7	0.5200 E+02	0.10 E - 01	0.4613 E+ 00	0.1549 E - 01	15
8	0.6866 E+02	0.10 E - 01	0.4720 E+ 00	0.1505 E - 02	18
9	0.6866 E+02	0.10 E - 02	0.9487 E+ 00	- 0.5315 E - 03	61
10	0.7894 E+02	0.10 E - 02	0.9488 E+ 00	0.8757 E - 04	28
11	0.7894 E+02	0.10 E - 03	0.9950 E+ 00	0.8757 E - 05	44
12	0.7894 E+02	0.10 E - 04	0.9995 E+ 00	0.8757 E - 06	122
13	0.7894 E+02	0.10 E - 05	0.9999 E+ 00	0.8756 E - 07	146
14	0.7894 E+02	0.10 E - 06	0.9999 E+ 00	0.8757 E - 08	170
15	0.7894 E+02	0.10 E - 07	0.9999 E+ 00	0.8757 E - 09	160
16	0.7894 E+02	0.10 E - 08	0.1000 E+ 00	0.8757 E - 10	320

TABELA 9 : Resultados Computacionais do Problema Teste 2. Parte 2 / 4

Iteração k	ε_k^-	$h(x^k)$	ε_k^+
1	- 0.10000 E+01	- 0.537671E -01	0.10000 E+01
2	- 0.10000 E+01	- 0.111098 E+00	0.10000 E+01
3	- 0.10000 E+01	- 0.219034E+00	0.10000 E+01
4	- 0.10000 E+01	- 0.410521E+00	0.10000 E+01
5	- 0.10000 E+01	- 0.699643E+00	0.10000 E+01
6	- 0.10000 E+01	- 0.891446E+00	0.10000 E+01
7	- 0.10000 E+01	- 0.786894E+00	0.10000 E+01
8	- 0.10000 E+01	- 0.776973.E+00	0.10000 E+01
9	- 0.10000E+ 00	- 0.998248E -01	0.10000 E+01
10	- 0.10000E+ 00	- 0.996997E -01	0.10000 E+01
11	- 0.10000 E -01	- 0.996997E -02	0.10000 E+01
12	- 0.10000 E -02	- 0.996997 E -03	0.10000 E+01
13	- 0.10000 E -03	- 0.996997 E -04	0.10000 E+01
14	- 0.10000 E -04	- 0.996997E -05	0.100000E+01
15	- 0.10000 E -05	- 0.996997E -06	0.100000E+01
16	- 0.10000 E -06	- 0.996997E -07	0.100000E+01

TABELA 10 : Resultados Computacionais do Problema Teste 2. Parte 3 / 4

Iteração k	λ_1^k	λ_2^k
1	0.169723R-01	0.312690E-02
2	0.376672E-01	0.256843E-02
3	0.784838E-01	0.190432E-02
4	0.158944E+00	0.335541E-02
5	0.318867E+00	0.124884E-01
6	0.604215E+00	0.171337E-02
7	0.189299E+00	0.784577E-03
8	0.143831E+00	0.999999E+00
9	0.199999E+01	0.999999E+00
10	0.199999E+01	0.1000000E+01
11	0.200000E+01	0.100000E+01
12	0.200000E+01	0.100000E+01
13	0.200000E+01	0.100000E+01
14	0.200000E+01	0.100000E+01
15	0.200000E+01	0.100000E+01
16	0.200000E+01	0.100000E+01

TABELA 11 : Resultados Computacionais do Problema Teste 2. Parte 4 / 4

Iteração k	$F(x^k, \alpha^k, \tau^k)$	$P(x^k, \alpha^k, \tau^k)$	$f(x^k)$
1	- 0.906622E+00	0.924172E - 01	- 0.999039E+00
2	- 0.546172E+00	0.149651E + 00	- 0.995823E+00
3	- 0.723201E+00	0.259977E + 00	- 0.982978E+00
4	- 0.496182E+00	0.437905E + 00	- 0.934088E+00
5	- 0.132102E+00	0.612395E + 00	- 0.744977E+00
6	0.221364E+00	0.233614E + 00	- 0.122499E+ 00
7	0.264859E+00	0.207566E - 01	0.244102E+00
8	0.263498E+00	0.103571E - 01	0.253140E+00
9	0.901804E+00	0.269209E - 02	0.899112E+00
10	0.901769E+00	0.129412E - 02	0.900475E+00
11	0.990176E+00	0.129338E - 03	0.990047E+00
12	0.999017E+00	0.129331E - 04	0.999004E+00
13	0.999901E+00	0.129330E - 05	0.999900E+00
14	0.999990E+00	0.129330E - 06	0.999990E+00
15	0.999999E+00	0.129330E - 07	0.999999E+00
16	0.100000E+01	0.129329E - 08	0.100000E+01

Para uma melhor avaliação da eficácia do algorítmo II, seguindo a recomendação dada em " Ad Hoc Committee to Revise Guidelines for Reporting Computational Experiments in Mathematical Programming " da Mathematical Programming Society, vide Jackson et all. [36], selecionamos um conjunto de problemas-teste do livro Hock e Schittkowski [34].

Unicamente devido a questões de concisão, nos atemos aos 15 problemas abaixo descritos, observando que nos demais o desempenho do algorítmo II foi análogo.

Problema 32

$$\text{minimizar} \quad (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2$$

$$\text{sujeito à :} \quad x_i \geq 0 \quad , \ i = 1, 2, 3$$

$$6x_2 + 4x_3 - x_1^3 - 3 \geq 0$$

$$1 - x_1 - x_2 - x_3 = 0$$

Ponto inicial $x^o = (0.1 , 0.7 , 0.2)$

Solução ótima encontrada na 9ª iteração em 1.92 segundos de CPU

$x^* = (0, 0, 1)$, $\lambda_1^* = 4.0032$, $\lambda_2^* = 1.9988$ e $f(x^*) = 1$

Problema 41

$$\text{minimizar} \quad 2 - x_1 x_2 x_3$$

$$\text{sujeito à :} \quad x_1 + 2x_2 + 2x_3 - x_4 = 0$$

$$0 \leq x_i \leq 1 \quad , \ i = 1, 2, 3$$

$$0 \leq x_4 \leq 2$$

Ponto inicial $x^o = (2, 2, 2, 2)$

Solução ótima encontrada na 6ª iteração em 1.60 segundos de CPU.

$x^* = (2/3 , 1/3 , 1/3 , 2)$, $\lambda^* = 0.111$ e $f(x^*) = 1.9259259$

Problema 53

minimizar $(x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$

sujeito à: $x_1 + 3x_2 = 0$

$x_3 + x_4 - 2x_5 = 0$

$x_2 - x_5 = 0$

$-10 \leq x_i \leq 10 \quad , \quad i = 1, \ldots, 5$

Ponto inicial $x^o = (2, 2, 2, 2, 2)$

Solução ótima encontrada na 9ª iteração em 2.95 segundos de CPU.

$x^* = 1/43(-33, 11, 27, -5, 11)$, $f(x^*) = 4.093023$

$\lambda_1^* = 5.9535$, $\lambda_2^* = 2.0465$ e $\lambda_3^* = 2.05$

Problema 55

minimizar $x_1 + 2x_2 + 4x_5 + \exp(x_1.x_4)$

sujeito à: $x_1 + 2x_2 + 5x_6 - 6 = 0$

$x_1 + x_2 + x_3 - 3 = 0$

$x_4 + x_5 + x_6 - 2 = 0$

$x_1 + x_4 - 1 = 0$

$x_2 + x_5 - 2 = 0$

$x_3 + x_6 - 2 = 0$

$x_i \geq 0 \quad , \quad i = 1, \ldots, 6$

$x_1 \leq 1 \quad , \quad x_4 \leq 1$

Ponto inicial $x^o = (1, 2, 0, 0, 0, 2)$

Solução ótima encontrada na 9ª iteração em 3.52 segundos de CPU.

$x^* = (1, 4/3, 1/3, 0, 1/3, 5/3)$, $\lambda^*_{max} = 0.6666$ e $\lambda^*_{min} = 0.3332$

$f(x^*) = 6.3333$

Problema 60

$$\text{minimizar} \quad (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4$$

$$\text{sujeito à:} \quad x_1(1 + x_2^2) + x_3^4 - 4 - 3\sqrt{2} = 0$$

$$-10 \leq x_i \leq 10 \quad , \quad i = 1, \dots, 10$$

Ponto inicial $x^o = (2, 2, 2)$

Solução ótima encontrada na 7ª iteração em 1.71 segundos de CPU.

$x^* = (1.104859, 1.196674, 1.535262)$ e $f(x^*) = 0.032568$

Problema 63

$$\text{minimizar} \quad 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1.x_2 - x_1.x_4$$

$$\text{sujeito à:} \quad x_i \geq 0 \quad ; \quad i = 1, 2, 3$$

$$8x_1 + 14x_2 + 7x_3 - 56 = 0$$

$$x_1^2 + x_2^2 + x_3^2 - 25 = 0$$

Ponto inicial $x^o = (2, 2, 2)$. Solução ótima encontrada na 5ª iteração em 1.26 segundos de CPU. $x^* = (3.512118, 0.216988, 3.552174)$ $\lambda_1^* = 0.2749$, $\lambda_2^* = 1.2234$ e $f(x^*) = 961.7154$

Problema 71

$$\text{minimizar} \quad x_1x_4(x_1 + x_2 + x_3) + x_3$$

$$\text{sujeito à:} \quad x_1 x_2 x_3 x_4 - 25 \geq 0$$

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 = 0$$

$$1 \leq x_i \leq 5 \quad , \quad i = 1, \dots, 4$$

Ponto inicial $x^o = (1, 5, 5, 1)$. Solução ótima encontrada na 5ª iteração em 1.49 segundos de CPU. $x^* = (1, 4.742999, 3.821150, 1.379408)$

$\lambda_1^* = 1.0879$, $\lambda_2^* = 0.1615$ e $f(x^*) = 17.0140$

Problema 1 do Bazaraa [6]

$$\text{minimizar} \quad (x_1 - 2)^2 + \tfrac{1}{4} x_2^2$$

$$\text{sujeito à :} \quad -x_1 + \tfrac{1}{2} x_2 + 1 \geq 0$$

$$2x_1 + 3x_2 - 4 = 0$$

Ponto inicial $x^o = (2, 0)$. Solução ótima encontrada na 7ª iteração em 1.32 segundos de CPU. $x^* = (1.7187, 0.1875)$ com $\lambda_1^* = 0.0$ e $\lambda_2^* = 0.2$

Problema 2 do Bazaraa [6]

$$\text{minimizar} \quad x_1^2 + 4x_2^2$$

$$\text{sujeito à :} \quad x_i \geq 0 \quad ; \quad i = 1, 2, 3, 4$$

$$x_1 + 2x_2 - x_3 - 1 = 0$$

$$-x_1 + x_2 + x_4 = 0$$

Ponto inicial $x^o = (2, 1, 3, 1)$

Solução ótima encontrada na 8ª iteração em 1.92 segundos de CPU

$x^* = (0.5, 0.25, 0, 0.25)$ com $\lambda_1^* = 1$ e $\lambda_2^* = 1$

Problema 73

$$\text{minimizar} \quad 24.55x_1 + 26.75x_2 + 39x_3 + 40.50x_4$$

$$\text{sujeito à :} \quad 2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 - 5 \geq 0$$

$$12x_1 + 11.9x_2 + 41.8x_3 + 52.1x_4 - 21 -$$

$$1.645\,(0.28x_1^2 + 0.19x_2^2 + 20.5x_3^2 + 0.62x_4^2)^{3/2} \geq 0$$

$$x_1 + x_2 + x_3 + x_4 - 1 = 0$$

$$x_i \geq 0 \ , \quad i = 1, \ldots, 4$$

Ponto inicial $x^o = (1, 1, 1, 1)$

Solução encontrada na 9ª iteração em 2.20 segundos de CPU.

$x^* = (0.6355214, 0.3064E\text{-}10, 0.3127019, 0.05177656)$, $f(x^*) = 29.8943$

$\lambda_1^* = 0.2433$, $\lambda_2^* = 0.411$, $\lambda_3^* = 0.580$ e $\lambda_4^* = 18.37$

Problema 81

$$\text{minimizar} \quad \exp(x_1 x_2 x_3 x_4 x_5) - 0.5\,(x_1^3 + x_2^3 + 1)^2$$

$$\text{sujeito à:} \quad -2.3 \le x_i \le 2.3 \quad ; \quad i = 1, 2$$

$$-3.2 \le x_j \le 3.2 \quad ; \quad j = 3, 4, 5$$

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$x_2 x_3 - 5 x_4 x_5 = 0$$

$$x_1^3 + x_2^3 + 1 = 0$$

Ponto inicial $x^o = (-2, 2, 2, -1, -1)$

Solução ótima encontrada na 3ª em 1.21 segundos de CPU

$x^* = (-1.717142, 1.159541, 1.827248, -0.763647, -0.763639)$

$\lambda_1^* = 0.0373$, $\lambda_2^* = 0.0396$, $\lambda_3^* = 0.0495$ e $f(x^*) = 0.05394$

Problema 111

$$\text{minimizar} \quad f(x) = \sum_{j=1}^{10} \exp(x_j).\{ c_j + x_j - \ln[\sum_{k=1}^{10} \exp(x_k)] \}$$

$$\text{sujeito à:} \quad \exp(x_1) + 2\exp(x_2) + 2\exp(x_3) + \exp(x_6) + \exp(x_{10}) - 2 = 0$$

$$\exp(x_4) + 2\exp(x_5) + \exp(x_6) + \exp(x_7) - 1 = 0$$

$$\exp(x_3) + \exp(x_7) + \exp(x_8) + 2\exp(x_9) + \exp(x_{10}) - 1 = 0$$

$$-100 \le x_i \le 100 \quad , \quad i = 1, \dots, 10$$

Valores de c_j , $j = 1, \ldots, 10$

$c_1 = -6.089$, $c_2 = -17.164$, $c_3 = -34.054$, $c_4 = -5.914$, $c_5 = -24.721$

$c_6 = -14.986$, $c_7 = -24.100$, $c_8 = -10.708$, $c_9 = -26.662$, $c_{10} = -22.179$

Ponto inicial $x^o = (-2.3 , \ldots , -2.3)$

Solução ótima encontrada na 7ª iteração em 5.21 segundos de CPU.

$x^* = (-3.2030 , -1.9123 , -0.2444 , -23.73 , -0.7216 , -7.2645 , -3.5959 ,$
$-4.0190 , -3.2895 , -2.3337)$

$\lambda^*_{max} = 15.57$, $\lambda^*_{min} = 10.59$ e $f(x^*) = -47.75967$

Problema 112

minimizar $f(x) = \sum_{j=1}^{10} x_j \left[c_j + \ln\left(\frac{x_j}{x_1 + \ldots + x_{10}} \right) \right]$

sujeito à : $x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$

$x_4 + 2x_5 + x_6 + x_7 - 1 = 0$

$x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0$

$x_i \geq 1.E{-6}$, $i = 1, \ldots, 10$

Valores de c_j , $j = 1, \ldots, 10$

$c_1 = -6.089$, $c_2 = -17.164$, $c_3 = -34.054$, $c_4 = -5.914$, $c_5 = -24.721$

$c_6 = -14.986$, $c_7 = -24.100$, $c_8 = -10.708$, $c_9 = -26.662$, $c_{10} = -22.179$

Ponto inicial $x^o = (0.1 , \ldots , 0.1)$

Solução ótima encontrada na 7ª iteração em 2.75 segundos de CPU .

$x^* = (0.046680 , 0.147730 , 0.783153 , 0.001414 , 0.485246 , 0.000693 ,$

$0.027399 , 0.017947 , 0.037314 , 0.096871)$

$\lambda^*_{max} = 15.20$, $\lambda^*_{min} = 9.774$ e $f(x^*) = -47.76109$

Problema 114

minimizar $5.04\, x_1 + 0.035\, x_2 + 10\, x_3 + 3.36\, x_5 - 0.063\, x_4 x_7$

sujeito à : $g_1(x) = 35.82 - 0.222\, x_{10} - b\, x_9 \geq 0$

$g_2(x) = -133 + 3\, x_7 - a\, x_{10} \geq 0$

$g_3(x) = -g_1(x) + (1/b - b)\, x_9 \geq 0$

$g_4(x) = -g_2(x) + (1/a - a)\, x_{10} \geq 0$

$g_5(x) = 1.12\, x_1 + 0.13167\, x_1 x_8 - 0.00667\, x_1 x_8^2 - a\, x_4 \geq 0$

$g_6(x) = 57.425 + 1.098\, x_8 - 0.038\, x_8^2 + 0.325\, x_6 - a\, x_7 \geq 0$

$g_7(x) = -g_5(x) + (1/a - a)\, x_4 \geq 0$

$g_8(x) = -g_6(x) + (1/a - a)\, x_7 \geq 0$ $\quad a = 0.99$

$h_1(x) = 1.22\, x_4 - x_1 - x_5 = 0$ $\quad b = 0.9$

$h_2(x) = 98000\, x_3 / (x_4 x_9 + 1000\, x_3) - x_6 = 0$

$h_3(x) = (x_2 + x_5) / x_1 - x_8 = 0$

$0.00001 \leq x_1 \leq 2000$, $85 \leq x_6 \leq 93$

$0.00001 \leq x_2 \leq 16000$, $90 \leq x_7 \leq 95$

$0.00001 \leq x_3 \leq 1200$, $3 \leq x_8 \leq 12$

$0.00001 \leq x_4 \leq 5000$, $1.2 \leq x_9 \leq 4$

$0.00001 \leq x_5 \leq 2000$, $145 \leq x_{10} \leq 162$

Ponto inicial x^o = (1745 , 12000 , 110 , 3048 , 1974 , 89.2 , 92.8 , 8 , 3.6 , 145)

Solução ótima encontrada na 9ª iteração em 5.93 segundos de CPU.

x^* = (1698.091 , 15816.30 , 54.0974 , 3031.223 , 2000 , 90.1157 , 95 , 10.492 , 1.5616 , 153.535) , $f(x^*) = -1768.74$,

$\lambda^*_{max} = 311.77$ e $\lambda^*_{min} = 0.6815$

Problema 119

$$\text{minimizar} \quad \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij} (x_i^2 + x_i + 1)(x_j^2 + x_j + 1)$$

$$\text{sujeito à :} \quad \sum_{j=1}^{16} b_{ij} x_j - c_i = 0$$

$$0 \le x_i \le 5 \quad , \quad i = 1, \dots, 16$$

Valores de a_{ij} , b_{ij} e c_i nas tabelas em anexo

Ponto inicial x^o = (10 , ... , 10)

Solução ótima encontrada na 15ª iteração em 16.20 segundos de CPU.

x^* = (0.039847 , 0.791983 , 0.202870 , 0.844357 , 1.269906 , 0.934739 , 1.681962 , 0.155294 , 1.567870 , 0 , 0 , 0 , 0.660203 , 0 , 0.674255 , 0)

$\lambda^*_{max} = 30.03$, $\lambda^*_{min} = 0.383$ e $f(x^*) = 0.244899697$

Tabela 12 : Valores de a_{ij} para o problema 119.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_{1j}	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1
a_{2j}	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0
a_{3j}	0	0	1	0	0	0	1	0	1	1	0	0	0	1	0	0
a_{4j}	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0
a_{5j}	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1
a_{6j}	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
a_{7j}	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
a_{8j}	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0
a_{9j}	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1
a_{10j}	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
a_{11j}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
a_{12j}	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
a_{13j}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
a_{14j}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
a_{15j}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
a_{16j}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tabela 13 : Valores de b_{ij} e c_j para o problema 119:

j	b_{1j}	b_{2j}	b_{3j}	b_{4j}	b_{5j}	b_{6j}	b_{7j}	b_{8j}	c_j
1	0.22	-1.46	-1.29	-1.10	0	0	1.12	0	2.5
2	0.20	0	-0.89	-1.06	0	-1.72	0	0.45	1.1
3	0.19	-1.30	0	0.95	0	-0.33	0	0.26	-3.1
4	0.25	1.82	0	-0.54	-1.43	0	0.31	-1.10	-3.5
5	0.15	-1.15	-1.16	0	1.51	1.62	0	0.58	1.3
6	0.11	0	-0.96	-1.78	0.59	1.24	0	0	2.1
7	0.12	0.80	0	-0.41	-0.33	0.21	1.12	-1.03	2.3
8	0.13	0	-0.49	0	-0.43	-0.26	0	0.10	-1.5
9	1.0	0	0	0	0	0	-0.36	0	
10	0	1.0	0	0	0	0	0	0	
11	0	0	1.0	0	0	0	0	0	
12	0	0	0	1.0	0	0	0	0	
13	0	0	0	0	1.0	0	0	0	
14	0	0	0	0	0	1.0	0	0	
15	0	0	0	0	0	0	1.0	0	
16	0	0	0	0	0	0	0	1.0	

CAPÍTULO 5

CONCLUSÕES

Neste trabalho , conforme proposta apresentada na introdução , consideramos primeiramente o problema de programação não- linear sujeito a restrições de igualdades. Foi apresentado um novo método para resolução desse problema , que se fundamentou na utilização da penalização hiperbólica. Esse método se constitui na relaxação das restrições de igualdades , através de folgas superiores e inferiores , criando uma região viável com interior não-vazio . Seguindo essas idéias , foi apresentado o algorítmo I para resolução do problema acima citado , e foi desenvolvida a correspondente teoria de convergência.

Na avaliação dessa metodologia , resolvemos vários problemas testes relacionados no livro de Hock e Shittkowiski [34]. Os resultados obtidos se comportaram conforme a teoria desenvolvida .

Ressaltamos que até mesmo os experimentos realizados com problemas testes degenerados (que ferem às condições pré-estabelecidos para desenvolvimento da teoria de convergência) foram coroadas de pleno sucesso no que se refere ao critério de robustez e , até mesmo , de eficiência . Todavia , para problemas maiores avaliamos que o comportamento oscilatório dos valores das folgas superiores e inferiores podem diminuir a eficiência do método.

Através da consolidação dos resultados previamente estabelecidos , estendemos nossos estudos à resolução do problema geral de programação não-linear sujeito a restrições de igualdades e desigualdades. Apresentamos o algorítmo II e o submetemos a um conjunto de problemas testes , também relacionados em Hock e Schittkowiski [34] , onde obtivemos igual sucesso na resolução desses problemas.

Fundamentado no sucesso demonstrado nesse conjunto de problemas resolvidos , acreditamos que a presente metodologia se constitui em uma alternativa válida para resolução de problemas práticos.

Com relação ao problema geral de programação não-linear faz-se , todavia , necessário um desenvolvimento teórico analisando suas propriedades de convergência. Tais estudos podem ter como ponto de partida as idéias relacionadas nos trabalhos previamente desenvolvidos sobre resolução do problema restrito a desigualdades , descrito por Xavier [62] em combinação com a teoria apresentada para resolução do problema restrito a igualdades , descrito neste trabalho.

Devemos agora destacar algumas características da metodologia adotada que julgamos ser de particular importância.

DIFERENCIABILIDADE

Uma primeira característica é que a função de penalidade hiperbólica $P(y, \lambda, \tau)$ é continuamente diferenciável , qualquer que seja a ordem da derivada considerada , e segundo quaisquer das variáveis y , λ ou τ , isto é , $P \in C^{\infty}$. Assim , a diferenciabilidade intrínsica oferecida pela funçào penalidade permite o perfeito funcionamento dos métodos que fazem o uso de informações das derivadas de primeira ordem , gradientes , e de segunda ordem , matriz hessiana .

PONTO INICIAL

Uma outra característica se relaciona com o ponto inicial, pois não existe a necessidade de um ponto especial satisfazendo quer as restrições de igualdades quer as restrições de desigualdades, para o início do processamento dos algorítmos.

COMBINAÇÃO DA PENALIZAÇÃO HIPERBÓLICA COM OUTROS MÉTODOS

A despeito desses pontos positivos, ressaltados, temos clareza que essa metodologia não se constitui em um recurso necessariamente ótimo e fechado para resolver qualquer tipo de problema, e em particular, para tratar de uma forma única qualquer tipo de restrição de igualdade ou de desigualdade.

É possível fazer uma combinação da penalização hiperbólica com o método do gradiente reduzido generalizado. Tal combinação poderia ser feita da seguinte maneira: somente as restrições não-lineares seriam tratadas pela penalização hiperbólica, enquanto que as restrições lineares seriam tratadas dentro do enfoque GRG. Produziríamos assim, um problema penalizado unicamente com restrições lineares. Para esse problema, usaríamos o método GRG, que conforme sobejamente registrado na literatura possui um excelente desempenho.

Alternativamente ao método GRG, poderíamos de uma forma similar considerar a combinação com outros métodos primais como: gradiente projetado, gradiente reduzido e, até mesmo, direções viáveis.

Da mesma forma podemos conjecturar a possibilidade de combinarmos com o método de programação quadrática sequencial , vide Martinez [43] , usando-se a penalização hiperbólica como função mérito .

LAGRANGEANO HIPERBÓLICO

Outra extensão natural da metodologia é fazer uso das informações dos multiplicadores de Lagrange dadas pela penalização hiperbólica de modo a criar um algorítmo tipo Lagrangeano Aumentado , conforme detalhado em Xavier [63] como também em Xavier e Maculan [65] .

EXPERIMENTAÇÃO COMPUTACIONAL

A despeito dos resultados computacionais realizados propomos um aprofundamento da experimentação computacional tratando da eficiência , medida em tempo de CPU , do número de iterações ou da comparação com outros algorítmos alternativos, seguindo rigorosamente os cânones especificados em JACKSON et al. [36].

EXTRAPOLAÇÃO

Finalmente , propomos um desenvolvimento da teoria e implementação de mecanismos de extrapolação , que os resultados computacionais obtidos insinuam claramente seguindo as mesmas linhas descritas em Xavier e Maculan [66] .

Essas serão algumas das linhas de estudos que deverão desenvolver este trabalho.

BIBLIOGRAFIA

[1] ABADIE, J. & CARPENTIER, J. - *Generalization of the Wolfe Reduced Gradient Method to the case of Nonlinear Constraints, In: Optimization,* ed. R. Fletcher, Academica Press, London, pp. 37-47, 1969.

[2] ABADIE, J. & GULGOU, J. - *Numerical Experiments with the GRG Method, In: Integer and Nonlinear Programming,* ed. J. Abadie, North-Holland, Amsterdan, pp. 520-536, 1970.

[3] ARROW, K. J. ; GOULD,F..J. & HOWE, S. M. - *A General Saddle Point Result for Constraints Optimization* . Mathematical Programming , Vol.5 , pp. 225-234, 1973.

[4] AVRIEL, M. - *Nonlinear Programming Analysis and Methods.* Prentice - Hall , Englewood Cliffs . 1976.

[5] BARD, Y. & GREENSTADT, J. L. - *A Modified Newton Method for Optimization with Equality Constraints* , ed. R. Fletcher , Academic Press , London , New York , pp. 299 - 306 , 1969.

[6] BAZARAA, M. S. ; SHERALI, H. D. & SHETTY, C. M. - *Nonlinear Programming , Theory and Algorithms.* John Wiley and Sons, New York, Second, 1973.

[7] BEALE, E. M. L. - *Numerical Methods in Nonlinear Programming Problems with Simple Constraints.* SIAM Journal on Control and Optimization , 20 , pp. 141-148, 1967.

[8] BEALE, E. M. L. - *Numerical Methods in Nonlinear Programming.* J. Abadie ed. North - Holland , Amsterdan , 1967.

[9] BERTSEKAS, D. P. - *Combined Primal Dual and Penalty Methods for Constrained Minimization.* SIAM J. Control , pp. 521 - 544 , 1975a.

[10] BERTSEKAS, D. P. - *On Penalty and Multiplier Methods for Constrained Minimization.* SIAM J. Control and Optimization, vol. 14, pp. 216 - 235 , 1975a.

[11] BERTSEKAS, D. P. - *Multiplier Methods : A Survey* . Automatica, pp. 133 - 143 , 1975b.

[12] BERTSEKAS, D. P. - *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press , San Diego , 1982.

[13] BIGGS , M. C. - *Constrained Minimization Using Recursive Equality Quadratic Programming in Numerical Methods for Nonlinear Optimization.* F. A. Lootsma ed. , Academic Press, London , New York , pp. 411 - 428 , 1972.

[14] BOX , M. J. ; DAVIES , D. & SWANN , W. H. - *Nonlinear Optimization Technique.* ICI. Monograph , Oliver and Boyd , Edinburgh , 1969.

[15] CAMP , G. D. - *Inequality Constrained Stationary Value Problems.* Operations Research , 3 , pp. 548 - 550 , 1955.

[16] CARROL , C. W. - *An Operations Research Approach to the Economic Optimization of a Kraft Pulping Process* , Ph.D. Thesis , Institute of Paper Chemistry, Appleton, Winsconsin , 1959.

[17] CARROL , C. W. - *The Created Response Surface Technique for Optimizing Nonlinear Systems* . Operations Research ,vol. 9 , pp. 169 - 184 , 1961.

[18] CASTILHO CÁRDENAS , R . A . - *Métodos de Lagrangeano Aumentado Usando Funções de Penalidades Generalizadas para Problemas de Programação Não-Linear* . Tese de D . Sc . , COPPE / UFRJ , Rio de Janeiro , Brasil , 1998 .

[19] COLVILLE, A. R. - *A Comparative Study on Problems Programming Codes, In : Proceedings of the Princeton Symposium on Mathematical Programming.* ed. H. W. Kuhn , Princeton University Press , 1970.

[20] COURANT , R. - *Variational Methods for the Solution of Problems of Equilibrium and Vibrations* . Bull. Ann. Math. Society, 49 , pp. 1 - 23 , 1943.

[21] EL - ALEM , M. M. - *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization* . Ph.D. Thesis , Rice University , Houston Texas , 1988.

[22] EVANS , J. P. ; GOULD , F. J. & TOLLE , J. W. - *Exact Penalty Functions in Nonlinear Programming* . Math. Progr. , 4 , pp. 72 - 97 , 1973.

[23] FIACCO , A. V. & McCORMICK , G. P. - *The Sequential Unconstrained Minimization Technique for Nonlinear Programming : A Primal Method* . Management Science , Vol. 10 , pp. 360 - 364 , 1964.

[24] FIACCO , A. V. & McCORMICK , G. P. - *Extensions of SUMT for Nonlinear Programming : Equality Constraints and Extrapolation.* Management Science . Vol 12 , pp. 816 - 828 , 1966.

[25] FIACCO , A. V. & McCORMICK , G. P. - *Nonlinear Programming Sequential Unconstrained Minimization Technique* . Jonh Wiley , New York , 1968 .

[26] FLETCHER , R. - *An Ideal Penalty Function for Constrained Optimization* . Journal of the Institute of Mathematics and its Applications . Vol 5 , pp. 319 - 342 , 1975.

[27] FLETCHER , R. - *Practical methods of Optimization , Constrained Optimization* . Vol 2 , John Wiley & Sons , New York and Toronto , 1981.

[28] FLETCHER , R. - *Penalty Functions , In Mathematical Programming - The State of Art* . (A. Bachen , M. Grotschel e B. Komte , editores) , Springer - Verlag , Berlin , pp. 87 - 144 , 1983.

[29] FRISCH , K. R. - *The Logarithmic Potential Method of Convex Programming.* University Institute of Economics Oslo , Memorandum , may 13 , 1955.

[30] GILL , P. E. ; MURRAY , W. & WRIGHT , M. H. - *Practical Optimization* . Academic Press , London and New York , 1981.

[31] HAARHOFF , P. C. & BUYS , J. D. - *A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints.* The Computer Journal , vol. 134 , nº 2 , pp. 178 - 184 , 1970.

[32] HESTENES , M. R. - *Multiplier and Gradient Methods.* Journal of Optimization Theory and Applications , vol. 4 , pp. 303 - 320 , 1969.

[33] HIMMELBLAU , D. M. - *Applied Nonlinear Programming.* McGraw - Hill , New York , 1972.

[34] HOCH , W. & SCHITTKOWSKI , K. - *The Examples for Nonlinear Programming Codes* . Springer - Verlag , Berlin and Heidelberg , 1981.

[35] HUARD , P. - *Resolution of Mathematical Programming with Nonlinear Constraints by the Methods of Centers* . In Nonlinear Programming , J. Abadie ed. , 1967.

[36] JACKSON , R. H. F. ; CHAIR ; PAUL T. BOGGS ; STEPHEN ; G. NASH & SUSAN POWELL - *Report of the Ad Hoc Committee to Revise the Guidelines for Reporting Computational Experiments* , In Mathematical Programming , 1989.

[37] KORT , B. W. & BERTSEKAS , D. P. - *Combined Primal Dual and Penalty Methods for Convex Programming* . SIAM Journal Control and Optimization , vol. 14 , pp. 268 - 294 , 1976.

[38] KOWALIK , J. - *Nonlinear Procedures and Desigin Optimization* . Acta Polytechica Scandinavica , 13 , pp. 1 - 47 , 1966.

[39] KUHN , W. W. & TUCKER , A. W. - *Nonlinear Programming , Proc. 2nd Berkeley Symp. on Mathematical Statistics and Probability.* University of California Press , Berkeley , pp. 481 - 492 , 1951.

[40] LOOTSMA , F. A. - *Constrained Optimization via Penalty Functions.* Philips Research Reports , 23 , pp. 408 - 423 , 1968.

[41] LOOTSMA , F. A. - *A Survey of Methods for Solving Constrained Minimization Problems via Unconstrained Minimization . In: Numerical Method for Nonlinear Optimizations* . ed. F. A. Lootsma , Academic Press , London , pp. 313 - 347, 1972.

[42] LUENBERGER , D. G. - *Introduction to Linear and Nonlinear Programming.* Addison Wesley , Menlo Park , 1973.

[43] MARTINEZ , J. M. & SANTOS , S. A. - *Métodos Computacionais de Otimização ,* 20º Colóquio Brasileiro de Matemática , IMPA , Rio de Janeiro , Brasil , 1995.

[44] MIELE , A. ; GRAGG , E. E. ; IYER , R. R. & LEVY , A. V. - *Use the Augmented Penalty Function in Mathematical Programming Problems* . Journal of Optimization Theory and Applications . Vol. 8 , pp. 115 - 130 , 1971a .

[45] MIELE , A. ; GRAGG, E. E. & LEVY , A. V. - *Use the Augmented Penalty Function in Mathematical Programming Problems , Part 2* . Journal of Optimization Theory and Applications . Vol. 8 , pp. 131 - 153 , 1971b.

[46] MIELE , A. ; MOSELEY , P. E. & GRAGG , E. E. - *A Modification of the Method of Multipliers for Mathematical Programming Problems , In Techniques of Optimization.* Balakrishnan A. V. ed. , Academic Press , New York , pp. 247 - 260 , 1972a.

[47] MIELE , A. ; MOSELEY , P. E. ; LEVY , A. V. & COGGIN , G. M. - *On the Method of Multipliers for Mathematical Programming Problems.* Journal of Optimization Theory and Applications. Vol. 10 , pp. 01 - 33 , 1972b.

[48] MINOUX , M. - *Mathematical Programming Theory and Algorithms.* John Wiley and Sons , Chichester , 1986.

[49] OSBORNE , M.R. & RYAN , D. M. - *On Penalty Function Methods for Nonlinear Programming Problems.* J. Math. Analysis and Applications.,31, pp.557-578 , 1970

[50] PIERRE , D. A. & LOWE , M. J. - *Mathematical Programming via Augmented Lagrangians , An Introduction with Computer Programs.* Addison - Wesley, Reading , 1975.

[51] POLAK , E. - *Computational Methods in Optimization , A Unified Apprach.* Academic Press , New York , 1971.

[52] POWELL , M. J. D. - *A Method for Nonlinear Constraints in Minimization Problems, In Optimization.* Fletcher R. ed. , Academic Press , London , pp. 283 - 298 , 1969.

[53] POWELL , M. J. D. - *Algorithms for Nonlinear Constraints that use Lagrangian Functions.* Mathematical Programming. Vol. 14 , pp. 224 - 248 , 1978.

[54] ROCKAFELLAR , R. T. - *Convex Analysis* . Princeton University Press , Princeton , New Jersey , 1970.

[55] ROCKAFELLAR , R. T. - *A Dual Approach in Solving Nonlinear Programming Problems by Unconstrained Optmization* . Mathematical Programming. Vol. 5 , pp. 354 - 373 , 1973a.

[56] ROCKAFELLAR , R. T. - *The Multiplier Method of Hestenes and Powell Applied to Convex Programming.* Journal of Optimization Theory and Applications. Vol. 12 , nº 6 , pp. 555 - 562 , 1973b.

[57] ROCKAFELLAR , R. T. - *Augmented Lagrange Multiplier and Duality in Nonconvex Programming* . SIAM Journal of Control. Vol. 12 , pp. 268 - 285 , 1974.

[58] ROSEN , J. B. - *The Gradient Projection for Nonlinear Programming - Part 1 : Linear Constraints* . Journal Soc. Ind. Appl. Math. Vol. 8 , pp. 181 - 217 , 1960.

[59] ROSEN , R. T. - *The Gradient Projection for Nonlinear Programming - Part II : Nomlinear Constraints* . Journal Soc. Ind. Appl. Math.. Vol. 9 , pp. 514 - 532 , 1961.

[60] WILSON , R. B. - *A Simplicial Algorithm for Concave Programming.* Unpublished Ph.D. Dissertation, Haward University Graduate School of Business Administration Boston , 1963.

[61] WOLFE , P. - *Methods of Nonlinear Programming . The Reduced Gradient Method, In : Recent Advances in Mathematical Programming.* eds. Graves and Wolfe, McGraw-Hill , New York , pp. 67 - 86 , 1963.

[62] XAVIER , A. E. - *Penalização Hiperbólica - Um Novo Método para Resolução de Problemas de Otimização* . Tese de M. Sc. COPPE / UFRJ , Rio de Janeiro , Brasil , 1982.

[63] XAVIER , A. E. - *Penalização Hiperbólica - Uma Abordagem Lagrangeana* . Trabalho Apresentado no XVI CNMAC , Friburgo , Rio de Janeiro , 1981.

[64] XAVIER , A. E. - *Penalização Hiperbólica* . Tese de D. Sc. COPPE / UFRJ , Rio de Janeiro , Brasil , 1992 .

[65] XAVIER , A . E. & MACULAN , N. - *Hiperbolic Lagrangean : A New Method of Multiplier* .Trabalho a ser publicado . Rio de Janeiro , Brasil , 1995.

[66] XAVIER , A . E . & MACULAN , N. - *Extrapolação em Penalização Hiperbólica.* II Congresso Latino Americano de Investigação Operativa e Inginieria de Sistemas (Trabalhos selecionados) , Buenos Aires , pp. 24 - 38 , 1984.

[67] ZANGWILL , W. L. - *Nonlinear Programming via Penalty Functions.* Management Science . Vol. 13 , pp. 344 - 358 , 1967.

[68] ZANGWILL , W. L. - *Nonlinear Programming - A Unified Approach* . Prentice Hall , Inc. Englewood Cliffs , New York , 1969.