

## 4 Técnicas de Otimização

Neste capítulo é apresentado o algoritmo de otimização que foi utilizado na solução do problema de otimização definido no Capítulo 3.

Muitos dos métodos de otimização com restrição tem como proposta reescrever o problema de otimização com restrição de forma a obter um novo problema de otimização só que desta vez sem restrição. Quando o problema de otimização com restrição é abordado desta forma, cálculos preliminares ao processo de otimização são sempre necessários, seja para definir a nova função objetivo a ser minimizada sem restrições ou seja para determinar o ponto inicial a otimização.

A técnica utilizada neste trabalho na definição do problema de otimização com restrição, apresentado no Capítulo 3, foi a técnica conhecida como SUMT (*Sequential Unconstrained Minimization Technique*) [10] e [11]. Sua idéia básica consiste em utilizar funções de penalidades para gerar uma seqüência de problemas de otimização sem restrição cujas soluções compreendem uma seqüência de soluções que convergem para a solução do problema original de otimização com restrição.

Desta forma, o problema com restrição cuja a função objetivo é  $f(\tilde{x})$  é substituído por uma seqüência de problemas sem restrições, cujas as funções objetivo são dadas por

$$f_k(\tilde{x}) = f(\tilde{x}) + r_k P(\tilde{x}) \quad (4-1)$$

onde  $r_k$  é um parâmetro positivo que tende para zero a medida em que  $k$  cresce e

$$P(\tilde{x}) = \sum_{i=1}^{N_r} \frac{w_i}{g_i(\tilde{x})} \quad (4-2)$$

Em (4-2),  $N_r$  representa o número de restrições associada ao problema

original,  $w_i = 1, \dots, N_r$  são os fatores que ponderam as restrições (que podem se manter fixos durante os cálculos [11]) e  $g_i(\tilde{x})$  é uma forma que define a  $i$ -ésima restrição retratada sob a forma:

$$g_i(\tilde{x}) \geq 0 \quad (4-3)$$

Note que:

$$\lim_{k \rightarrow \infty} f_k(\tilde{x}) = f(\tilde{x}) \quad (4-4)$$

Assim, a função objetivo do  $k$ -ésimo problema sem restrição se escreve:

$$f_k(\tilde{x}) = f(\tilde{x}) + r_k \sum_i^{N_r} \frac{w_i}{g_i(\tilde{x})}, \quad k = 1, 2, 3, \dots$$

Geometricamente, as funções de penalidades substituem as restrições por barreiras ou vales.

Na solução da seqüência de problemas sem restrição, foi utilizado o método de Newton que requer, a cada passo, o cálculo do Gradiente e da Matriz Hessiana associada a função  $f_k(\tilde{x})$ . Para evitar os erros de aproximação associados a métodos numéricos, o cálculo dessas quantidades foi feito utilizando-se expressões analíticas cujas deduções se encontram no Apêndice B.1.

Na teoria SUMT, o problema inicial é resolvido para  $k = 1$  e a solução encontrada se torna o ponto inicial para o próximo problema com  $k = 2$  e  $r_2 < r_1$ . Esse processo é repetido até que a precisão desejada seja atingida. Os valores de  $w_i$  se mantiveram fixos e iguais a unidade conforme [11]. O valor de  $r_k$  foi iniciado igual a unidade e reduzido de 0.1 a cada iteração até atingir a precisão desejada para as restrições.

Deve-se ressaltar que os pontos por onde passam o algoritmo de otimização devem se manter dentro do domínio de otimização durante todas as iterações. Se um ponto não realizavel for atingido em qualquer instante, então os cálculos continuam com um passo reduzido a partir do último ponto realizavel. As propriedades de convergência deste método são investigadas em [10]. Assim, torna-se importante a análise do processo de otimização, conforme definido na seção a seguir.

## Valor inicial da Otimização

Considere o  $k$ -ésimo problema de otimização sem restrição cuja função objetivo é dada por

$$f(\tilde{x})_k = f(\tilde{x}) + r_k P(\tilde{x})$$

Partindo  $\tilde{x}$  de um ponto inicial  $\tilde{x}_{0,k}$ , o método de Newton conduz a uma solução ótima, utilizando a regra

$$\tilde{x}_{i+1,k} = \tilde{x}_{i,k} + H^{-1}(\tilde{x}_{ik})G(\tilde{x}_{ik})$$

onde  $H$  é a Matriz Hessiana e  $G$  o Vetor Gradiente definido no Apêndice B.1. Utilizando-se do método de Newton determina-se, para o  $k$ -ésimo problema, a solução

$$\tilde{x}_k^*$$

Essa solução se torna ponto inicial para o próximo problema, ou seja

$$\tilde{x}_{0,k+1} = \tilde{x}_k^*$$

Desta forma, para o  $k$ -ésimo problema com ponto inicial em  $\tilde{x}_{0,k}$  tem-se como solução  $\tilde{x}_k^*$ . Para o problema  $k + 1$  com ponto inicial em  $\tilde{x}_{0,k+1}$  tem-se como solução  $\tilde{x}_{k+1}^*$  e assim sucessivamente.

Assim temos a seqüência de soluções:  $\tilde{x}_1^*, \tilde{x}_2^*, \tilde{x}_3^*, \dots$

onde em [10], converge para a solução ótima  $\tilde{x}^*$  do problema original, ou seja,

$$\lim_{k \rightarrow \infty} \tilde{x}_k^* = \tilde{x}^*$$

Assim, torna-se importante a escolha do ponto inicial  $\tilde{x}_{0,1}^*$  de todo o processo.

O valor de  $\tilde{x}_{01}$  foi determinado priorizando os pontos mais a esquerda do arco de serviço de cada sistema que respeite a ordem pré estabelecida e a distância mínima de convivência entre os sistemas de forma a garantir o nível de interferência de entrada única e agregada abaixo do limiar especificado.

Conforme comentado na seção 3.4 a pré-fixação da ordem dos sistemas reduz a complexidade do problema em análise. Para se obter a solução fim, deve-se considerar todas as possíveis ordenações, comparando-se as soluções a elas associadas. Quando o número de satélites envolvidos é  $n$ , o número de ordenações possíveis é  $n!$ . Isso torna computacionalmente árduo gerar todas as ordenações. No entanto, a maioria dessas ordens não satisfazem as restrições impostas pelos arcos de serviço dos sistemas. Assim, foi desenvolvido um algoritmo, utilizando uma estrutura em árvore que permite gerar apenas as ordens que respeitam as restrições de arco de serviço. Este algoritmo é apresentado no Apêndice C.

Dado  $n$  sistemas e utilizando o algoritmo descrito no Apêndice C, o número de ordenações que respeitam a restrição de arco de serviço é menor do que  $n!$ . Mesmo assim a quantidade de possíveis ordens à se otimizar é grande. Por este motivo define-se um critério, baseado na Matriz de Espaçamento Orbital, para identificar para cada ordenação, antes de aplicar o algoritmo de otimização, aquelas que não fornecerão um valor de função objetivo menor do que o valor obtido anteriormente com uma ordem qualquer e podem ser excluídas. Este critério é apresentado abaixo.

### **Critério para eliminação de ordens a priori do algoritmo de otimização**

Seja a Matriz de Espaçamento Orbital Mínima ( $\Theta$ ) onde seus elementos correspondem a separação orbital mínima  $\theta_{ij}$  entre os satélites  $i$  e  $j$  que garante que as restrições de entrada única dos dois sistemas sejam satisfeitas. Ou seja,

$$\theta_{ij} = \max(\theta_{i \leftarrow j}, \theta_{j \leftarrow i})$$

onde  $\theta_{i \leftarrow j}$  é a separação orbital mínima entre os satélites  $i$  e  $j$  para que a interferência de entrada única que o sistema  $j$  provoca no sistema  $i$  esteja acima do limiar especificado.

Desta forma, define-se o seguinte sequência de parâmetros para cada ordenação analisada:

$$\begin{aligned}d_k &= \max(\theta_{jk} + d_j) \\k &= 2, \dots, n \\j &= 1, \dots, k - 1\end{aligned}$$

onde

$$d_1 = 0$$

e  $k$  é o número de sistemas em cada iteração. Assim, eliminam-se todas as ordenações nas quais

$$d_n > f(\tilde{x}^*)$$

onde  $\tilde{x}^*$  corresponde a "melhor ordenação" dentre aquelas já testadas anteriormente.

Dada a uma ordenação qualquer, com  $k = n$  ( $n$  sistemas considerados), e  $d_n < f(\tilde{x}^*)$  nada pode-se afirmar sobre o resultado do processo de otimização desta ordem, uma vez que o parâmetro  $d_n$  apenas considera a restrição de entrada única.