

4

Métodos de Solução dos Problemas Apresentados

Os sete problemas delineados na seção anterior serão individualmente discutidos de forma matematicamente rigorosa, sendo ressaltados os aspectos consagrados na literatura e os aspectos inovadores frutos da pesquisa aqui realizada.

Para fins de formalização, define-se o conceito de enlace.

Definição 1

Um *enlace* é uma conexão ponto a ponto entre duas unidades de rádio caracterizado por um elemento transmissor (Tx) e um elemento receptor (Rx). Seja \mathcal{L} uma coleção de L enlaces. Se estes utilizam o mesmo recurso de comunicação (canal, unidade de tempo, código, etc.) ocorre uma interferência mútua entre as unidades de rádio.

Formalmente a qualidade de serviço básica ao longo de todos os problemas aqui discutidos consiste em garantir que a SINR experimentada pelo enlace i deve obedecer a:

$$SINR_i = \frac{G_{ii} \cdot P_i}{n_i + \sum_{\substack{i=1 \\ j \neq i}}^L G_{ij} \cdot P_j} \geq \gamma_i \quad (4.1)$$

onde:

- G_{ij} é o “ganho de enlace” (número adimensional no intervalo $[0,1]$) entre o transmissor do enlace j e o receptor do enlace i .
- P_i é a potência do transmissor do enlace i .

- n_i é a potência média de ruído térmico percebido pelo receptor do enlace i .
- γ_i é a SINR mínima a ser atendida pelo receptor do enlace i .

Conforme foi discutido na seção anterior, a desigualdade acima quando estendida para todos os enlaces pode ser vista na forma matricial:

$$(\mathbf{I} - \mathbf{H}) \cdot \underline{P} \geq \underline{\eta} \quad (4.2)$$

Onde:

- $\mathbf{I} - \mathbf{H}$ é uma matriz quadrada de diagonal nula e elementos não-negativos e limitados em 1 fora da diagonal.
- $\underline{\eta}$ é um vetor não-negativo.
- \underline{P} é o vetor de potências dos transmissores dos enlaces, que por razões físicas, tem que ser positivo.

Para fins de simplicidade será assumido que a matriz $(\mathbf{I} - \mathbf{H})$ é não-singular, o que não causa uma perda de generalidade. Se esta matriz for singular, é possível perturbar \mathbf{H} de maneira arbitrariamente pequena de forma a eliminar a condição de singularidade. Isto se faz alterando minimamente qualquer um dos elementos da matriz, o que já elimina a singularidade.

4.1.

O Problema da Viabilidade de Compartilhamento de Canais

O problema a ser discutido nesta sub-seção consiste em determinar as condições que a matriz \mathbf{H} deve satisfazer de modo que o conjunto Ω definido

como o conjunto dos pontos \underline{P} pertencentes a \mathbb{R}^L que satisfaz (4.2) tenha interseção com \mathbb{R}_+^L .

Antes de responder esta pergunta, se faz necessário apresentar algumas propriedades do conjunto Ω acima definido.

- É um conjunto convexo ilimitado (cone) com um único ponto extremo dado por:

$$\underline{P}^* = (1 - \mathbf{H})^{-1} \eta \quad (4.3)$$

- Este ponto extremo \underline{P}^* tem todas as suas componentes não-negativas ou não-positivas.

- Se este ponto extremo for não-positivo (não-negativo) então Ω é formado apenas de pontos não-positivos (não-negativos). Além disso, nenhum ponto de Ω tem em módulo coordenadas de menor potência do que este ponto extremo.

Estas propriedades podem ser visualizadas no exemplo abaixo, onde se particulariza o caso de dimensão L igual a 2.

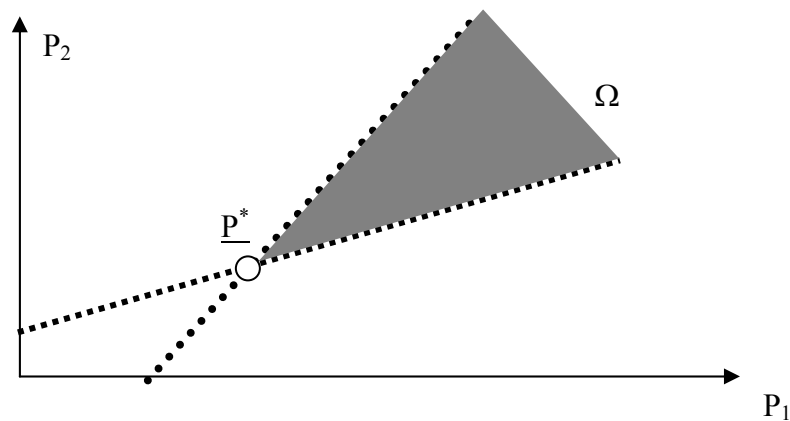


Figura 4-1 – Região de solução f-viável

Em uma abordagem formal, definem-se as soluções que interessam da forma:

Definição 2:

Um conjunto de enlaces é dito *fundamentalmente viável* (f-viável) quando o ponto extremo associado à matriz \mathbf{H} correspondente atende à seguinte relação:

$$P^* = (1 - \mathbf{H})^{-1} \eta \geq \underline{0} \quad (4.4)$$

Um resultado existente na literatura e bastante conhecido, diz respeito à condição necessária e suficiente para um enlace seja f-viável. Este resultado é descrito pelo seguinte teorema cuja demonstração pode ser encontrada em [71]:

Teorema 1:

Um conjunto de enlaces é f-viável quando o raio espectral (definido como o maior autovalor de uma matriz) da matriz \mathbf{H} for inferior a 1.

O cálculo do raio espectral de uma matriz pode ser feito por algoritmos de complexidade da ordem de n^3 , onde n é a dimensão da matriz \mathbf{H} . Esta complexidade é elevada o suficiente de modo a tornar inviável o problema da determinação de todos os subconjuntos f-viáveis. Estes formados por todas as submatrizes (matriz em que uma ou mais colunas e linhas são retiradas) possíveis de \mathbf{H} .

O cálculo do raio espectral de uma matriz implica a obtenção de todos os seus autovalores, mas existem algumas técnicas que podem determinar se seu raio espectral é inferior a 1 ou não com esforço reduzido. Por exemplo, podemos obter um limitante superior para seu valor que é simplesmente a norma da matriz:

$$\rho(A) \leq \|A\| \quad (4.5)$$

Este limitante é fácil de se calcular, mas é uma medida de pouca utilidade, só interessando quando a norma for menor que 1. Caso contrário não acrescenta informação nenhuma, pois nada garante que caso uma matriz tenha norma próxima de 1, esta tenha mais probabilidade de ser f-viável do que uma matriz de norma com valor elevado. Uma opção alternativa com resultados mais confiáveis consiste na estimativa do raio espectral usando o conjunto de círculos de Gerschgorin [71] como descrito abaixo:

Os autovalores de uma matriz $\mathbf{A} \in \mathbb{C}^{n \times n}$ estão contidos na união \mathcal{G}_r dos n círculos de Gerschgorin definidos por:

$$|z - a_{ii}| \leq r_i, \quad \text{onde} \quad r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{para} \quad i = 1, 2, \dots, n \quad (4.6)$$

Em outras palavras, os autovalores estão dentro de uma coleção de círculos centrados em a_{ii} (que no caso presente são todos nulos, pois \mathbf{H} tem diagonal nula) com o raio dado pela soma dos valores absolutos em \mathbf{A}_{i*} com a_{ii} retirado. De uma maneira mais simples, o raio de cada círculo é a soma de todos os elementos de uma coluna sem o elemento da diagonal principal (é possível provar que o mesmo vale para a soma de todos os elementos das linhas), e o centro está localizado no valor do elemento da diagonal principal. Além do mais, se a união \mathcal{U} de k círculos de Gerschgorin não se encontra com qualquer dos outros $n-k$ círculos, então existem exatamente k autovalores nos círculos em \mathcal{U} . Esta propriedade, embora interessante, não é útil na situação presente uma vez que todos os discos de Gerschgorin têm o mesmo centro localizado na origem.

A aplicação destes resultados ao caso particular da matriz \mathbf{H} aqui em estudo revela que se a desigualdade abaixo for atendida, o seu raio espectral encontra-se dentro do disco unitário. Entretanto nada se pode dizer quando esta condição não é atendida. Note ainda que o esforço computacional associado a esta expressão é tão baixo (máximo de n somas de $n-1$ parcelas), que mesmo no caso do resultado

não ser útil, o tempo consumido em seu cálculo é desprezível em relação aos métodos exatos que se seguem, valendo a pena como uma tentativa.

$$\max_{i=1,2,\dots,n} \sum_{j=1}^n H_{ij} < 1 \quad (4.7)$$

A combinação dos dois métodos acima permite a obtenção razoavelmente rápida de uma estimativa do raio espectral de uma matriz. Entretanto nas situações onde esta estimativa for inútil, ou seja os valores obtidos forem maiores que os limitantes, tem-se que reportar ao cálculo efetivo do raio espectral.

O trabalho de pesquisa desenvolvido durante a elaboração desta tese permitiu determinar uma condição equivalente de f-viabilidade ainda não reportada na literatura que envolve apenas o cálculo de determinantes. Esta condição pode ser resumida no teorema 2 cuja demonstração é apresentada na sub-seção seguinte.

Teorema 2:

Um conjunto de enlaces é f-viável quando qualquer sub-matriz quadrada de **(I-H)**, cujos índices de linha e coluna são idênticos, tem determinante não-negativo.

Se o interesse for em determinar unicamente a f-viabilidade de um particular conjunto de L enlaces, certamente a verificação pelo Teorema 1 requer um esforço computacional menor do que aquele produzido pelo Teorema 2. Embora o esforço computacional do cálculo do determinante seja da ordem de n^2 , em relação a este último teorema serão necessários o cálculo de C_L^1 determinantes de dimensão 1, C_L^2 determinantes de dimensão 2, e assim em diante até C_L^L determinantes de dimensão L, onde C_L^N é combinação de L elementos formando grupos de N

elementos (análise combinatória). E este esforço se verificou superior ao do cálculo do raio espectral.

Entretanto, será visto a seguir, que se o problema for determinar todos os subconjuntos f -viáveis de um conjunto predefinido, o Teorema 2 será extremamente conveniente.

4.1.1. Demonstração do Teorema 2

Para provar o teorema é preciso determinar quais condições são necessárias para forçar um ponto extremo único a ser positivo. Esta prova será feita por indução. Para começar será mostrado que é verdade para $K=2$.

$$\underline{x}^* = \mathbf{A}^{-1} \underline{b} = \begin{pmatrix} a & -c \\ -d & b \end{pmatrix}^{-1} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{a.b - c.d} \cdot \begin{bmatrix} b.u + c.v \\ d.u + a.v \end{bmatrix} \quad (4.8)$$

Conclui-se que $\underline{x} > 0$ só e somente se:

$$\det(\mathbf{A}) = a.b - c.d > 0 \quad (4.9)$$

Agora será mostrado que se a propriedade mencionada anteriormente é válida para todo $k \leq M < N$, então é válido para $k = M + 1$. Para mostrar isso, seja B um elemento genérico de $\mathbf{Z}_{M+1}(\mathbf{A})$, então

$$[\mathbf{B}^{-1}]_{ij} = (-1)^{i+j} \cdot \frac{M_{ji}(\mathbf{B})}{\det(\mathbf{B})} \quad (4.10)$$

Quando a linha j e a coluna i de \mathbf{B} são eliminadas, se $i > j$ então a linha $i - 1$ e a coluna j não contêm elementos negativos. Do mesmo modo se $i < j$ então a linha i e a coluna $j - 1$ não contêm elementos negativos. Assim algumas trocas

apropriadas de linhas e colunas podem transformar esses *menores* em matrizes com as propriedades desejadas. Define-se:

- $T_i(\mathbf{B})$ a matriz idêntica a \mathbf{B} com trocas cíclicas das suas primeiras i linhas.
- $T^j(\mathbf{B})$ a matriz idêntica a \mathbf{B} com trocas cíclicas das suas primeiras j colunas.
- $T_i^j(\mathbf{B}) = T_i [T^j(\mathbf{B})]$.

Então pode facilmente ser mostrado que:

$$[\mathbf{B}^{-1}]_{ij} = \frac{1}{\det(\mathbf{B})} \cdot \begin{cases} \det(M_{ii}(\mathbf{B})) & sei = j \\ -\det(T_{i-1}^j [M_{ji}(\mathbf{B})]) & sei > j \\ -\det(T_i^{j-1} [M_{ji}(\mathbf{B})]) & sei < j \end{cases} \quad (4.11)$$

Se $i = j$ então $M_{ii}(\mathbf{B})$ pertence a $\mathbf{Z}_M(\mathbf{A})$ cujo sinal é conhecido. Quando $i \neq j$, a matriz que aparece no numerador é um elemento de \mathbf{F}_M^1 . Chamando esta matriz por \mathbf{C} e expandindo seu determinante com respeito a sua primeira coluna, obtêm-se:

$$\det(\mathbf{C}) = \sum_{i=1}^{M-1} (-1)^{i+1} \cdot c_{i1} \cdot \det(M_{i1}(\mathbf{C})) \quad (4.12)$$

Novamente, levando em consideração que quando a linha $i > 1$ e coluna 1 são eliminados de \mathbf{C} , a coluna $(i-1)$ da matriz resultante não possui elementos negativos. Assim $(i-2)$ trocas convenientes de coluna transformam esta matriz em uma de \mathbf{F}_M^1 . Deste modo a expressão acima pode ser reescrita como:

$$\det(\mathbf{C}) = c_{11} \cdot \det(M_{11}(\mathbf{C})) - \sum_{i=2}^{M-1} c_{i1} \cdot \det(T^{i-1} [M_{i1}(\mathbf{C})]) \quad (4.13)$$

Onde todos os coeficientes são não negativos. Se for definido $D_i = T^{i-1} [M_{i1}(\mathbf{C})]$, e se for expandido os determinantes dessas matrizes em volta de sua primeira coluna, e se for feita uma troca apropriada de colunas, chega-se a:

$$\det(\mathbf{C}) = c_{11} \cdot \det(M_{11}(\mathbf{C})) - \sum_{i=2}^{M-1} c_{i1} \cdot [[D_i]_{11} \cdot \det[M_{11}(D_i)] - \Lambda] \quad (4.14)$$

Onde:

$$\Lambda = \sum_{j=3}^{M-1} [D_i]_{j1} \cdot \det[T^{j-1}[M_{j1}(D_i)]] \quad (4.15)$$

Se o procedimento acima for repetido até o ponto que se encontre matrizes de dimensão 2, têm-se a expressão seguinte:

$$\begin{aligned} \det(\mathbf{C}) = & \lambda^1 \cdot \det(\mathbf{W}^1) + \\ & - \sum_{i=1}^{M-1} \lambda_i^2 \det(\mathbf{W}_i^2) + \\ & + \sum_{i=2}^{M-1} \sum_{j=3}^{M-1} \lambda_{ij}^3 \det(\mathbf{W}_{ij}^3) - \dots \end{aligned} \quad (4.16)$$

Onde $\mathbf{W}_{ij}^k \in \mathbf{Z}_{M-i}(\mathbf{A})$ e conseqüentemente eles apresentam sinais alternados. Conseqüentemente $-\det(\mathbf{C})$ possui o mesmo sinal de qualquer elemento de $\mathbf{Z}_M(\mathbf{A})$. Assim todos os elementos de \mathbf{B}^{-1} têm o sinal governado pelo $\det(\mathbf{B})$ e para gerar um ponto extremo positivo, $\det(\mathbf{B})$ precisa ter sinal oposto a qualquer elemento de $\mathbf{Z}_M(\mathbf{A})$, provando assim o teorema.

4.2.

O Problema da Determinação dos Subconjuntos f-viáveis

Primeiro vão ser determinados todos os subconjuntos f-viáveis de um conjunto \mathcal{L} de L enlaces. Este procedimento consome um razoável tempo de processamento (é bom lembrar que está se procurando um método mais rápido que o cálculo do raio espectral), mas a determinação destes subconjuntos será de extrema importância nos problemas que se seguirão, e nos algoritmos que serão apresentados.

Duas observações que se podem concluir sobre a estrutura de subconjuntos f-viáveis são:

- Se A é um subconjunto f-viável, qualquer subconjunto de A também é f-viável.
- Se A não é um subconjunto f-viável, qualquer subconjunto que contenha A também não é f-viável.

Assim um procedimento intuitivo mas eficiente, baseado nas observações acima, para a determinação de subconjuntos f-viáveis consiste em inicialmente descobrir quais os pares de enlaces f-viáveis. De posse desta lista, determina-se as trincas de enlaces f-viáveis, não verificando todas as trincas possíveis, mas tentando acrescentar um enlace somente às duplas f-viáveis. Intuitivamente este procedimento é o mais eficiente possível, pois só as duplas que possuem real chance de se tornarem trincas f-viáveis são investigadas. Após a geração de todas as trincas f-viáveis, este procedimento pode ser repetido para a geração das quádruplas f-viáveis e assim por diante.

Este procedimento pode ser formalizado por meio do seguinte algoritmo

Algoritmo de f-viabilidade

Seja $L_k = \{\underline{w}_1^k, \underline{w}_2^k, \dots, \underline{w}_{N_k}^k\}$ o conjunto de todos os subconjuntos f-viáveis de dimensão k de $L = \{1, 2, \dots, N\}$ e denotemos por \underline{w}_i^k como o vetor $\{w_{i1}^k, w_{i2}^k, \dots, w_{ik}^k\}$ onde $w_{i1}^k < w_{i2}^k < \dots < w_{ik}^k$.

```

- Seja  $L_{k+1} = \phi$ 
- para  $i = 1$  a  $N_k$ 
  ○ para  $j = w_{ik}^k + 1$  a  $N$ 
    ■ faça  $comp = TRUE$ 
    ■ para  $n = 1$  a  $k$ 
      ● se  $\underline{w}_i^k - \{w_{in}^k\} + \{j\} \notin L_k$  então
        ○  $comp = FALSE$ 
        ○ break
      ● end
    ■ end
  ○ end
  ○ if  $comp = TRUE$  e FEASIBLE( $\underline{w}_i^k + \{j\}$ ) = TRUE
    então INSERT ( $\underline{w}_i^k + \{j\}$ ) em  $L_{k+1}$ 
- end

```

Repare que no algoritmo acima o procedimento FEASIBLE pode ser implementado tanto por cálculo de raio espectral como por meio de determinantes, sendo que agora o último tem vantagens sobre o primeiro em termos de esforço computacional.

Além desta vantagem existe uma outra que reduz mais ainda o esforço computacional e que será a seguir discutida.

Considere que um certo subconjunto de enlaces S_n de dimensão n seja considerado f -viável. Em algum momento na determinação dos subconjuntos f -viáveis de dimensão $n+1$, terá que se acrescentar a S_n um dos $(L-n)$ enlaces pertencentes ao complemento de S_n , gerando um subconjunto S_{n+1} e verificar a sua f -viabilidade. A solução imediata para este problema consiste em obter o determinante da matriz H_{n+1} associada ao subconjunto S_{n+1} e verificar o seu sinal.

Será mostrado aqui um algoritmo iterativo que a partir de determinantes de ordem inferior, pode-se calcular os de ordem superior com uma notável economia de esforço computacional. Na medida em que S_n e S_{n+1} diferem apenas em um único elemento, a matriz H_{n+1} é formada pela matriz H_n onde se acrescenta uma linha e uma coluna como abaixo:

$$H_{n+1} = \begin{bmatrix} H_n & \underline{a}_n \\ \underline{b}_n^T & c_n \end{bmatrix} \quad (4.17)$$

Denominando de Δ_n o determinante de H_n , sabe-se que pela identidade do determinante de Jacobi, a seguinte equação é verdadeira:

$$\Delta_{n+1} = \Delta_n \cdot (c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n) \quad (4.18)$$

Assim como S_n é f-viável (o que implica em $\Delta_n > 0$), para que S_{n+1} seja f-viável é necessário e suficiente que $c_n > \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n$, este cálculo têm esforço computacional baixo se a matriz inversa de H_n for conhecida.

Um subproduto importante deste resultado é que a verificação de todos os casos de subconjuntos S_{n+1} a partir do mesmo subconjunto S_n requer apenas uma única inversão de matriz, o que reduz o esforço computacional de todo o processo.

Resta ainda o problema do cálculo da matriz inversa de H_n . A idéia aqui é mostrar que este cálculo pode ser feito de forma iterativa com baixo esforço computacional. Para tal vamos assumir que L_n e U_n são as matrizes da decomposição LU (decomposição em duas matrizes, uma só com elementos abaixo da diagonal principal, e a outro só com elementos acima desta) da matriz H_n e que esta decomposição é conhecida. Logo o cálculo da matriz inversa de H_n pode ser trivialmente feito lembrando que:

$$H_n = L_n \cdot U_n \Leftrightarrow H_n^{-1} = U_n^{-1} \cdot L_n^{-1} \quad (4.19)$$

Aparentemente não há vantagens aqui porque se assumiu que a decomposição LU é conhecida e que também são conhecidas às inversas das matrizes que constituem esta decomposição.

Entretanto as vantagens surgem quando se tenta repetir este processo para os subconjuntos de dimensão $n+2$. Neste caso precisa-se conhecer a decomposição LU de H_{n+1} assim com as inversas das matrizes que constituem esta decomposição.

Um resultado publicado permite afirmar que:

$$H_{n+1} = \begin{bmatrix} H_n & \underline{a}_n \\ \underline{b}_n^T & c_n \end{bmatrix} = \begin{bmatrix} L_n & \underline{0} \\ \underline{b}_n^T \cdot U_n^{-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} U_n & L_n^{-1} \cdot \underline{a}_n \\ \underline{0}^T & c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n \end{bmatrix} = L_{n+1} \cdot U_{n+1} \quad (4.20)$$

Revelando que esta decomposição LU pode ser facilmente calculada a partir das matrizes L_n e U_n e suas inversas. Note que neste estágio, todas estas matrizes são supostas conhecidas. Entretanto resta ainda a necessidade de determinar as inversas de L_{n+1} e U_{n+1} , caso contrário a iteratividade mencionada não é alcançada. Através de procedimentos algébricos bastante simples pode-se mostrar que estas inversas são expressas por:

$$L_{n+1}^{-1} = \begin{bmatrix} L_n & \underline{0} \\ \underline{b}_n^T \cdot U_n^{-1} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} L_n^{-1} & \underline{0} \\ -\underline{b}_n^T \cdot U_n^{-1} \cdot L_n^{-1} & 1 \end{bmatrix} = \begin{bmatrix} L_n^{-1} & \underline{0} \\ -\underline{b}_n^T \cdot H_n^{-1} & 1 \end{bmatrix} \quad (4.21)$$

$$U_{n+1}^{-1} = \begin{bmatrix} U_n & L_n^{-1} \cdot \underline{a}_n \\ \underline{0}^T & c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n \end{bmatrix}^{-1} = \begin{bmatrix} U_n^{-1} & -H_n^{-1} \cdot \underline{a}_n / (c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n) \\ \underline{0}^T & 1 / (c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n) \end{bmatrix} \quad (4.22)$$

onde todos os termos existentes são conhecidos.

Expressaremos o esforço computacional pela 4-upla $[q_+, q_-, q^*, q]$ que define respectivamente o número de adições, subtrações, multiplicações e divisões

envolvidas na operação em questão. Estamos assumindo que conhecemos as matrizes U_n , L_n e suas inversas e queremos determinar U_{n+1} , L_{n+1} e suas inversas. A tabela abaixo apresenta os valores destas grandezas para os cálculos indicados na primeira coluna.

Cálculo de	e_+	e_-	e_*	$e_/'$
$\underline{b}_n^T \cdot U_n^{-1} \text{ e } L_n^{-1} \cdot \underline{a}_n$	$n(n-1)$	0	n^2	0
$c_n - \underline{b}_n^T \cdot H_n^{-1} \cdot \underline{a}_n$	n	1	$n-1$	0
Δ_{n+1}	0	0	1	0
$U_{n+1}^{-1} \text{ e } L_{n+1}^{-1}$	$n(n-1)/2$	0	$n^2/2$	0
TOTAL	$n \cdot [1+4 \cdot (n-1)]$	1	$n \cdot (4n+1)$	0
DETERMINANTE	$\frac{n \cdot (n+1) \cdot (n+2)}{3}$	0	$\frac{n \cdot (n+1) \cdot (n+2)}{3} + (n+2)$	n

Tabela 3 – Tabela de esforço computacional

O cálculo do esforço computacional do determinante é determinado por meio de uma seqüência de operações elementares sobre linhas da matriz para torná-la triangular. O determinante é então calculado pelo produto dos termos da diagonal.

Se admitirmos como equivalentes os pares (soma, subtração) e (multiplicação, divisão) e que os segundos demandam um esforço computacional para seu cálculo α vezes maior que os primeiros, então a razão entre os esforços computacionais para o cálculo dos determinantes de Δ_{n+1} e Δ_n é dada por:

$$\rho = \frac{n+1}{3} \cdot \frac{(1+\alpha) \cdot n \cdot (n+2) + 6}{3 \cdot (1+\alpha) \cdot n^2 + (1-2\alpha) \cdot n + \alpha} \quad (4.23)$$

A figura abaixo ilustra o gráfico desta função:

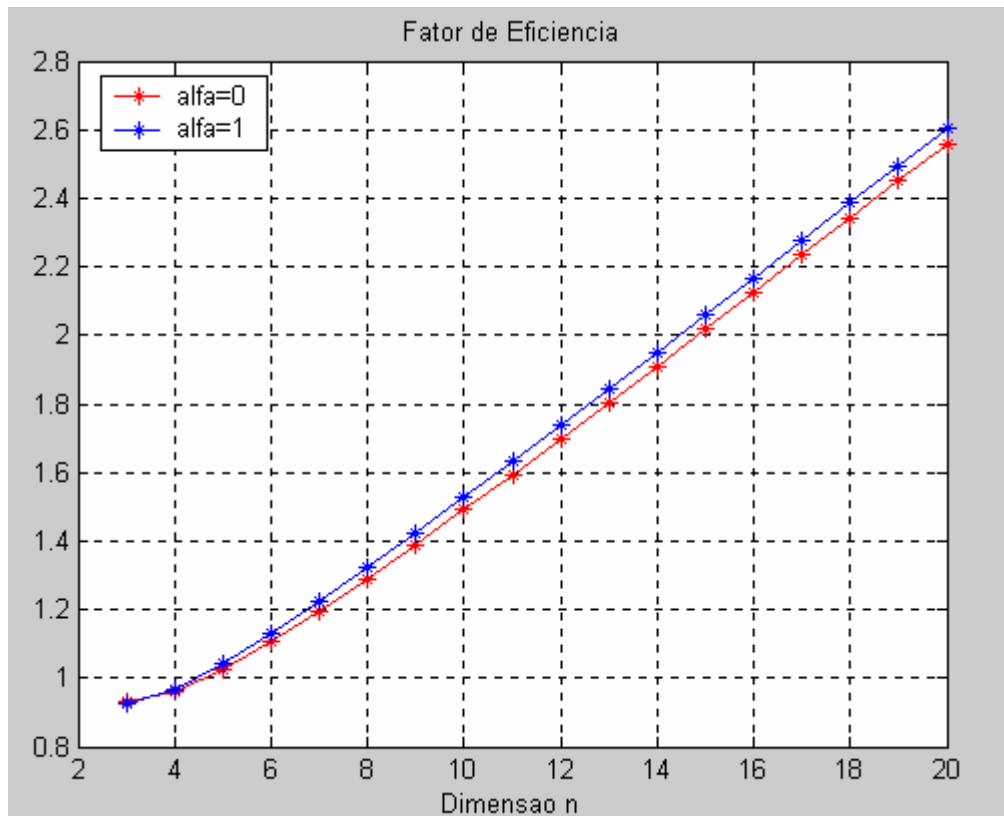


Figura 4-2 – Comparação de esforços computacionais

Fica claro que, independente do peso entre as operações equivalentes, a partir da dimensão $n=5$, este método é vantajoso quando comparado com o cálculo direto do determinante.

4.3. O problema da limitação de potência

Mesmo nos casos em que um subconjunto de enlaces é f-viável, isso não quer dizer que ele é satisfatório sob o ponto de vista físico. A potência do transmissor de cada enlace tem que se situar numa faixa factível, que aqui será denominado de $[P_{\min}, P_{\max}]$. Agora está se procurando valores para as potências que pertençam ao conjunto $\Omega^* = \Omega \cap \Gamma$ onde $\Gamma = \{x \in \mathcal{R}^N \text{ tais que } P_{\min} \leq x_i \leq P_{\max} \text{ para todo } i\}$. A seguinte definição descreve o tipo de subconjunto de enlaces que serão agora analisados.

Definição 3:

Um conjunto de enlaces f -viável é dito *estruturalmente viável* (e-viável) quando atende também as restrições de potência acima mencionadas.

Para saber se esta interseção $\Omega^* = \Omega \cap \Gamma$ é vazia ou não, pode-se formular um problema semelhante à Fase 1 do algoritmo Simplex. As desigualdades:

$$\begin{aligned} (I - H) \cdot \underline{P} &\geq \underline{\eta} \\ \underline{P} &\geq \underline{P}_m \\ \underline{P} &\leq \underline{P}_M \end{aligned} \quad (4.24)$$

Transformam-se por inclusão de variáveis de folga não negativas \underline{u} , \underline{v} , e \underline{w} :

$$\begin{aligned} (I - H) \cdot \underline{P} - \underline{u} &= \underline{\eta} \\ \underline{P} - \underline{v} &= \underline{P}_m \\ \underline{P} + \underline{w} &= \underline{P}_M \end{aligned} \quad (4.25)$$

Infelizmente não é possível de forma fácil obter uma solução básica factível para este conjunto. Entretanto se forem introduzidas as variáveis artificiais não negativas \underline{r} e \underline{s} , tem-se:

$$\begin{aligned} (I - H) \cdot \underline{P} - \underline{u} + \underline{r} &= \underline{\eta} \\ \underline{P} - \underline{v} + \underline{s} &= \underline{P}_m \\ \underline{P} + \underline{w} &= \underline{P}_M \end{aligned} \quad (4.26)$$

e uma solução factível básica que pode ser imediatamente identificada é expressa por:

$$\begin{aligned} \underline{P} = \underline{u} = \underline{v} &= \underline{0} \\ \underline{r} &= \underline{\eta} \\ \underline{s} &= \underline{P}_m \\ \underline{w} &= \underline{P}_M \end{aligned} \quad (4.27)$$

Se agora for definido um problema de otimização onde a função objetivo a ser minimizada é expressa por:

$$z = \sum_{i=1}^N r_i + s_i \quad (4.28)$$

Tem-se que Ω^* é não vazio se e somente se o valor ótimo de z for zero.

Conclui-se então que se a solução do problema de otimização descrito for zero, então o subconjunto f -viável de enlaces também é e -viável.

Embora existam pacotes de software (“solvers”) comerciais e de domínio público que resolvem este problema (o MATLAB é a opção mais simples), convém salientar que a confecção de um programa que resolva o problema aqui discutido é de complexidade moderada.

O mesmo algoritmo discutido em seção anterior para determinação de todos os subconjuntos f -viáveis pode ser modificado para a geração de todos os subconjuntos e -viáveis. É suficiente que o procedimento FEASIBLE também já discutido inclua o problema de programação linear discutido nesta seção.

4.4. Problemas Baseados em Seleção de Subconjuntos Viáveis

De posse de métodos para a determinação de todos os subconjuntos e -viáveis de um conjunto de enlaces, é possível partir para se resolver problemas ligados à partição do conjunto de enlaces de modo a obter uma partição ótima segundo algum critério.

O contexto descrito nas seções anteriores permite pelo menos definir dois problemas de otimização, cuja solução é de interesse para os projetistas de sistemas de redes móveis sem fio. São eles, a saber:

- Conseguir uma alocação com número mínimo de canais
- Conseguir uma alocação com mínima potência máxima

Antes de discutir em detalhes cada um destes problemas, seria conveniente definir a base comum em que ambos se baseiam, na medida em que se busca partições de um particular conjunto e por definição, os elementos que a constituem dever ser disjuntos e quando unidos constituir o todo.

Considere, como em situações anteriores, um conjunto de enlaces $\mathcal{L} = \{1, 2, 3, \dots, L\}$ e que por métodos descritos em seções anteriores foi possível determinar a lista de cardinalidade M de todos os subconjuntos e-viáveis de \mathcal{L} da forma $\Pi = \{\Pi_1, \Pi_2, \dots, \Pi_M\}$

Uma partição de \mathcal{L} formada por elementos de Π , se existir pode ser representada por um vetor binário $\underline{x} = \{x_1, x_2, \dots, x_M\}$ onde x_i assume o valor 1 se e somente se Π_i pertence à partição escolhida, sendo 0 caso contrário.

Para caracterizar o envolvimento dos enlaces nos elementos da partição, constrói-se uma matriz binária A de dimensão $L \times M$ de modo que o elemento a_{ij} assume o valor 1 se e somente se o enlace i faz parte do subconjunto viável Π_j .

Para que \underline{x} descreva de fato uma partição é condição necessária e suficiente que a seguinte relação linear seja satisfeita:

$$A \cdot \underline{x} = \underline{1} \quad (4.29)$$

onde $\underline{1}$ é um vetor de dimensão M onde todas as suas componentes são unitárias.

4.4.1. O Problema do Número Mínimo de Canais

Usando a vetor \underline{x} descrito na sub-seção anterior, o problema denominado de número mínimo de canais consiste em minimizar a função objetivo abaixo:

$$z = \sum_{i=1}^M x_i \quad (4.30)$$

sujeito à restrição imposta por (4.29). Trata-se de um problema de programação linear-(0,1) para o qual existem vários programas de computador (“solvers”) capazes de resolvê-lo. Entretanto situações reais podem revelar casos onde o valor de M atinge algumas centenas o que torna a obtenção de sua solução bastante custosa em termos de esforço computacional. Esta limitação naturalmente sugere a busca de procedimentos heurísticos que reduzam este esforço sem comprometer em demasia o valor da solução encontrada. Algumas seções posteriores se dedicarão a fornecer estas soluções heurísticas e uma análise de sua eficiência também será apresentada.

4.4.2 O Problema de Mínima Potência Máxima

Dada a natureza binária da variável de decisão do problema anterior, é bastante razoável que a solução do problema de número mínimo de canais não seja única. Se L^* é a cardinalidade da partição ótima deste problema, faz sentido e é desejável conhecer qual destas soluções apresenta a menor potência máxima, ou seja, a potência usada no enlace com maior potência na partição é mínima. Este problema se resolvido, não só gera a alocação com o menor número de “canais” como também fornece a solução mais econômica em termos de potência.

Entretanto, antes que possamos formular este problema precisamos resolver um outro que consiste em determinar para cada subconjunto e-viável de enlaces da lista Π , os menores valores de potência que satisfazem a condição de e-viabilidade. Considere que o elemento Π_q de Π seja da forma $\{i_1, i_2, \dots, i_k\}$ representando enlaces que operarão com potências $\{P_1, P_2, \dots, P_k\}$. Se H e η representarem respectivamente a matriz de ganhos e o vetor de ruídos normalizados associados aos enlaces de Π_q , então estas condições em conjunto com a e-viabilidade permitem formular o problema de interesse como:

$$\begin{aligned} \min z &= \max \{P_1, P_2, \dots, P_k\} \\ \text{sujeito a :} \\ (I - H) \cdot \underline{P} &\geq \underline{\eta} \\ \underline{P} &\geq \underline{P}_m \\ \underline{P} &\leq \underline{P}_M \end{aligned} \quad (4.31)$$

Este problema é classificado na literatura de otimização como do tipo min-max com restrições lineares. Um programa bastante adequado para resolvê-lo encontra-se no pacote de computador MATLAB.

Para fins de entendimento, apresentamos a seguir uma ilustração gráfica para este problema no caso específico de dimensão dois:

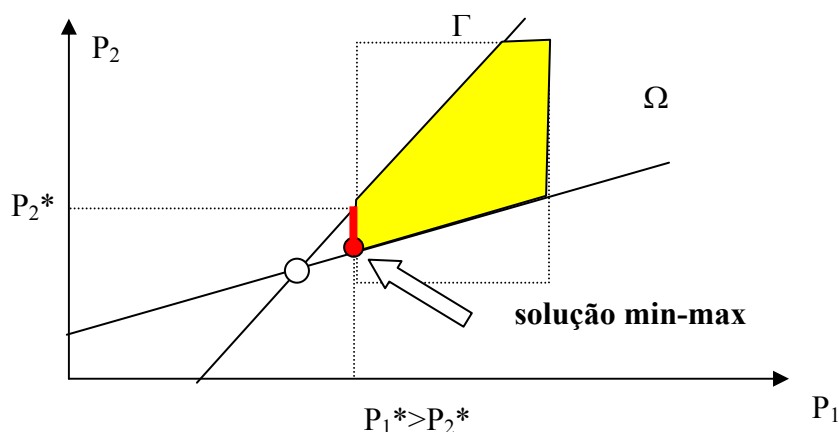


Figura 4-3 – Min-max de dimensão 2

Neste exemplo, a solução mini-max (minimização do valor máximo de uma função) não é única (todos os pontos da linha grossa vermelha) mas o valor da função objeto é o mesmo para todos eles. Quando a solução não é única, pode-se definir uma seqüência de problemas min-max nas variáveis restantes de modo a achar a solução mais adequada.

Uma vez conhecido, para cada subconjunto e-viável Π_i , o seu valor de potência P min-max, aqui denotado por \tilde{P}_i , podemos descobrir qual a partição com menor cardinalidade que minimiza o maior valor \tilde{P}_i utilizado. Novamente recaímos num problema min-max com restrições lineares binárias com a seguinte formulação:

$$\begin{aligned} \min z &= \max \{ \tilde{P}_1 \cdot x_1, \tilde{P}_2 \cdot x_2, \dots, \tilde{P}_M \cdot x_M \} \\ \text{sujeito a :} \\ A \cdot \underline{x} &= \underline{1} \\ \sum_{i=1}^M x_i &= L^* \\ \underline{x} &\in \{0, 1\}^M \end{aligned} \quad (4.32)$$

Onde \underline{x} tem o mesmo significado anteriormente descrito e L^* é o valor da função objetivo na solução ótima do número mínimo de canais.

Convém lembrar que, nos casos em que o número de canais não for um recurso exageradamente escasso, a última restrição linear deste problema pode ser “relaxada” através da substituição de L^* por um valor L maior do que L^* . Este relaxamento certamente permitirá que valores menores de mínima potência máxima possam ser obtidos. Lembremo-nos de que este valor condiciona o tempo de vida da bateria de um telefone celular, que é considerado um dos principais recursos a serem preservados.

Lamentavelmente este problema não é linear e a condição de binariedade introduz uma dificuldade extra na obtenção de sua solução. Entretanto, esta

mesma natureza binária da variável de decisão, torna possível transformar este problema em um outro linear (e conseqüentemente passível de ser resolvido por programas como o XPRESS) pagando-se o preço de um aumento substancial no número de variáveis de decisão. Recai-se assim no domínio do mesmo programa utilizado para a solução do problema de alocação mínima de canais.

No momento o interesse aqui é o de mostrar que este problema pode ser linearizado por partes, onde se tira partido fundamentalmente da condição binária do problema.

Para fim de clareza na apresentação dos resultados, considere o seguinte problema não linear binário:

$$\begin{aligned}
 \min z &= \max \{ \alpha_1 \cdot x_1, \alpha_2 \cdot x_2, \dots, \alpha_N \cdot x_N \} \\
 \text{sujeito a :} \\
 A \cdot \underline{x} &= \underline{1} \\
 \sum_{i=1}^K x_i &= L \\
 \underline{x} &\in \{0, 1\}^N
 \end{aligned} \tag{4.33}$$

4.5. Uma Heurística Para Determinação do Número Mínimo de Canais

Embora a seção 4.4.1 tenha abordado de modo formal o problema da determinação do número mínimo de canais para uma coleção de enlaces que devam atender uma SINR mínima, percebe-se que esta solução é de elevado esforço computacional, além de requerer possivelmente programas de computador comerciais para a sua solução.

Estes argumentos podem ser considerados suficientes para que se busque soluções sub-ótimas, mas de reduzido grau de demanda computacional. Essas

heurísticas ou procedimentos “ad hoc” podem ser gerados às centenas, bastando para isso criatividade e bom senso. Se elas serão viáveis sob um ponto de vista mais pragmático, se elas gerarão soluções razoavelmente próximas da solução ótima, se seu esforço computacional é pequeno o suficiente para permitir aplicações em tempo real, todas essas são questões de grande interesse e que precisam ser respondidas.

Dentre esta coleção de heurísticas destacam-se os métodos denominados de *gulosos* ou *gananciosos* (“greedy”) que tentam resolver o problema por procedimentos que tentem chegar da forma mais rápida e imediata naquilo que parece ser a solução ótima para o problema. Esta estratégia pode não ser a melhor opção, e ainda ocorrem casos onde ela não é sequer indicada. A título de exemplo é sabido que, principalmente em problemas de otimização no contínuo, algumas funções objetivo são tão mal comportadas, a melhor direção a se caminhar para maximizar essa função não coincide com a direção do gradiente (a mais gulosa no sentido de crescimento) da função objetivo.

O modelo formal que se assume nesta seção é o mesmo que foi discutido em seções anteriores.

Dispõe-se de um conjunto de L enlaces para os quais se conhece o conjunto Π_2 de todos os seus subconjuntos de dimensão 2 que são e-viáveis.

No caso específico de dimensão 2, a verificação da e-viabilidade é de complexidade bastante reduzida, como ilustrado abaixo:

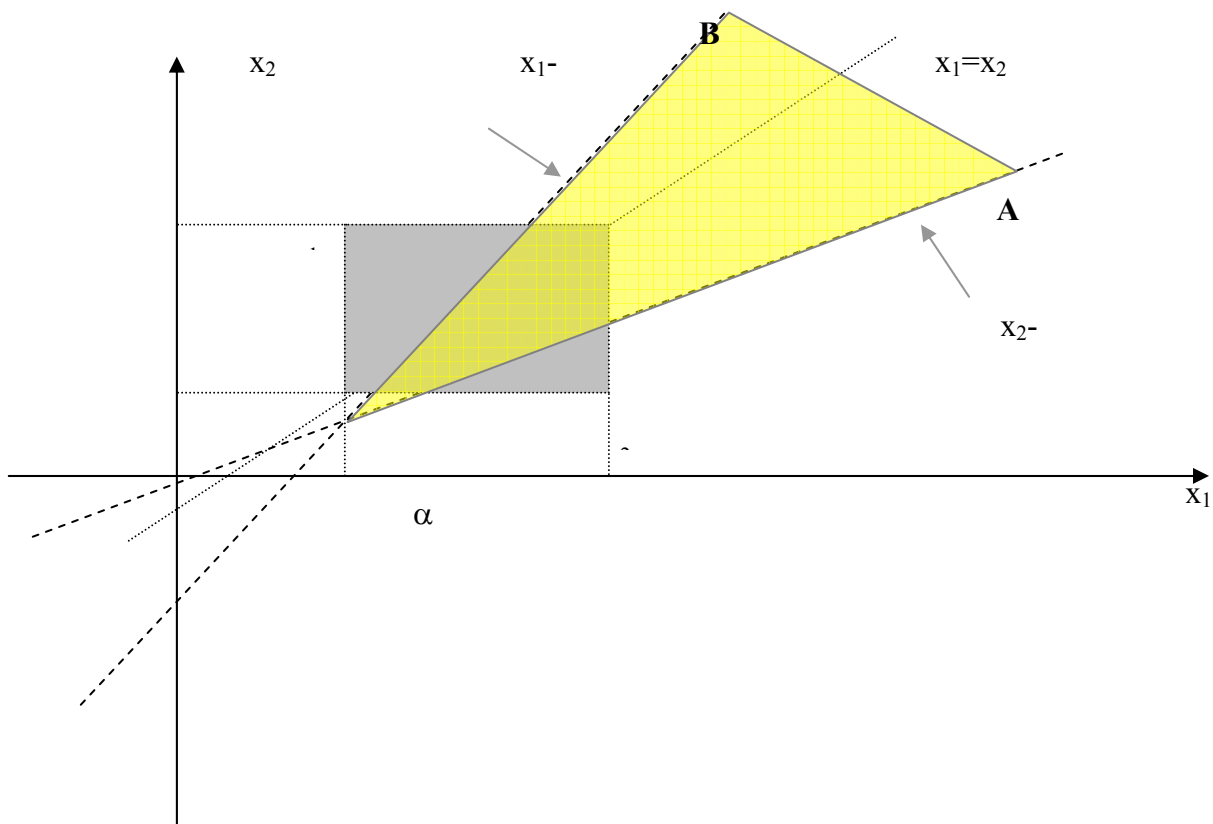


Figura 4-4 – Verificação de e-viabilidade em dimensão 2

Para que a região de potências factíveis (retângulo) tenha interseção com a que caracteriza a f-viabilidade, é necessário e suficiente que :

- O ponto P_1 esteja “acima” da reta A
- O ponto P_2 esteja “abaixo” da reta B

Estas duas condições implicam que:

$$\beta \geq \max[\eta_1 + g_{12} \cdot \alpha, \eta_2 + g_{21} \cdot \alpha] \quad (4.34)$$

Diz-se que o enlace i é *2-compatível* com o enlace j quando o par (i,j) pertencer a Π_2 . Caso contrário diz-se que enlace i é *incompatível* com o enlace j

Para evitar possíveis polarizações devido à ordem de escolha dos enlaces, pode-se promover uma permutação aleatória dos enlaces, e assim a sua lista fica na da forma $\{i_1, i_2, \dots, i_L\}$. Denota-se por $B(k)$ o conjunto de todos os enlaces que utilizam canal k

- faça, $C=1$, $B(1) = \{i_1\}$ e $B(2) = B(3) = \dots = \phi$
- para $j = 2$ a L
 - o para $k = 1$ a C
 - se enlace i_j é compatível com todos os enlaces em $B(k)$ então coloca-se i_j em $B(k)$ e *break*
 - o end
 - o se enlace i_j não foi alocado
 - faça $C = C+1$
 - coloca-se i_j em $B(C)$
 - o end
- end

Ao término deste algoritmo, C fornecerá uma estimativa para o número mínimo de canais.

É fácil perceber que a ação deste algoritmo é o de colocar enlaces em grupos já existentes desde que haja compatibilidade do enlace corrente com todos os elementos deste conjunto. No caso em que isto não ocorra, um novo conjunto é criado para contê-lo.

Observe que é possível que ao final, enlaces de um mesmo conjunto não sejam globalmente compatíveis porque o único teste efetuado foi o de 2-compatibilidade. Por exemplo, é possível que i_1 , i_2 e i_3 sejam compatíveis dois a dois sem que o sejam em grupo de três.

Entretanto o que se tem observado quando este algoritmo é testado em diversos exemplos é que na vastíssima maioria dos casos, os subconjuntos finais são de enlaces compatíveis. Assim a não verificação da compatibilidade dos

grupos de enlaces de cardinalidade três ou superior, representa uma considerável economia de esforço computacional.

Concebe-se então uma Fase 2 para este algoritmo, onde os subconjuntos gerados na fase discutida (doravante chamada de Fase 1) são globalmente verificados e se apresentarem incompatibilidade, esses subconjuntos de tamanho reduzido devem ser fracionados de modo a atender a desejada compatibilidade.

Não será aqui considerada nenhuma técnica especial de fracionamento deste conjunto porque sendo em geral pequeno (tamanho tipicamente inferior a 5), este conjunto pode ser por exemplo, fracionado por métodos de força bruta.

4.6. O Problema de Mínima Interferência Revisitado

Nos problemas anteriores, todas as partições de interesse feitas no conjunto de enlaces tinham que ser e-viáveis. Obter tais partições é uma tarefa extremamente demandante sob o ponto de vista de complexidade computacional. Assim são necessárias tentativas de abrandar o problema e obter soluções que não impliquem a necessidade de utilização de “solvers” sofisticados. Uma delas é a seguir discutida.

Considere o problema de N enlaces onde estão disponíveis F recursos denominados, por pura concisão, de “canais”. Toda vez que dois enlaces compartilham um mesmo canal ocorre interferência ente eles. Para descrever este compartilhamento de canais, considere uma variável X descrita na forma de uma matriz binária $N \times F$, onde X_{ij} vale 1 se e somente se canal j é utilizado pelo link i . Note que uma vez que cada enlace utiliza um único canal, tem-se que cada linha

de X contém apenas um único valor diferente de zero. A relação sinal-interferência-ruído expressa em (3.4) pode ser reescrita como:

$$SINR_i = \frac{\Gamma_i \cdot P_i}{n_i + \sum_{\substack{i=1 \\ j \neq i}}^N G_{ij} \cdot P_j \cdot \sum_{k=1}^F X_{ik} \cdot X_{jk}} \quad (4.35)$$

Onde algumas modificações foram feitas em termos de notação. Por exemplo, o ganho de enlace entre o transmissor e o receptor do enlace i chama-se agora Γ_i , fazendo com que G_{ii} não faça mais sentido. Podemos reutilizar essa variável definindo o ruído n_i como o produto $G_{ii} \cdot P_i$ e assim reescrever a expressão acima como:

$$SINR_i = \frac{1}{\sum_{i=1}^N \frac{G_{ij} \cdot P_j}{\Gamma_i \cdot P_i} \cdot \sum_{k=1}^F X_{ik} \cdot X_{jk}} \quad (4.36)$$

Se admitirmos que as potências são conhecidas, podemos redefinir a fração que aparece acima como Q_{ij} e assim:

$$SINR_i = \frac{1}{\sum_{i=1}^N Q_{ij} \cdot \sum_{k=1}^F X_{ik} \cdot X_{jk}} = \frac{1}{D_i(X)} \quad (4.37)$$

Um problema importante é determinar a matriz X de modo que o valor mínimo da SINR (ao longo dos enlaces i) observada nos enlaces seja maximizado. Uma vez que o numerador da SINR não depende da variável de decisão X , este problema é equivalente a:

$$\min_X z = \max_i D_i(X) \quad (4.38)$$

Onde :

$$D_i(\underline{X}) = \sum_{j=1}^L Q_{ij} \cdot \sum_{f=1}^F x_{if} \cdot x_{jf} \quad (4.39)$$

Sujeito às restrições:

$$\begin{aligned} \sum_{f=1}^F x_{if} &= 1 \\ x_{if} &\in \{0,1\} \end{aligned} \quad (4.40)$$

Este problema tem natureza combinatorial, o que o coloca em princípio numa categoria de problemas cujas soluções são computacionalmente pesadas. Existem F^N diferentes valores para X , mas devemos levar em consideração que não existe diferença em soluções que se distinguem por “renomeação” dos canais. Assim o número de soluções distintas é muito menor do que F^N , mas dependendo dos valores de F e N sua busca exaustiva pode ainda ser inviável.

Uma forma equivalente, porém mais compacta de descrever a solução X é por meio de um vetor \underline{Y} pertencente a $\{1,2,\dots,F\}^L$ da forma $\underline{Y}=(Y_1, Y_2, \dots, Y_L)$ onde Y_i descreve o canal usado pelo enlace i . Numa tentativa de reduzir as chances de visitar soluções idênticas em termos de renomeação de canais, será fixado $Y_i = 1$ para todas as soluções de interesse.

A lógica do algoritmo a ser proposto é do tipo guloso: inicia-se com uma solução arbitrária e busca-se melhorá-la progressivamente. A grande maioria dos algoritmos gulosos procede desta forma. Será mostrado, entretanto que o esforço computacional associado é muito pequeno e pode ser feito a partir de um *tableau*, a semelhança dos algoritmos simples de Programação Linear.

Para o melhor entendimento do algoritmo vejamos uma definição importante.

Definição 3

Duas soluções $\underline{Y}^{(1)}$ e $\underline{Y}^{(2)}$ são ditas *k-adjacentes* quando elas diferem em exatamente k posições.

O objetivo é descobrir relações explícitas de melhoria de solução em função da k -adjacência e procurar um caminho $\underline{Y}^{(1)}, \underline{Y}^{(2)}, \underline{Y}^{(3)}, \dots$ de soluções adjacentes que são progressivamente melhores.

4.6.1

O Caso da 1-Adjacência

Sejam $\underline{Y}^{(1)}$ e $\underline{Y}^{(2)}$ duas soluções 1-*adjacentes* onde se espera que a segunda seja melhor do que a primeira. Estas soluções são idênticas em todas as posições exceto em uma posição denominada u onde:

$$\begin{aligned} Y_u^{(1)} &= r \\ Y_u^{(2)} &= s \end{aligned} \quad (4.41)$$

Sejam $L^{(1)}$ e $L^{(2)}$ os subconjuntos de enlaces que utilizam, pela solução 1, os canais r e s , respectivamente.

Note que, na passagem da solução 1 para 2, todos os v enlaces pertencentes a $L^{(1)} - \{u\}$ terão seus níveis de interferência reduzidos de Q_{vu} enquanto que todos os enlaces v pertencentes a $L^{(2)}$ terão seus níveis de interferência aumentados de Q_{vu} . Finalmente o enlace u , que agora passa a pertencer a $L^{(2)}$, tem seu nível de interferência modificado para:

$$D_u = \sum_{j \in L^{(2)}} Q_{uj} \quad (4.42)$$

Será designado como *enlace crítico* da solução corrente \underline{Y} aquele onde ocorre o maior valor de D , por exemplo, q . Claramente, para que este valor D_{\max} possa ser reduzido, é necessário que se altere o canal de algum enlace que use o mesmo canal de q . Os candidatos u a esta alteração são apenas aqueles em que $Y_u = Y_q$, o que reduz consideravelmente o escopo da busca. Precisa-se agora determinar uma forma de escolher o novo canal de u . A idéia é percorrer todos os

valores possíveis, que não ultrapassem o valor D_{\max} . A seguir apresenta-se um algoritmo onde as idéias aqui expostas são implementadas:

ALGORITMO PARA 1-A

- Seja \underline{Y} a solução corrente, q o link crítico e r o seu canal ($r = Y_q$).
- Construa o conjunto $L^{(1)} = \{i \in \{1, 2, \dots, L\} \text{ tais que } Y_i = r\}$
- para todo $u \in L^{(1)}$ faça:
 - para todo $s \in \{1, 2, \dots, F\} - \{r\}$, faça:
 - Determine $L^{(2)} = \{i \in \{1, 2, \dots, L\} \text{ tais que } Y_i = s\}$
 - flag = *TRUE*
 - para todo $v \in L^{(2)}$, faça:
 - se $D_v + Q_{vu} \geq D_q$ então flag = *FALSE* and break
 - end
 - se flag = *TRUE* e $\sum_{v \in L^{(2)}} Q_{uv} \geq D_q$ então flag = *FALSE*
 - se flag = *TRUE* então uma solução foi achada. Faça $Y_i = s$ e FIM
 - end
- end

Entenda que este algoritmo poderia ser mais “guloso” ainda, isto é, determinar todas as possibilidades e escolher a que provoca a maior redução do valor de D_w . Isto implicaria em percorrer todas os casos possíveis, armazená-los e fazer a decisão no final.

4.6.2 O Caso da 2-Adjacência

Assume-se que algoritmo anterior foi aplicado e que se chegou a uma solução \underline{Y}^* que não pode ser melhorada pela busca de uma outra que lhe seja 1-

adjacente. Para que o nível máximo de interferência possa ser reduzido, é necessário que se altere o canal de algum enlace u que use o mesmo canal do enlace crítico w . Entretanto, qualquer que seja o novo valor deste canal (i.e. Y_u), não haverá redução do nível máximo de interferência porque não há, pela hipótese assumida no início deste parágrafo, solução 1-adjacente que reduza este nível. Conseqüentemente qualquer que seja a escolha deste canal, algum enlace que o compartilha, terá que ter seu canal alterado. Esta imbricação resulta em uma redução considerável nos vizinhos 2-adjacentes, que devem ser procurados para melhorar a solução. Apresenta-se a seguir a descrição de um algoritmo que explora esta idéia:

ALGORITMO PARA 2-A

- Seja \underline{Y} a solução corrente, w o enlace crítico e r o seu canal ($r = Y_w$)
- construa o conjunto $L^{(1)} = \{i \in \{1, 2, \dots, L\} \text{ tais que } Y_i = r\}$
- para todo $u \in L^{(1)}$ faça:
 - para todo $s \in \{1, 2, \dots, F\} - \{r\}$, faça:
 - Determine $L^{(2)} = \{i \in \{1, 2, \dots, L\} \text{ tais que } Y_i = s\}$
 - para todo $v \in L^{(2)}$, faça:
 - flag = TRUE
 - para todo $a \in L^{(2)} - \{v\}$ faça:
 - se $D_a + Q_{av} - Q_{av} > D_q$ então flag=FALSE break
 - end
 - se flag = TRUE e $\sum_{a \in L^{(2)} - \{v\}} Q_{ua} \geq D_q$ então flag = FALSE
 - se flag = FALSE então break
 - para todo $t \in \{1, 2, \dots, F\} - \{s\}$
 - Calcule $L^{(3)} = \{i \in \{1, 2, \dots, L\} \text{ tais que } Y_i = t\}$
 - para todo $w \in L^{(3)}$, faça:
 - se $D_w + Q_{wv} \geq D_q$ então flag=FALSE break
 - end
 - se flag=TRUE e $\sum_{a \in L^{(3)}} Q_{va} \geq D_q$ então flag=FALSE
 - se flag = FALSE então break
 - end
 - se flag = TRUE então faça $Y_u = s$ $Y_v = t$ e FIM
 - end
 - end
- end

A grande vantagem desta estratégia é que ela pode ser repetida dando origem a algoritmos para busca de soluções melhores por 3-adjacência, 4-adjacência e assim por diante. Entretanto a busca por k-adjacência só pode ser feita quando as buscas por adjacências de ordem inferior a k falham.

Assim um algoritmo geral que explore esses conceitos pode ser concebido como descrito abaixo:

ALGORITMO GERAL

1. faça $k = 1$
2. procure uma solução k-adjacente “melhor”
3. se achou, faça $k = 1$ e vá para 2.
4. caso contrário, faça $k = k+1$ e vá para 2

Observe que neste algoritmo não está determinado o critério de parada. Este pode ter várias formas como por exemplo a) pelo número de iterações b) pelo valor máximo de k.

4.7.

Metáforas Evolucionais para o Problema de Mínima Interferência

Todos os métodos de solução discutidos nesta tese são ou métodos formais locais ou heurísticas gulosas também locais. Por local quer-se dizer que são métodos iterativos cuja seqüência de soluções produzida é monotonicamente melhor (no sentido de reduzir/aumentar a função objetivo do problema de minimização/maximização), fazendo com que esta solução seja capturada por ótimos locais da função objetivo.

Em problemas de otimização em geral deseja-se determinar as soluções ótimas globais, mas as técnicas para sua determinação são frequentemente sofisticadas e computacionalmente caras. Uma possível linha de algoritmos

criados para permitir a busca de soluções globais é aquela genericamente denominada de *métodos evolucionais*. Estes métodos operam por meio de uma metáfora, na qual ha uma correspondência entre elementos abstratos do mundo evolucionar e elementos concretos do mundo do modelo em questão. Uma destas metáforas de grande frequência de aplicação nestes problemas é aquela que dá origem aos chamados *algoritmos genéticos*.

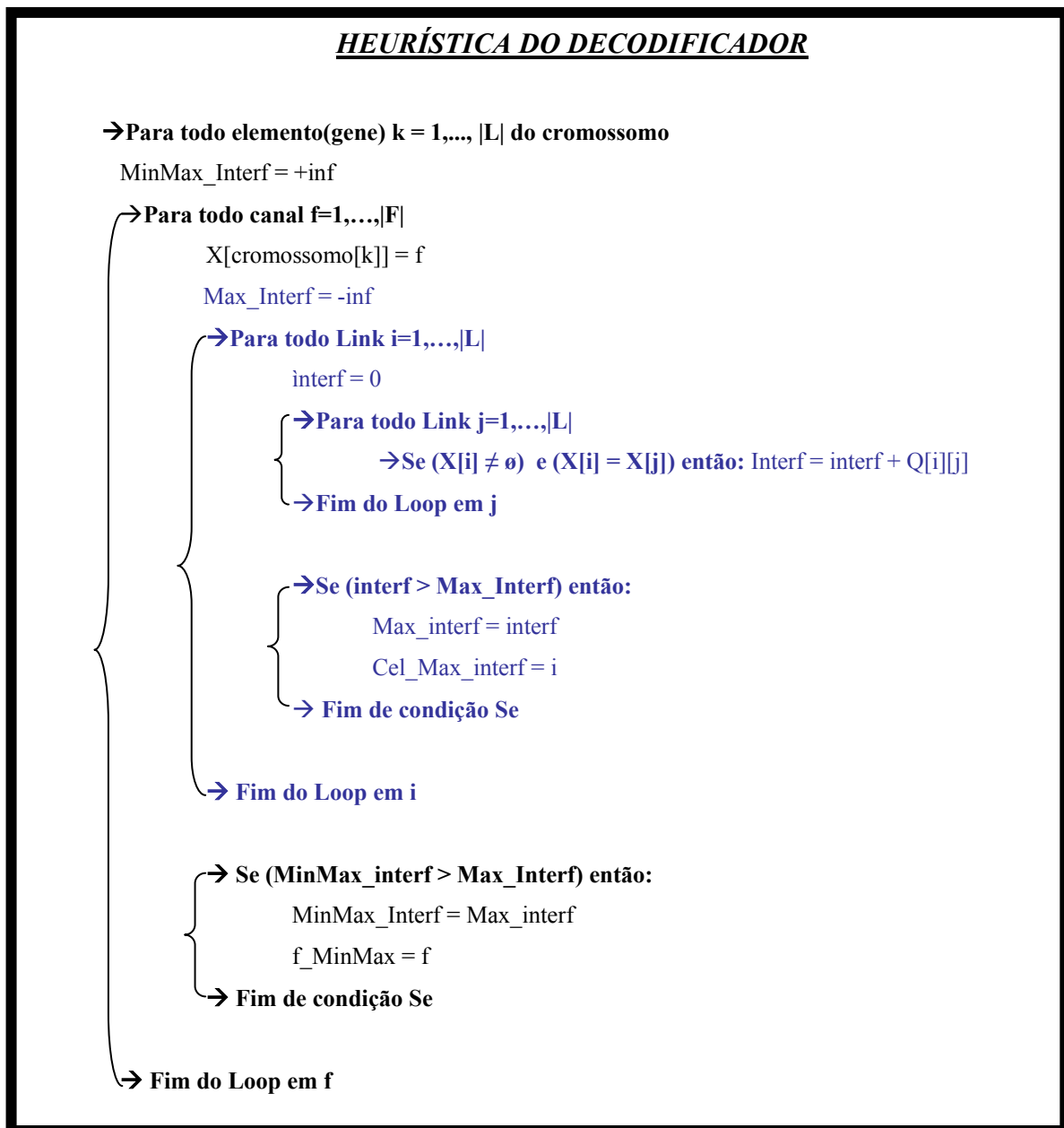
Nesta classe de algoritmos criam-se populações de indivíduos (em que cada elemento contém um cromossomo) que metaforicamente representam as soluções admissíveis para o problema de otimização. Cada indivíduo, por meio de seu cromossomo, possui um indicador de qualidade (valor da função objetivo). A idéia do algoritmo é o de promover cruzamentos de membros desta população produzindo uma nova geração (cujos cromossomos foram gerados pelos processos de *crossover* e *mutação*) da qual são selecionados os indivíduos “mais capazes” (novamente no sentido da função objetivo) como se os princípios evolucionistas darwinianos comandassem esse processo. Esta produção de novas gerações se repete perpetuamente, até que os membros da geração sobrevivente atinjam níveis de qualidade satisfatórios.

O problema de interesse a ser resolvido por um algoritmo genético é o discutido na seção anterior e caracterizados pelas expressões (4.38) a (4.40).

Em grande parte, o sucesso ou fracasso da aplicação de algoritmos genéticos está relacionado com a representação cromossomial da solução admissível do problema de otimização. Embora a formulação acima induza uma definição natural para cromossomo, achou-se que esta não levaria em conta nenhuma informação sobre a instância do problema. Assim indivíduos serão caracterizados por um cromossomo descrito por um vetor de comprimento L (no. de enlaces) e que contém os inteiros de 1 a L numa ordem aleatória. Esta ordem será a seguida no processo de atribuição de canais aos enlaces.

Associado a cada cromossomo I existe um vetor de atribuição S , de mesma dimensão de I , onde $S(k)$ é o canal atribuído ao enlace k . Esta atribuição se fará na ordem dos enlaces expressa no cromossomo I , e o canal $S(k)$ corresponderá ao caso que minimiza a máxima interferência até o presente momento. Este

procedimento é executado por meio de um *decodificador*, cuja descrição algorítmica é a seguir apresentada:



Depois de construído vetor de atribuição S para cada cromossomo I , este vetor S é avaliado e armazenado em conjunto com cada cromossomo.

Esta avaliação não é nada mais do que o valor máximo de interferência produzido e seu cálculo segue o procedimento abaixo descrito.

A população de indivíduos foi inicializada gerando-se permutações aleatórias do conjunto de enlaces e o processo de geração de populações

sucessoras foi realizado por meio de operadores de embaralhamento, crossover de ordem e mutação de ordem e posição.

AVALIAÇÃO DO INDIVÍDUO (K)

Max_Interf = -inf

→ Para todo Link $i=1, \dots, |L|$

interf = 0

→ Para todo Link $j=1, \dots, |L|$

→ Se $(X[i] \neq \emptyset)$ e $(X[i] = X[j])$ então: $\text{Interf} = \text{interf} + Q[i][j]$

→ Fim do Loop em j

→ Se $(\text{interf} > \text{Max_Interf})$ então:

Max_interf = interf

Cel_Max_interf = i

→ Fim de condição Se

→ Fim do Loop em i

Avaliação[K] = Max_interf

Cel_Max_Interf_Individuo[K] = Cel_Max_interf