



**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Um Método para Determinação de Pontos de Operação com  
Diversidade em Linha Digital de Assinante usando  
Balanceamento de Espectro e Algoritmo Evolucionário**

**Johelden Campos Bezerra**

**BELÉM - PARÁ**  
**Fevereiro de 2011**

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Johelden Campos Bezerra

**Um Método para Determinação de Pontos de Operação com  
Diversidade em Linhal Digital de Assinante usando  
Balanceamento de Espectro e Algoritmo Evolucionário**

Tese de Doutorado submetida à Banca Examinadora do Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal do Pará como um dos requisitos para obtenção do título de “Doutor em Engenharia Elétrica”.

Orientador: Evaldo Gonçalves Pelaes

Co-orientador: Aldebaro Klautau

Belém-Pará-Brasil

2011

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Um Método para Determinação de Pontos de Operação com Diversidade em  
Linha Digital de Assinante usando Balanceamento de Espectro e Algoritmo  
Evolucionário**

Autor: Johelden Campos Bezerra

TESE DE DOUTORADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA \_\_\_\_\_ PARA OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES. APROVADA EM / / .

**BANCA EXAMINADORA:**

---

Prof. Dr. Evaldo Gonçalves Pelaes - Orientador UFPA

---

Prof. Dr. Aldebaro B. da Rocha Klautau Júnior - Co-orientador UFPA

---

Prof. Dr. Renato Rocha Lopes - UNICAMP

---

Prof. Dr. Antônio Marcos de Lima Araújo - IFPA/IESAM

---

Profa. Dra. Carolina de Mattos Affonso - UFPA

---

Prof. Dr. Ronaldo de Freitas Zampolo - UFPA

Visto:

---

Prof. Dr. Marcus Vinícius Alves Nunes  
COORDENADOR DO PPGEE/ITEC/UFPA

# Resumo

Este trabalho apresenta um método para encontrar um conjunto de pontos de operação, os quais são ótimos de Pareto com diversidade, para linhas digitais de assinante (DSL - *digital subscriber line*). Em diversos trabalhos encontrados na literatura, têm sido propostos algoritmos para otimização da transmissão de dados em linhas DSL, que fornecem como resultado apenas um ponto de operação para os modems. Esses trabalhos utilizam, em geral, algoritmos de balanceamento de espectro para resolver um problema de alocação de potência, o que difere da abordagem apresentada neste trabalho. O método proposto, chamado de *diverseSB*, utiliza um processo híbrido composto de um algoritmo evolucionário multiobjetivo (MOEA - *multi-objective evolutionary algorithm*), mais precisamente, um algoritmo genético com ordenamento por não-dominância (NSGA-II - *Non-Dominated Sorting Genetic Algorithm II*), e usando ainda, um algoritmo de balanceamento de espectro. Os resultados obtidos por simulações mostram que, para uma dada diversidade, o custo computacional para determinar os pontos de operação com diversidade usando o algoritmo *diverseSB* proposto é muito menor que métodos de busca de “força bruta”. No método proposto, o NSGA-II executa chamadas ao algoritmo de balanceamento de espectro adotado, por isso, diversos testes envolvendo o mesmo número de chamadas ao algoritmo foram realizadas com o método *diverseSB* proposto e o método de busca por força bruta, onde os resultados obtidos pelo método *diverseSB* proposto foram bem superiores do que os resultados do método de busca por força bruta. Por exemplo, o método de força bruta realizando 1600 chamadas ao algoritmo de balanceamento de espectro, obtém um conjunto de pontos de operação com diversidade semelhante ao do método *diverseSB* proposto com 535 chamadas.

**Palavras-chave:** linha digital do assinante, algoritmo de balanceamento de espectro, otimização multiobjetivo, algoritmo genético, NSGA-II, soluções ótimas de Pareto.

# Abstract

This work presents a method for finding diversity set operating points, which are Pareto optimal and diverse, to digital subscriber lines (DSL). Several works presented in the literature have proposed algorithms for optimizing data transmission in DSL lines, which results in a unique operating point for the modems. These works use spectrum balancing algorithms to solve the power allocation problem, which differs from the approach presented in this work. The proposed method, called *diverseSB*, uses a hybrid solution that consists of the non-dominated sorting genetic algorithm-II (NSGA-II), based on a multi-objective optimization, and a spectrum balancing algorithm. The simulation results showed that, for a given diversity, the computational cost for find the operating points with diversity using the *diverseSB* proposed algorithm is much smaller than “brute-force” search methods. In the proposed method, NSGA-II perform calls to the spectrum balancing algorithm adopted, so many tests involving the same number of calls to the algorithm were performed with the *diverseSB* proposed and with the brute-force search method, and the results of *diverseSB* proposed were better than brute-force search method. For example, to obtain a diverse set operating points the brute-force method performs 1,600 calls to the spectrum balancing algorithm and the *diverseSB* proposed performed 535 calls.

**Keywords:** digital subscriber line, spectrum balancing algorithm, evolutionary algorithm, genetic algorithm, NSGA-II, Pareto optimal solution.

# Agradecimentos

A Deus, pelos dons concedidos.

Ao meu orientador, Prof. Dr. Evaldo Pelaes, pelas amizade, pelo tempo dedicado a minha orientação e por suas palavras de apoio: “Isso é Tranquilo!”, “Vai dar tudo certo!”.

Ao meu co-orientador, Prof. Dr. Aldebaro Klautau, pela amizade e pelo tempo dedicado a minha orientação.

Aos meus pais (Orlando e Iacy) que me deram educação e me mostraram o caminho certo a seguir.

A minha esposa e a minha filha Luiza por fazerem parte da minha vida.

Às minhas irmãs e minha vó pelas palavras de incentivo e orações.

A todos os meus familiares pelo grande incentivo e apoio durante esta jornada.

À Ericsson, pelo apoio financeiro.

Aos meus amigos do LaPS, pelas críticas e sugestões.

*A Deus, aos meus pais, minha esposa Michelle e filha Luiza, minhas irmãs, minha avó, tios e sobrinhas. In memoriam a minha tia Iacymira de Souza Campos.*

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Siglas</b>	<b>xiv</b>
<b>Lista de Símbolos</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Linha Digital de Assinante - DSL . . . . .	1
1.2 O Canal DSL . . . . .	2
1.3 Gerenciamento em Linhas DSL . . . . .	4
1.4 O Estado da Arte . . . . .	5
1.5 Contribuição da Tese . . . . .	6
<b>2 Algoritmos DSM</b>	<b>7</b>
2.1 Introdução . . . . .	7
2.2 Modelo do Sistemas DSL . . . . .	7
2.3 Formulação do Problema de Gerenciamento do Espectro . . . . .	9
2.4 Iterative Water-Filling . . . . .	10
2.5 Optimal Spectrum Balancing . . . . .	11
2.6 Iterative Spectrum Balancing . . . . .	13
2.7 Autonomous Spectrum Balancing . . . . .	13
2.7.1 Successive Convex Approximation for Low-complexity (SCALE) . . . . .	15
2.8 Considerações Finais . . . . .	16



---

<b>3</b>	<b>Algoritmo Evolucionário para Otimização Multiobjetivo</b>	<b>17</b>
3.1	Algoritmo Evolucionário . . . . .	17
3.2	Algoritmo Genético . . . . .	18
3.2.1	Elementos do Algoritmo Genético . . . . .	18
3.2.2	Parâmetros do AG . . . . .	24
3.3	Elitist Nondominated Sorting Genetic Algorithm - NSGA-II . . . . .	25
<b>4</b>	<b>Geração de Pontos de Operação com Diversidade usando Algoritmo Evolucionário</b>	<b>26</b>
4.1	Introdução . . . . .	26
4.2	Descrição do Problema . . . . .	27
4.2.1	Balanceamento de Espectro Multiobjetivo . . . . .	27
4.2.2	Análise do Problema de Balanceamento de Espectro Multiobjetivo . . . . .	30
4.3	O Método Proposto para Solucionar o Problema de DSSB . . . . .	34
4.3.1	Algoritmo de Balanceamento de Espectro para Soluções Multiobjetivo . . . . .	34
4.3.2	Algoritmo Proposto: <i>diverseSB</i> . . . . .	36
<b>5</b>	<b>Resultados Obtidos</b>	<b>39</b>
5.1	Simulações com 2-usuários . . . . .	40
5.2	Simulações com 3-usuários . . . . .	43
5.3	Simulações para cenários com mais de 3 usuários . . . . .	43
<b>6</b>	<b>Conclusão</b>	<b>47</b>
	<b>Referências bibliográficas</b>	<b>50</b>
	<b>Trabalhos Publicados Pelo Autor</b>	<b>54</b>
<b>A</b>	<b>Algoritmos DSM</b>	<b>55</b>
A.1	Algoritmo Iterative Water-Filling - IWF . . . . .	55
A.2	Optimal Spectrum Balancing - OSB . . . . .	55
A.2.1	OSB para o caso de 2 usuário . . . . .	55
A.2.2	OSB para o caso de $N$ usuários . . . . .	57
A.3	Iterative Spectrum Balancing - ISB . . . . .	57
A.4	Autonomous Spectrum Balancing - ASB . . . . .	57

---

A.5	Algoritmo Successive Convex Approximation for Low Complexity - SCALE . . . .	58
-----	--	----

# Lista de Figuras

1.1	Degradação da taxa de dados devido ao <i>crosstalk</i> presente em uma linha ADSL transmitindo em <i>downstream</i> . . . . .	2
1.2	Ilustração do Near-end <i>crosstalk</i> (NEXT) e Far-end <i>crosstalk</i> (FEXT), do ruído impulsivo e da interferência de radiofrequência. . . . .	3
3.1	Fluxograma do princípio de funcionamento do AG. . . . .	19
3.2	Exemplo de torneio entre 4 jogadores. Os elementos de dentro do retângulo são as soluções que têm maior aptidão em cada comparação e formam a nova população de indivíduos. . . . .	21
3.3	Operador de cruzamento de um ponto. . . . .	21
4.1	Exemplo de dois conjuntos de solução para exemplificar diversidade, sendo (a) solução não dominada com pouca diversidade (b) solução não dominada com boa diversidade. . . . .	27
4.2	Exemplo de dois conjuntos de soluções $\bar{\mathbf{S}}_Z^a$ e $\bar{\mathbf{S}}_Z^b$ (veja valores na Tabela 4.1) que tem o mesmo valor (28,6) de diversidade de acordo com a métrica <i>spacing</i> da Eq. (4.5). . . . .	31
4.3	Fluxograma do método proposto para solucionar o problema de DSSB. . . . .	37
4.4	Algoritmo <i>diverseSB</i> Proposto. . . . .	38
5.1	Cenário DSL geral com 4-usuários. . . . .	39
5.2	Cenário DSL com 2-usuários ( $N = L = 2$ ). . . . .	41

5.3	Pontos de operação para o cenário de 2 usuários usando (a) discretização uniforme, (b) discretização logarítmica, (c) discretização aleatória e (d) método proposto ( <i>diverseSB</i> ). Todos os quatro métodos realizam 144 <i>num_SB</i> , de onde $Z = 12$ pontos são selecionados (alguns pontos não podem ser distinguíveis visualmente - estão muito próximos ou se sobrepõem), tal com em (d)). . . . .	42
5.4	Diversidade para diferentes valores de <i>num_SB</i> . Na figura são identificados os métodos de discretizações: <i>Uniforme</i> , <i>Logaritmo</i> , <i>Aleatório</i> e o método proposto <i>diverseSB</i> como <i>Proposto</i> . . . . .	43
5.5	Cenário DSL com 3-usuários. . . . .	44
5.6	OS resultados para as simulações com 3-usuários, (a) discretização uniforme, (b) discretização logarítmica, (c) discretização aleatória e (d) método proposto <i>diverseSB</i> . Todos os quatro métodos usaram 1.000 <i>num_SB</i> e forneceram $Z = 100$ pontos de operação. . . . .	44
5.7	Sumário dos resultados de diversidade obtidos com as simulações apresentadas, onde os valores foram normalizados pelos resultados do método proposto. . . . .	46

# Lista de Tabelas

2.1	Comparação entre os principais aspectos dos algoritmos de DSM. . . . .	16
4.1	Dois conjuntos diferentes de soluções $\bar{\mathbf{S}}_Z^a$ e $\bar{\mathbf{S}}_Z^b$ para $L = 2$ e $Z = 4$ . As soluções são mostradas nos gráficos da Fig. 4.2. . . . .	31
A.1	Algorithm 1 - Iterative Waterfilling. . . . .	55
A.2	Algoritmo 2 - OSB para 2 users. . . . .	56
A.3	Algoritmo 3 - OSB (N usuários). . . . .	57
A.4	Algoritmo 4 - ISB. . . . .	57
A.5	Algoritmo ASB-I. . . . .	58
A.6	Algoritmo Successive Convex Approximation for Low Complexity (SCALE) . . . .	58

# Lista de Siglas

ADSL	-	<i>Asymmetric Digital Subscriber Lines</i>
AG	-	Algoritmo Genético
ASB	-	<i>Autonomous Spectrum Balancing</i>
AWGN	-	Ruído Gaussiano Branco Aditivo - <i>Additive White Gaussian Noise</i>
DMT	-	Multi-tom discreto
DSL	-	Linha Digital de Assinante ( <i>Digital Subscriber Line</i> )
DSM	-	Gerenciamento Dinâmico do Espectro ( <i>Dynamic Spectrum Management</i> )
DSSB	-	Balanceamento de Espectro com Solução Diversa ( <i>Diverse Set Spectrum Balancing</i> )
EC	-	Escritório Central
FDD	-	Duplexação por Divisão de Frequência
FEXT	-	<i>Far-end crosstalk</i>
FOA	-	Função de Objetivo Agregado
HDSL	-	<i>High-bit-rate Digital Subscriber Lines</i>
ISB	-	<i>Iterative Spectrum Balancing</i>
ISI	-	Interferência Intersimbólica ( <i>Intersymbol Interference</i> )
IWF	-	<i>Iterative Water-Filling</i>
MOEA	-	Algoritmo Evolucionário Multiobjetivo ( <i>Multi-Objective Evolutionary Algorithm</i> )
NEXT	-	<i>Near-end crosstalk</i>
NSGA-II	-	AG por Ordenamento por Não-dominância ( <i>Non-Dominated Sorting Genetic Algorithm II</i> )
<i>num_SB</i>	-	Número de vezes que um algoritmo de SB foi executado
OSB	-	<i>Optimal Spectrum Balancing</i>
PSD	-	Densidade Espectral de Potência ( <i>Power Spectrum Density</i> )
PSO	-	<i>Particle Swarm Optimization</i>
RA	-	<i>Rate-Adaptive - Taxa Adaptativa</i>
SB	-	Balanceamento de Espectro - <i>Spectrum Balancing</i>
SBX	-	<i>Simulated Binary Crossover</i>
SCALE	-	<i>Successive Convex Approximation for Low-complexity</i>
SINR	-	Relação Sinal - Interferência mais Ruído - <i>Signal to Interference plus Noise Ratio</i>
SIR	-	Relação Sinal-Interferência - <i>Signal-to-Interference Ratio</i>
SMC	-	<i>Spectrum Management Center</i>
SNR	-	Relação Sinal-Ruído - <i>Signal-to-Noise Ratio</i>
TDD	-	Duplexação por Divisão do Tempo
TR	-	Terminal Remoto
VDSL	-	<i>Very-high-bit-rate Digital Subscriber Lines</i>

# Lista de Símbolos

$\bar{\mathbf{H}}$	- Matriz de ganho dos canais direto e de <i>crosstalk</i>
$\mathbf{N}$	- Matriz de ruído (todos menos <i>crosstalk</i> )
$\mathbf{O}$	- Vetor com todas as funções objetivos
$\mathbf{S}$	- Matriz de PSDs de todos os usuários de dimensão $N \times K$
$\bar{\mathbf{S}}_{\infty}^{\dagger}$	- PSDs ótimas de Pareto
$\mathcal{L}$	- Lagrangeano do problema de otimização
$\mathcal{O}_i$	- Função objetivo $i$
$\Delta$	- O tamanho passo para a discretização dos eixos dos pesos
$\omega$	- Variável peso, usado na formulação de soma ponderada
$\sigma_k^n$	- PSD de ruído na linha $n$
$\mathbf{P}_i$	- População $i$
$\mathbf{x}_k$	- Vetor do sinal transmitido no tom $k$
$\mathbf{y}_k$	- Vetor do sinal recebido no tom $k$
$\mathbf{z}_k$	- Vetor do ruído aditivo no tom $k$
$\tilde{\alpha}_k^n$	- Ganho do canal de <i>crosstalk</i>
$\tilde{\alpha}_k^n$	- Ganhos do canal de <i>crosstalk</i>
$\tilde{\sigma}_k$	- PSD de ruído da linha virtual
$\tilde{b}_k^n$	- Bit loading da linha de referência $n$ no tom $k$
$\tilde{s}_k^n$	- PSD transmitida pela linha de referência
$b_k^n$	- Bit loading do usuário $n$ no tom $k$
$d_z$	- Representa o menor valor da soma das diferenças absolutas entre dois pontos de uma solução no espaço objetivo
$f_s$	- Taxa de símbolos
$G$	- Número de pontos em cada eixo (faixa de valores dos pesos)
$h_k^{n,m}$	- Canal do TX $m$ para RX $n$ no tom $k$
$I_{max}$	- Número máximo de Gerações
$N$	- Número de usuários
$L$	- Dimensão do Espaço Objetivo
$P_n$	- Potência máxima do usuário $n$
$p_n$	- Potência transmitida
$r_n$	- Taxa de dados
$s^n$	- Vetor contendo as PSDs do usuário $n$
$w^n$	- Coeficiente de peso
$Z$	- Tamanho da população

# Capítulo 1

## Introdução

Apresenta-se neste capítulo uma breve introdução sobre a tecnologia DSL (Linha Digital de Assinante - “Digital Subscriber Line”), enfatizando-se os efeitos de degradação do sinal e realizando uma breve introdução à técnica de gerenciamento dinâmico do espectro (DSM - “Dynamic Spectrum Management”), usada como alternativa para minimizar os efeitos dessas degradações. Na sequência é apresentado o estado da arte dos trabalhos que seguem a mesma linha de pesquisa deste. O capítulo é finalizado com uma seção apresentando a contribuição da tese.

### 1.1 Linha Digital de Assinante - DSL

A tecnologia DSL surgiu para atender os anseios da sociedade por serviços mais rápidos e mais confiáveis para transmissão de dados, pois os serviços disponíveis apresentavam diversos problemas tais como: sinal de ocupado, congestionamento no tráfego de dados e larguras de banda muito estreitas. A família xDSL (onde o “x” representa as variantes ADSL, VDSL, HDSL, entre outras) é uma tecnologia para transmissão digital de dados, áudio e vídeo, através do par-trançado, não apenas para grandes corporações, mas também para usuários residenciais e pequenas empresas. A grande vantagem da tecnologia xDSL é a facilidade de ser instalada na infra-estrutura telefônica de fio de cobre existente.

A idéia básica das tecnologias xDSL é ter um modem em cada um dos extremos da linha telefônica de cobre, um do lado do usuário e o outro no lado do provedor do serviço. As linhas telefônicas são utilizadas com a ajuda de avançadas técnicas de modulação.

Existem vários tipos de tecnologia DSL, as quais diferem-se principalmente, pelo modo de operação, velocidades de transmissão nos sentidos de *downstream* (do provedor de serviço para o usuário) e *upstream* (do usuário para o provedor de serviço) e pelas suas aplicações típicas. Algumas dessas tecnologias são ditas simétricas, ou seja, suas velocidades em ambas as direções são as mesmas. Porém, a grande maioria das tecnologias de modems xDSL são assimétricas, pois o tráfego *downstream* em uma rede de comunicações é muito maior do que o tráfego *upstream*.



## 1.2 O Canal DSL

A transmissão DSL ocorre entre modems ligados por par-trançado, onde muitos pares são agrupados em um mesmo cabo. O agrupamento dos pares trançados provoca o surgimento do acoplamento eletromagnético ( *Crosstalk*) entre as linhas. O  *Crosstalk* pode ser considerado o principal distúrbio no sistema de transmissão, o qual degrada o desempenho dos demais serviços compartilhados no cabo. A Fig. 1.1 ilustra a degradação da taxa de transmissão de dados, quando um modem ADSL ( *asymmetric digital subscriber lines*) está transmitindo em uma linha com  *Crosstalk* (ou seja, na presença de uma outra linha interferente com serviço ADSL) em comparação a uma linha sem ruído. Nela observa-se que o  *Crosstalk* exerce uma grande influência na taxa de dados para as distâncias de até 3,5 km, porém após essa distância, a atenuação da linha torna-se bem mais significativa, praticamente equivalente ao efeito do  *Crosstalk*<sup>1</sup>, o que pode ser verificado pela aproximação das duas curvas e, inclusive alguns serviços como VDSL não são especificados para linhas com comprimento maiores que 1.200 m [1]. Minimizar a interferência do  *Crosstalk* implica em cuidados na fabricação, instalação, manutenção e administração da rede telefônica.

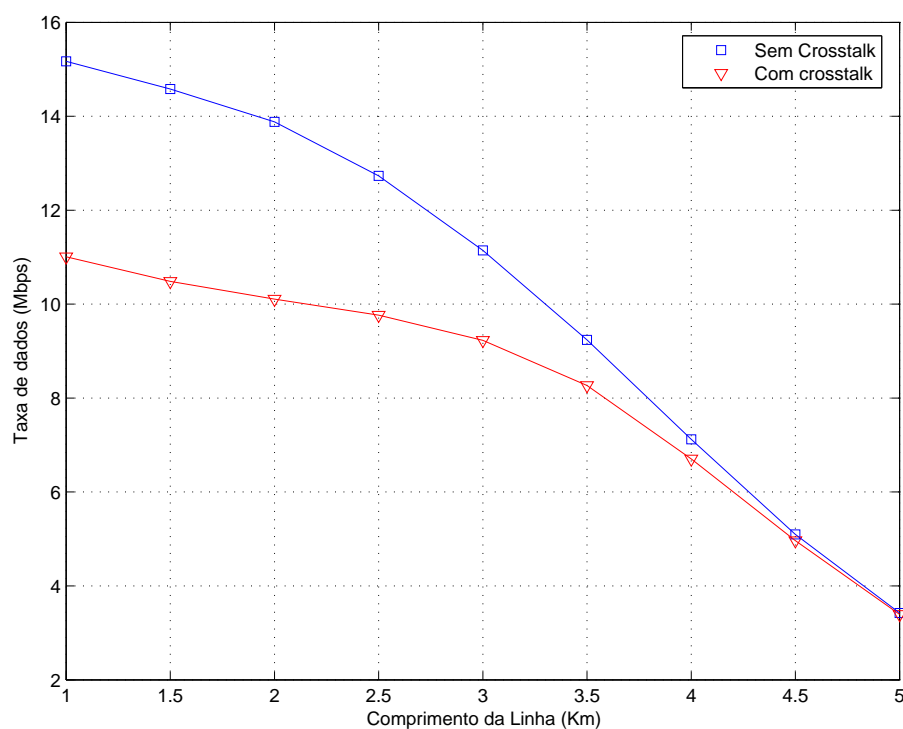


Fig. 1.1: Degradação da taxa de dados devido ao  *Crosstalk* presente em uma linha ADSL transmitindo em  *downstream*.

O  *Crosstalk* é sensível à frequência e ao nível de potência do sinal. A energia de acoplamento em altas frequências é maior do que em baixas frequências. Consequentemente, se dois sinais são transmitidos em iguais valores de potência, o nível de interferência de  *Crosstalk* entre os mesmos será

<sup>1</sup>O par-trançado sofre interferência de outras fontes de ruído, tais como, as apresentadas na Fig. 1.1.

maior na faixa das altas frequências. Além disso, o *crosstalk* também é diretamente proporcional à potência do sinal de distúrbio (proveniente do sistema interferente). Quanto mais forte for o sinal de distúrbio, maior será o *crosstalk*, o que justifica o uso da técnica de limitar a energia dos sinais aplicados na rede, como um meio de controlar este tipo de interferência.

O *crosstalk* pode ocorrer basicamente de duas formas, como mostra a Fig. 1.2. A primeira, *Near-end crosstalk* - NEXT, ocorre quando é causado por interferentes transmitindo em direção oposta à direção da transmissão do usuário de interesse, por isso o receptor e os geradores de interferência estão localizados no mesmo extremo do cabo. O FEXT (*Far-end crosstalk*), por sua vez, é originado por interferentes transmitindo na mesma direção de transmissão do usuário de interesse, ou seja, estão localizados em extremos contrários.

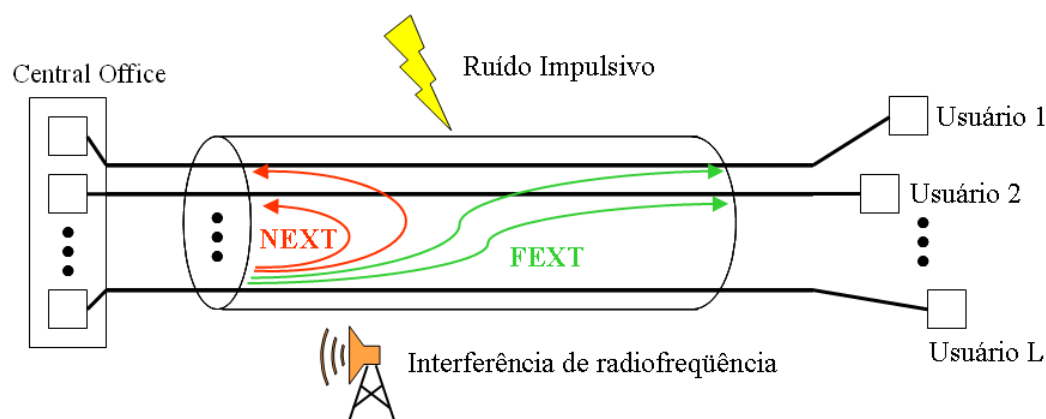


Fig. 1.2: Ilustração do *Near-end crosstalk* (NEXT) e *Far-end crosstalk* (FEXT), do ruído impulsivo e da interferência de radiofrequência.

O NEXT é geralmente mais forte que o FEXT. Para evitar o NEXT, a transmissão DSL usa duplexação por divisão de frequência (FDD), onde são separadas as bandas de frequência de *upstream* e *downstream*, ou duplexação por divisão do tempo (TDD), onde a transmissão *upstream* ocorre em um *slots* de tempo distintos da transmissão *downstream*. Os padrões Norte Americano e o Europeu usam FDD, enquanto o padrão Japonês usa TDD [1, 2]. Neste trabalho considera-se sistemas FDD e, por isso, o FEXT torna-se o principal componente do *crosstalk* a ser combatido com o uso das técnicas de DSM.

Além do *crosstalk*, existem outras fontes de interferência em DSL, com ilustrado na Fig. 1.2, chamadas geralmente de ruído. Os tipos de ruído que têm mais relevância em sistemas DSL são os que têm comportamento não-estacionário, como os de rádio interferência (principalmente provenientes de rádio AM e de rádio amador) e os ruídos impulsivos (gerados por equipamentos elétricos, campainhas de telefones, fenômenos atmosféricos, etc.).

## 1.3 Gerenciamento em Linhas DSL

O sistema DSL é projetado para operar nas piores condições de *crosstalk*, onde todos os usuários compartilham o mesmo intervalo de frequência e fazem uso de máscaras espectrais fixas para transmissão. Este procedimento é conhecido como gerenciamento estático do espectro (SSM - “static spectrum management”). Entretanto, o *crosstalk* pode variar ao longo do tempo, devido, por exemplo, ao procedimento de liga/desliga dos modems, de modo que a utilização de máscaras espectrais fixas contribuem para uma utilização pouco eficiente do canal [3].

Portanto, a abordagem ideal para os projetos de sistemas de comunicação DSL é considerar uma transmissão conjunta multi-usuário e alocar o canal de comunicação adaptativamente, dependendo da interferência, sendo que esta alocação refere-se à potência e à taxa de transmissão de cada usuário. Tal abordagem é denominada de Gerenciamento Dinâmico do Espectro (DSM) [4].

A abordagem proposta pelo DSM é assegurar a compatibilidade espectral dos sistemas DSL de modo adaptativo, considerando assim as características particulares de cada rede de transmissão.

Algoritmos que implementam DSM são classificados em diferentes categorias de acordo com o grau de coordenação entre as múltiplas linhas DSL [5, 1]:

- DSM nível 0: corresponde a SSM, onde as técnicas DSM não são aplicadas;
- DSM nível 1: a PSD (Densidade Espectral de Potência - *Power Spectrum Density*) de cada linha é adaptada de maneira autônoma, por isso nenhuma forma de coordenação entre diferentes modems é permitida;
- DSM nível 2: uma alocação de potência é coordenada entre os multi-usuários por um centro de gerenciamento de espectro (SMC - “spectrum management center”);
- DSM nível 3: os sinais de duas ou mais linhas podem ser gerados simultaneamente e formam um vetor de sinais para transmissão, conhecido como transmissão multi-usuário vetorizada (este item não é discutido nesta Tese).

A partir da classificação dos algoritmos em níveis 1, 2 e 3, em alguns casos, não é possível definir a qual desses níveis um determinado algoritmo DSM pertence. Porém, além desses níveis, os algoritmos aqui serão classificados com autônomos ou centralizados.

Os algoritmos DSM podem ser projetados para três modos de operação [6]:

- Taxa Adaptativa (Maximiza a taxa de dados)
- Margem Adaptativa (usa toda potência disponível para manter uma taxa de dados fixa)
- Margem Fixa (usa somente a potência necessária para garantir um serviço de alta qualidade, ou seja, trabalha com a minimização de potência)

A maioria dos algoritmos existentes na literatura é formalizada levando em consideração o critério de maximização da taxa, no entanto, nenhum estudo comparativo entre os algoritmos DSM obtidos por diferentes modos de operação foi encontrado.

## 1.4 O Estado da Arte

Vários algoritmos DSM têm sido propostos na literatura, os quais têm sido denominados, em geral, de autônomos (os quais têm característica distribuída<sup>2</sup>) e centralizados (todo o processamento é realizado de maneira centralizada na SMC). Dentre eles destacam-se, os apresentados em [2, 7, 8, 9, 10, 11, 12]. Esse algoritmos resolvem um problema mono-objetivo, onde eles determinam um ponto de operação. Além disso, as soluções atuais tratam o problema como uma soma ponderada (*weighted-sum*), onde os pesos devem ser conhecidos pelo operador. Supondo a existência de  $N$  usuários, pode-se variar os  $N-1$  pesos e gerar uma região de taxa com vários pontos de operação, já que para cada combinações de pesos se tem um ponto de operação. Contudo esse mapeamento é não linear e por isso não é trivial encontrar um conjunto de pontos de operação com diversidade (conceito que será tratado no Capítulo 4). Por fim, na literatura a preocupação dos algoritmos propostos está em minimizar o efeito provocado pelo *crosstalk*.

Algoritmos evolucionários têm sido usados em diversas áreas, em especial pode-se tratar dos problemas de otimização mono-objetivo e multiobjetivos, tal como apresentado em [13]. Em sistema DSL o trabalho [14] apresenta um algoritmo de balanceamento de espectro usando otimização por enxame de partícula (PSO - *particle swarm optimization*). Esse algoritmo, também encontra um único ponto de operação e é dito ser ótimo e de baixa complexidade.

Quando um novo método para resolver problemas de busca e otimização é proposto, uma descrições visual pode ser adequada para demonstrar o desempenho do método. Porém, quando a metodologia se torna popular, diferentes implementações existem e torna-se necessário comparar o seu desempenho em vários problemas de teste. Esse procedimento é comum para algoritmos evolucionário multi-objetivo. Em especial, quando deseja-se avaliar se as soluções obtidas são ótimas de Pareto e se são soluções não-dominadas (os conceito de solução de Pareto e de não-dominancia serão abordados no capítulo 4), algumas métricas podem ser adotadas, tais como: *spacing*, *spread*, *maximum spread* [15]. Cada uma dessas métricas tem características específicas para a avaliar as soluções, por exemplo, a métrica *spacing* avalia o espaçamento entre as soluções, a métrica *spread* avalia se as soluções estão bem espalhadas dentro do espaço de soluções e *maximum spread* revela o valor do espaçamento das soluções extremas.

---

<sup>2</sup>O processamento é completamente dividido entre os vários modems da rede, dessa forma, se conhece apenas indiretamente a alocação de potência dos demais usuários através da manipulação da SNR medida na linha

## 1.5 Contribuição da Tese

Neste trabalho são apresentados dois métodos, um para determinar um conjunto de pontos de operação e o segundo é um método de avaliação da diversidade de soluções não-dominadas obtidas a partir do primeiro método.

O primeiro método proposto trata de encontrar um conjunto de pontos de operação, os quais são *ótimos de Pareto com diversidade*, para sistemas DSL, e com isso pretende-se fornecer um mecanismo para as Concessionárias Locais de Telefonia (STFC), as quais são os provedores do serviço DSL, realizarem o planejamento da oferta do serviço DSL, permitindo auxiliar as STFC na avaliação da capacidade e criação de um plano para melhorar o uso da rede. Esse método proposto utiliza um processo híbrido composto de um algoritmo evolucionário multiobjetivo (MOEA - *multi-objective evolutionary algorithm*), especificamente, por um algoritmo genético por ordenamento por não-dominância (NSGA-II - Non-Dominated Sorting Genetic Algorithm II) baseado em uma otimização multiobjetivo e pelo uso de um algoritmo de balanceamento de espectro, em que podem ser usados quase todos os algoritmos de balanceamento de espectro que usam a formulação de soma ponderada.

O método proposto para avaliar diversidade das soluções é baseado na geometria computacional, onde são determinados os desvios padrões das áreas das facetas compostas por pontos pertencentes as soluções, cujo o objetivo é avaliar a diversidade da soluções final.

O restante deste trabalho está organizado como segue: No Capítulo 2 são apresentados alguns algoritmos de balanceamento de espectro que podem ser utilizados no método proposto. O Capítulo 3 apresenta o conceito de algoritmos evolucionários e, é descrito o algoritmo genético, onde mais especificamente, também apresenta-se o algoritmo genético multiobjetivo, o NSGA-II, o qual será utilizado neste trabalho.

No Capítulo 4 será apresentado o método proposto, onde inicialmente discuti-se o conceito de diversidade, em seguida, descreve-se a formulação multiobjetivo que será o problema a ser resolvido pelo método e, também apresenta uma metodologia de análise de desempenho para avaliar as soluções do problema proposto. No Capítulo 5 são apresentados os resultados das simulações e no Capítulo 6 as conclusões do trabalho.

# Capítulo 2

## Algoritmos DSM

### 2.1 Introdução

Os modems DSL não devem usar mais potência de transmissão que o necessário para alcançar suas taxas de dados alvo, garantindo boa qualidade de serviço, e não podem usar largura de banda maior que a disponível para comunicação.

Com o objetivo de alcançar essa eficiência de transmissão, técnicas adaptativas de alocação de potência, conhecidas como gerenciamento dinâmico do espectro (DSM - *Dynamic Spectrum Management*), podem ser usadas para ajustar a densidade espectral de potência (PSD - *Power Spectral Density*) transmitida pelos modems DSL. A alocação das PSDs usando as técnicas de DSM dependerão das características do canal e do nível de *crosstalk* na linha.

Neste capítulo serão apresentados os principais algoritmos DSM publicados na literatura, os quais servirão de base para o desenvolvimento da tese de doutorado.

### 2.2 Modelo do Sistemas DSL

Para desenvolvimento de algoritmos DSM, um modelo padrão frequentemente usado considera que  $N$  assinantes xDSL empregam modulação DMT (Multi-ton Discreto - *Discrete Multitone*). O DMT é um sistema multi-portadora, cuja a idéia básica é particionar a largura de banda do canal em um conjunto de  $K$  subcanais que operam em paralelo. O benefício do particionamento do canal em um conjunto de subcanais independentes (idealmente) e paralelos é que a transmissão em regiões ruidosas ou de altas atenuações do canal podem ser evitadas. O particionamento da largura de banda do canal em subcanais pode ser feita usando qualquer função ortogonal porém, no DMT esse particionamento é feito usando a transformada de Fourier, particularmente a transformada discreta de Fourier inversa (IDFT - *inverse discrete Fourier Transform*). [16, 1].

O cenário multiusuário DSL é modelado como sendo constituído por subcanais independentes (tons), como segue:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k, \quad (2.1)$$

onde  $\mathbf{y}_k \triangleq [y_k^1, \dots, y_k^N]^T$  é o vetor ( $N \times 1$ ) do sinal recebido no tom  $k$ , tendo  $\mathbf{x}_k$  e  $\mathbf{z}_k$  uma estrutura similar;  $\mathbf{x}_k$  é o vetor do sinal transmitido no tom  $k$ ;  $\mathbf{z}_k$  é o vetor do ruído no tom  $k$ ; a PSD transmitida no tom  $k$  pelo usuário  $n$  é dada por  $s_k^n \triangleq E[|x_k^n|^2]$ ;  $\mathbf{H}_k$  é a matriz  $N \times N$  da função de transferência do canal obtida a partir da transformada de Fourier da resposta ao impulso do canal  $h(t)$ , onde  $h_k^{n,m} \triangleq [\mathbf{H}_k]_{n,m \in N}$  é o ganho do canal do transmissor  $m$  para receptor  $n$  no tom  $k$  e, os elementos da diagonal principal ( $n = m$ ) são os ganhos dos canais diretos, enquanto que os outros elementos representam os canais de *crosstalk*; e  $\mathbf{z}_k$  é o vetor do ruído aditivo no tom  $k$  e é composto pelo ruído térmico, *crosstalk alien* (provenientes de outros serviços), pela interferência de rádio frequência, etc. A PSD de ruído na linha  $n$  é  $\sigma_k^n \triangleq E[|z_k^n|^2]$ .

Pode-se pensar que, para simplificar a equalização, os sistemas de modulação multi-portadora possam considerar o particionamento da largura de banda do canal em um número grande o bastante de subcanais tal que, a resposta em frequência do subcanal seja aproximadamente plana. Se assim o fosse, o critério de Nyquist poderia ser satisfeito e a interferência intersimbólica (ISI - *Intersymbol Interference*) do símbolo nos símbolos subsequentes pode ser evitada. O receptor pode então demodular o sinal símbolo por símbolo, sem um equalizador complexo, como os usados nos sistemas de portadora única. Porém, em situações práticas, o modelo de um canal plano não é conseguido, e assim a ISI não é completamente eliminada pelo processo de particionamento do canal. Em vez disso, considerando que cada sub-canal se comportará como um canal AWGN (ruído gaussiano branco aditivo - *additive white gaussian noise*), e a ISI será combatida usando-se o prefixo cíclico. O prefixo cíclico é uma cópia das últimas amostras no domínio do tempo do símbolo DMT, de tal forma que a ISI fique confinada no prefixo cíclico, se o comprimento da resposta ao impulso do subcanal (em amostras) for menor ou igual ao tamanho do prefixo cíclico, para uma dada taxa de amostragem. No receptor as amostras correspondentes ao prefixo cíclico são descartadas, eliminando assim a ISI [1, 16].

Os canais diretos apresentam ganhos mais significativos que os canais de *crosstalk*, e como no ambiente DSL as condições de forte interferência não são duradouras, os esquemas de subtração dessas interferências são inaplicáveis e de alta complexidade computacional. Por essa razão, os sistemas DSL atuais tratam *crosstalk* como ruído [17]. Com todas essas considerações, o *bit loading* do usuário  $n$  no tom  $k$  é dado por:

$$b_k \triangleq \log_2 \left( 1 + \frac{1}{\Gamma} \frac{|h_k^{n,n}|^2 s_k^n}{\sum_{m \neq n} |h_k^{n,m}|^2 s_k^m + \sigma_k^n} \right), \quad (2.2)$$

onde  $\Gamma$  representa o *SNR-gap* ou somente *gap* e é função da probabilidade de erro alvo ( $Pe$ ) e do código de linha e geralmente é um valor constante para uma grande faixa de valores  $b_k$  [1].

O *bit loading* permite que o transmissor possa variar, adaptativamente, o número de bits

usados em cada tom, escolhendo um diferente tipo de constelação, usando as informações enviadas pelo receptor correspondentes às SNR (Relação-sinal ruído - *signal-to-noise ratio*), medidas em cada tom recebido, dada por:

$$SNR = \frac{1}{\Gamma} \frac{|h_k^{n,n}|^2 s_k^n}{\sum_{m \neq n} |h_k^{n,m}|^2 s_k^m + \sigma_k^n}. \quad (2.3)$$

A taxa de transmissão de dados na linha  $n$  é então

$$R_n = f_s \sum_k b_k^n, \quad (2.4)$$

onde  $f_s$  é taxa de símbolos DMT.

Cada modem  $n$  está sujeito a uma restrição de potência, dadas as suas limitações:

-  $\sum_k s_k^n \leq P_n, \forall n$ , onde  $P_n$  é a potência máxima do usuário  $n$ .

## 2.3 Formulação do Problema de Gerenciamento do Espectro

Os algoritmos DSM geralmente lidam com a maximização de taxa, sujeita a uma restrição de potência. O problema de gerenciamento de espectro para algoritmos DSM autônomos é formulado como:

$$\max_{s^1} R_1 \quad \text{tal que,} \quad \sum_{k \in K} s_k^1 \leq P_1, \quad (2.5)$$

$$\text{e } 0 \leq s_k^1 \leq s_k^{1,max}, \quad \forall k.$$

Onde o usuário a ser maximizado recebe o índice de 1,  $s^1$  é o vetor contendo as PSDs do usuário de interesse,  $s_k^{1,max}$  é o valor máximo da PSD do usuário de interesse no tom  $k$ .

Para algoritmos DSM centralizados, uma outra restrição é necessária,

$$R_n \geq R_n^{alvo} \quad \forall n > 1, \quad (2.6)$$

onde  $R_n^{alvo}$  representa a taxa de dados alvo (possíveis) para o usuário  $n$ .

Nas próximas seções serão apresentados os algoritmos autônomos e centralizados usados para solucionar o problema de gerenciamento de espectro no ambiente DSL.



## 2.4 Iterative Water-Filling

O algoritmo *iterative water-filling* (IWF) foi proposto por Wei Yu em 2002 [2]. A solução apresentada pelo IWF usa a maximização da taxa de bits de múltiplos modems, onde cada um otimiza sua própria PSD, considerando que o *crosstalk* proveniente dos outros modems é dado. Como os outros modems também mudarão suas PSDs, o *crosstalk* medido pelo primeiro modem não será constante o que provocará a mudança de sua PSD. Isso resulta em uma iteração dos modems pela aplicação do procedimento de *water-filling*, mas todo esse procedimento de gerência pelo algoritmo IWF ocorre de forma autônoma.

Através da medida de ruído na linha, cada usuário tentará evitar as bandas de frequência mais ruidosas e tenderá a alocar as bandas menos ruidosas, de modo que possam ser mais eficientemente usadas. Essa autonomia e sua baixa complexidade são as duas principais vantagens do IWF.

O IWF pode, potencialmente, aumentar a taxa de dados e o alcance das linhas DSL. Isto é possível porque cada modem pode determinar, independentemente dos outros modems, seu espectro de transmissão, usando as medidas do seu canal direto, do *crosstalk* e do ruído de fundo. O IWF não precisa de um centro de gerenciamento de espectro (SMC - *spectrum management center*), como acontece com os algoritmos centralizados [18].

O modelo apresentado pela Eq. (2.5) é tipicamente um problema de otimização e pode ser resolvido usando Lagrangeano. Em cada sub-canal  $k$ , a alocação de potência correspondente é encontrada pela equação de *water-filling*, onde os modems fazem alocação de suas PSDs de acordo com a relação ruído-canal em cada tom, como apresentado a seguir.

Para realizar a otimização, defini-se primeiro uma função objetivo que incorpore tanto a equação de maximização de taxa quanto a restrição de potência apresentado na Eq. (2.5), como mostrado por:

$$\mathcal{L}(s_k^1) = f_s \sum_k b_k^1 + \lambda_1 \left( P_1 - \sum_{k \in K} s_k^1 \right), \quad (2.7)$$

$$\mathcal{L}(s_k^1) = f_s \sum_k \log_2 \left( 1 + \frac{1}{\Gamma} \frac{|h_k^{1,1}|^2 s_k^1}{\sum_{m \neq 1} |h_k^{1,m}|^2 s_k^m + \sigma_k^1} \right) + \lambda_1 \left( P_1 - \sum_{k \in K} s_k^1 \right),$$

$$\mathcal{L}(s_k^1) = f_s \log_2 e \sum_k \log_e \left( 1 + \frac{1}{\Gamma} \frac{|h_k^{1,1}|^2 s_k^1}{\sum_{m \neq 1} |h_k^{1,m}|^2 s_k^m + \sigma_k^1} \right) + \lambda_1 \left( P_1 - \sum_{k \in K} s_k^1 \right),$$

onde  $\lambda_1$  é o multiplicador de Lagrange do usuário de interesse e pode ser interpretado como o “preço” pago pela violação da restrição. Os valores de  $\lambda$  são especificado pelas condições de Karush-Kuhn-Tucker (KKT). Portanto, derivando  $\mathcal{L}$  em relação a  $s_k^1$ , fixando depois o resultado como igual a zero e, finalmente, organizando os termos, obtemos a solução pode ser escrita como a equação de

*water-filling*,

$$\left[ s_k^n = \frac{\log_2 e}{\lambda_n} - N_n^k \right]^+, \quad (2.8)$$

onde  $[x]^+ \triangleq \max(0, x)$  e

$$N_n^k = \frac{\Gamma \sum_{m \neq n} |h_k^{n,m}|^2 s_k^m + \sigma_k^n}{|h_k^{n,n}|^2}. \quad (2.9)$$

O nível de *water-filling* ( $\lambda_1$ ) é escolhido, tal que os modems possam alcançar suas taxas-alvo ( $R^{alvo}$ ), as quais permitem ajustar suas PSDs de acordo com Eq. (2.8) de maneira iterativa (para ajustar sua relação ruído-canal  $N_1^k$ ) com os outros modems. O algoritmo escrito em [17] é apresentado no Apêndice A, na Tabela A.1.

## 2.5 Optimal Spectrum Balancing

O algoritmo *Optimal Spectrum Balancing* (OSB) foi proposto por Cendrillon et al, em 2004 em conferência e em 2006 em revista [19, 7]. Trata-se de um algoritmo centralizado que necessita que todas as PSD sejam calculadas em conjunto e em um centro de gerenciamento de espectro (SMC). Esse algoritmo atinge a melhor relação possível entre as taxa de dados dos diferentes modems na rede, permitindo operar em qualquer ponto na região de taxa.

Um passo fundamental, relativo à solução apresentada pelo OSB, é a análise da soma de taxas com pesos. Como matematicamente provado em [17], para um dado valor  $w$ , o problema de gerenciamento de espectro de potência, para o caso de dois usuários, é equivalente a

$$\max_{s_1, s_2} wR_1 + (1 - w)R_2. \quad (2.10)$$

Dado que a região de taxa é não-convexa [1], todos os pontos na região de taxa podem ser encontrados através da Eq. (2.10), onde  $w$  (valor entre 0 e 1) é um tipo de peso, ou prioridade, que é dado para o usuário 1. Variando  $w$ , será possível encontrar todos os pontos ótimos na fronteira da região de taxa. Para o caso de  $N$ -usuários, a soma dos  $w_n$  deverá sempre ser igual a um.

De posse da Eq. (2.10) e assumindo a restrição total de potência (Eq. 2.5), é possível escrever o Lagrangeano como

$$\mathcal{L} = wR_1 + (1 - w)R_2 - \lambda_1(P_1 - \sum_k s_k^1) - \lambda_2(P_2 - \sum_k s_k^2), \quad (2.11)$$

onde  $\lambda_n$  é um multiplicador de Lagrange para o usuário  $n$ .

Agora, definindo o Lagrangeano em um tom  $k$ ,

$$\mathcal{L}_k = wb_k^1 + (1 - w)b_k^2 - \lambda_1 s_k^1 - \lambda_2 s_k^2. \quad (2.12)$$

A expressão do Lagrangeano Eq. (2.11) pode ser reescrita através da soma  $\mathcal{L}_k$  em todos os tons:

$$\mathcal{L} = \sum_k \mathcal{L}_k + \lambda_1 P_1 + \lambda_2 P_2. \quad (2.13)$$

Assim, torna-se possível dividir o problema de otimização em  $K$  tons, os quais dependerão somente de  $w$ ,  $\lambda_1$  e  $\lambda_2$ . O objetivo é encontrar o par de PSDs  $(s_k^1, s_k^2)$  que maximize o Lagrangeano e os correspondentes  $\lambda_n$ 's, tal que a restrição de potência total seja atendida. O algoritmo OSB para esse caso é apresentado na Tabela A.2 do Apêndice A.

O OSB para  $N$  usuários é a extensão do algoritmo OSB para 2 usuários. O problema de gerenciamento do espectro apresentado na Eq. (2.10) é então reescrito usando a mesma aproximação para maximização das somas de taxas com pesos:

$$\max_{s_n, n \in N} \sum_n \omega_n R_n \quad \text{tal que,} \quad \sum_{k \in K} s_k^n \leq P_n, \quad \forall n, \quad (2.14)$$

$$\text{e } 0 \leq s_k^n \leq s_k^{n, \max}, \quad \forall k,$$

onde os pesos  $\omega_1, \dots, \omega_{N-1}$  são escolhidos, tal que a restrições de potência dos usuários  $1, \dots, N-1$  sejam respeitadas. Para resolver (2.14), o Lagrangeano é usado:

$$\max_{s_k^1, \dots, s_k^N} \mathcal{L}_k \quad \text{tal que } 0 \leq s_k^n \leq s_k^{n, \max}, \quad \forall n, \quad (2.15)$$

onde o Lagrangeano no tom  $k$  é definido como:

$$\mathcal{L}_k(s_k^1, \dots, s_k^N, \omega_1, \dots, \omega_N, \lambda_1, \dots, \lambda_{N-1}) = \sum_n \omega_n h_k^n(s_k^1, \dots, s_k^N) - \lambda_n s_k^n. \quad (2.16)$$

Para encontrar  $\lambda_n$  e  $\omega_n$  é usado o método do sub-gradiente, com a finalidade de reduzir a complexidade computacional com o aumento do número de usuários fornecida pelo método da Bisseção (usada no caso de 2-usuários). No Apêndice A é apresentado, na Tabela A.3, o algoritmo OSB para  $N$  usuários.

Essa solução, apesar de encontrar os pontos ótimos de operação na região de taxa, é muito complexa para ser implementada em aplicações comerciais. O algoritmo tem uma complexidade que aumenta exponencialmente com o número de usuários  $N$ , o que torna o problema computacionalmente intratável para mais de 5 ou 6 usuários.

Apesar disso, esse algoritmo é de grande utilidade para fornecer um limitante superior dos outros algoritmos DSM, sejam eles autônomos ou centralizados.

## 2.6 Iterative Spectrum Balancing

Uma aproximação alternativa para redução da complexidade computacional do método OSB, é apresentada pelo algoritmo ISB (Iterative Spectrum Balancing), proposto por Cendrillon e Moonen em 2005 [8], o que otimiza a PSD de maneira iterativa entre todos os usuários, tal que cada usuário é atualizado, enquanto que as PSDs de todos os outros usuários são mantidas fixas. O método ISB resulta em um algoritmo de baixa complexidade com desempenho próximo do ótimo. O procedimento de otimização é descrito por:

$$\max_{s_k^n} \mathcal{L}_k(s_k^1, \dots, s_k^N) \quad \text{tal que,} \quad 0 \leq s_k^n \leq s_k^{n,max}, \quad \forall n. \quad (2.17)$$

O sumário do algoritmo é apresentado no Apêndice A na Tabela A.4. A busca  $N$ -dimensional exaustiva usada no OSB pode ser substituída por uma busca unidimensional exaustiva, dado que a otimização das PSDs é realizada por um único usuário de cada vez. De maneira similar ao OSB, os multiplicadores Lagrangeanos  $\lambda_n$  e  $w_n$  são atualizados usando o método do sub-gradiente. Dessa forma a complexidade necessária para ajustar as PSDs é reduzida à escala quadrática em  $N$ .

O ISB resulta em um algoritmo próximo do ótimo e, ao contrário OSB, é computacionalmente tratável para grandes valores de  $N$  e realizável se uma SMC é disponível, considerando que o algoritmo requer que cada usuário precisa ter conhecimento das PSDs de ruído e dos canais de *crosstalk* de todos os modems do cenário. Essas informações tornam o ISB um algoritmo diferente do IWF por não ser autônomo, contudo o ISB proporciona um melhor desempenho do sistema. Experimentos mostram que o modem com o ISB pode alcançar o dobro da taxa de dados que modems com IWF [8].

## 2.7 Autonomous Spectrum Balancing

O Autonomous Spectrum Balancing (ASB), proposto por Cendrillon et al. em um simpósio em 2006 [20] e em 2007 foi publicado em revista [10], é um algoritmo para alocação de potência em canais DSL interferentes e seletivos em frequência.

A idéia básica é usar o conceito de linha de referência (linha virtual) para imitar uma linha vítima no canal de interferência.

Pela alocação de níveis de espectro de potência que proporcione a proteção das linhas de referência, um bom compromisso entre a maximização global (entre os modems) e autônoma pode ser alcançado.

O ASB alcança um bom compromisso entre modo de operação (autônomo), complexidade (linear com  $N$  e  $K$ ) e desempenho próximo do ótimo, quando comparado com os vários aspectos de diferentes algoritmos de alocação de potência (IWF, OSB e ISB).

A solução ótima do problema apresentado na Eq. (2.5) é obtida quando cada usuário escolhe

um nível de PSD que conduz a uma boa relação entre maximizar sua própria taxa e minimizar os danos causado nos outros usuário. Baseado nisso, os autores do ASB introduziram o conceito de “linha de referência” e, em vez de resolver o problema da Eq. (2.5), cada usuário tenta maximizar a taxa de dados da linha de referência, sujeito a sua taxa-alvo de dados e a uma restrição de potência. Assim, o problema apresentado na Eq. (2.5) pode ser reescrito como segue

$$\max_{s^n} R^{n,ref} \quad \text{tal que,} \quad R^n \geq R^{n,alvo}, \quad (2.18)$$

$$\text{tal que,} \quad \sum_{k \in K} s_k^n \leq P^{n,alvo},$$

onde  $K$  é o número total de tons e  $R^{n,ref}$  é a taxa da linha de referência do usuário  $n$ , definida como

$$R^{n,ref} = \sum_{k \in K} \tilde{b}_k^n = \sum_{k \in K} \log \left( 1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k} \right), \quad (2.19)$$

onde  $\tilde{s}_k^n$  é a PSD transmitida;  $\tilde{\alpha}_k^n$  e  $\tilde{\sigma}_k$  são os ganhos do canal de *crosstalk* e PSD de ruído, respectivamente, ambos normalizados com o ganho do canal direto  $|h_k^{n,n}|$ ;

Usando Lagrangeano para a restrição de taxa-alvo, no problema descrito pela Eq. (2.18), com um coeficiente de peso (variável dual)  $w^n$ , o problema torna-se:

$$\max_{s^n} w^n R^n + R^{n,ref} \quad \text{tal que} \quad \sum_{k \in K} s_k^n \leq P^n. \quad (2.20)$$

Os coeficientes de peso  $w^n$  necessitam ser ajustados para assegurar uma restrição de taxa .

O algoritmo ASB é dividido em uma versão básica (ASB-I) e uma variação (ASB-II), a última desfruta de uma menor complexidade e de prova de convergência [20].

O ASB-I é obtido usando a solução dual, decompondo o problema (2.20) em vários sub-problemas menores, um para cada tom, e definindo um  $\mathcal{L}_k^n$  (Lagrangeano) como a função objetivo do  $n$ -ésimo usuário no tom  $k$ ,

$$\max_{\lambda^n \geq 0} \sum_{k \in K} \max_{s_k^n} \mathcal{L}_k^n \quad \text{tal que,} \quad \sum_{k \in K} s_k^n \leq P^n, \quad (2.21)$$

onde

$$\mathcal{L}_k^n = w_n b_k^n + \tilde{b}_k^n - \lambda_n s_k^n. \quad (2.22)$$

A PSD ótima que maximiza  $\mathcal{L}_k^n$  pode ser encontrada resolvendo as condições de primeira ordem,  $\partial \mathcal{L}_k^n / \partial s_k^n = 0$ .

Os usuários atualizam  $\lambda_n$ , respeitando a restrição de potência, e atualizam  $w_n$ , para assegurar

a restrição de taxa-alvo. O algoritmo do ASB é mostrado no Apêndice A na Tabela A.5.

A solução de  $\partial \mathcal{L}_k^n / \partial s_k^n = 0$  é conseguida encontrando as raízes de uma equação cúbica. Então, para reduzir essa complexidade, é assumido que a linha de referência opera com alta SINR (relação sinal-interferência mais ruído - *signal to interference plus noise ratio*) sempre que está ativa: Se  $\tilde{s}_k^n > 0$ , então  $\tilde{s}_k^n \gg \tilde{\sigma}_k \gg \tilde{\alpha}_k^{n,m} s_k^n$  para qualquer realizável  $s_k^n$ .

A escolha ótima da PSD em ASB-II pode ser resumida com segue:

$$s_k^{n,II}(w^n, \lambda^n, s_k^{-n}) = \begin{cases} \left( \frac{w^n}{\lambda_n + \tilde{\alpha}_k^n / \tilde{\sigma}_k} - \sum_{m \neq n} \alpha_k^{n,m} s_k^m - \sigma_k^n \right)^+, & k \in \bar{K} \\ \left( \frac{w^n}{\lambda_n} - \sum_{m \neq n} \alpha_k^{n,m} s_k^m - \sigma_k^n \right)^+, & k \in \bar{K}^C \end{cases} \quad (2.23)$$

onde  $\bar{K} \triangleq \{k | \tilde{s}_k > 0, k \in K\}$  representa os tons onde a linha de referência está ativa e  $\bar{K}^C \triangleq \{k | \tilde{s}_k = 0, k \in K\}$  representa os tons onde a linha de referência está inativa.

Essa é essencialmente uma solução do tipo water-filling, com diferentes níveis d'água para diferentes tons, que foi chamado de *water-filling seletivo em frequência* [20].

### 2.7.1 Successive Convex Approximation for Low-complexity (SCALE)

Em [9], foi proposto um novo algoritmo, que através de técnicas de relaxamento e aproximações convexas, transformou o problema da Eq. (2.5) em um problema convexo. Esse algoritmo ficou conhecido como SCALE, e apresenta resultados ótimos com complexidade comparável com a do IWF.

Esse algoritmo é proposto em uma forma semi-distribuída, sendo necessária uma troca de mensagem entre os modems e a SMC. Caso essa troca de mensagem não esteja disponível, o algoritmo converge para a solução do IWF.

A equação de atualização das PSDs é dada através da expressão

$$p_n^k = \frac{\omega_n \alpha_n^k}{\lambda_n + \Phi_n^k}, \quad (2.24)$$

onde  $\alpha_n^k = \frac{SNR_n^k}{1 + SNR_n^k}$  e  $\Phi_n^k$  é a mensagem recebida pelo usuário  $n$  da SMC, que é definida como

$$\Phi_n^k = \sum_{m \neq n} |h_{n,m}^k|^2 \Psi_m^k. \quad (2.25)$$

Os termos  $\Psi_m^k$  são as mensagens enviadas dos modems  $n \neq m$  para a SMC, que são calculados através da equação [9]

$$\Psi_m^k = \omega_m \alpha_m^k \frac{SNR_m^k(\mathbf{P}^k)}{|h_{m,m}^k|^2 p_m^k}. \quad (2.26)$$

O SCALE é uma das formas mais eficiente de resolver o problema (2.5), em termos de desempenho e complexidade. Entretanto, ainda é necessário um agente central para coordenar as

trocas de mensagens bem como o conhecimento completo das funções de transferência de FEXT.

## 2.8 Considerações Finais

A Tabela 2.1 apresenta um resumo comparativo dos principais aspectos dos algoritmos DSM apresentados. Através do exposto na tabela, o algoritmo ASB pode ser o preferível em relação aos demais, pois tem baixa complexidade, é quase ótimo e autônomo, esta última característica é uma das grandes vantagens do ASB pois, a SMC (necessária para os algoritmos centralizados) ainda não existe na prática. O termo *distribuído* referente a coluna *Processamento* da Tabela 2.1 significa que o algoritmo precisa de uma SMC, mas a mesma não é responsável por todo o processamento, como apresentado na subseção 2.7.1 referente ao algoritmo SCALE.

Tabela 2.1: Comparação entre os principais aspectos dos algoritmos de DSM.

	Complexidade	Desempenho	Processamento
IWF	$O(NK)$	sub-ótimo	autônomo
ASB	$O(NK)$	quase ótimo	autônomo
ISB	$O(KN^2)$	quase ótimo	centralizado
OSB	$O(Ke^N)$	ótimo	centralizado
SCALE	$O(NK \log K)$	quase ótimo	distribuído

## Capítulo 3

# Algoritmo Evolucionário para Otimização Multiobjetivo

Neste capítulo serão apresentados os conceitos de algoritmos evolucionários, em seguida, os conceitos relacionados ao algoritmo genético. Este capítulo é finalizado com a apresentação do algoritmo genético multiobjetivo (NSGA-II), o qual será utilizado neste trabalho.

### 3.1 Algoritmo Evolucionário

Os algoritmos evolucionários resolvem problemas usando modelos computacionais baseados nos princípios da evolução natural. Existem vários modelos computacionais propostos, porém todos têm em comum os conceitos da evolução da espécie: seleção, mutação e reprodução. Esses conceitos são denominados de *operadores genéticos*, que são formas computacionais de simular os fenômenos físicos da natureza, os quais dependem do desempenho do indivíduo dentro do ambiente [21].

Os algoritmos evolucionários funcionam mantendo uma população de indivíduos ou cromossomos<sup>1</sup>, os quais se comportam de maneira semelhante com a evolução da espécie. Em seguida, são aplicados *operadores genéticos*, tal como a mutação. Cada indivíduo da população sofre *avaliação* que quantifica a sua qualidade para a solução do problema. De acordo com a *avaliação* são aplicados *operadores genéticos* que garantem a sobrevivência dos mais aptos [21].

Uma das principais diferenças para os métodos clássicos de busca e algoritmos de otimização é que os algoritmos evolucionários usam uma população de soluções em cada iteração, em vez de somente uma solução, o que permite que sejam encontradas múltiplas soluções em uma execução. Se um problema de otimização tem somente um ótimo, espera-se que todos os indivíduos da população converjam para essa solução ótima. Porém, se um problema de otimização tem múltiplas soluções ótimas, o algoritmo evolucionário pode ser usado para encontrar múltiplas soluções ótimas [15].

---

<sup>1</sup>Os dois termos, indivíduos e cromossomos, serão usados alternadamente no texto.



## 3.2 Algoritmo Genético

O algoritmo genético (AG) foi apresentado por Jonh Holland em 1975, e é muito usado como ferramenta para realizar busca e otimização para vários tipos de problemas, incluindo a área de engenharia, baseado nos mecanismos de seleção natural e genética.

Uma característica importante do AG é que como ele foi projetado de acordo com os princípios da seleção natural, em problemas de otimização com muitos máximos locais, a busca por uma solução melhor ou por um máximo global continua, pois o AG não para a busca por um outro indivíduo melhor.

O AG é um método onde uma população aleatória de cromossomos é criada e submetida aos operadores genéticos de seleção, cruzamento (ou recombinação) e mutação [15, 21]. No fluxograma da Fig. 3.1, pode-se observar as etapas do AG: - antes de cada população ser submetida aos operadores genéticos, a mesma passa por um processo de avaliação onde os cromossomos são avaliados por uma *função de avaliação* para determinar suas aptidões (ou *fitness*) para a solução do problema. A função de avaliação pode ser função do valor das soluções das funções objetivos, mas muitas vezes a aptidão é o próprio valor da função objetivo (como será abordado neste trabalho); o operador de *seleção* escolhe os cromossomos que irão se reproduzir, e uma *avaliação* dos cromossomos totais é realizada para gerar a nova geração; a partir da nova geração, o operador de cruzamento troca parte de dois cromossomos e espera-se que melhores indivíduos sejam criados; a mutação muda de maneira aleatória os valores (alelos) de alguma parte do cromossomo na esperança de serem criados outros melhores. Uma vez que nenhuma destas operações (seleção, cruzamento e mutação) são realizadas de forma determinística, estas alegações não são garantidas, nem explicitamente testadas, durante uma geração AG [22, 15].

### 3.2.1 Elementos do Algoritmo Genético

Os algoritmos genéticos têm, pelo menos os seguintes elementos: população de indivíduos ou cromossomos, seleção, cruzamento e mutação, os quais são detalhados na sequência.

#### População de Indivíduos

Cada indivíduo ou cromossomo de uma população é uma representação codificada de uma possível solução do problema que se deseja resolver. Cada cromossomo consiste de “gene” (por analogia com as partes fundamentais que compõe o cromossomo em biologia), que são as características, por exemplo, bit. Cada gene pode assumir um determinado *valor* dentre os possíveis valores, o que é chamado de “alelo”, por exemplo no caso do cromossomo ser bit o alelo será '0' ou '1' [21].

O AG trabalha com uma forma codificada dos parâmetros a serem otimizados e não com os parâmetros propriamente ditos. As representações mais utilizadas para os indivíduos em AGs

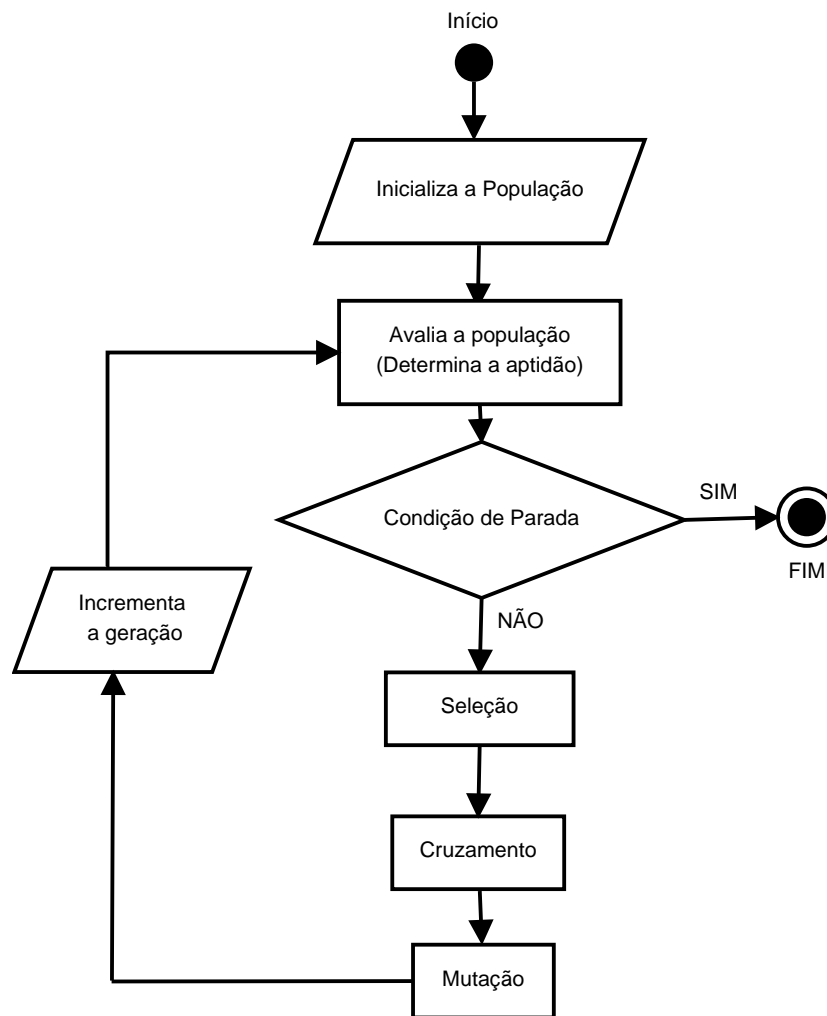


Fig. 3.1: Fluxograma do princípio de funcionamento do AG.

são as codificações binárias e reais. Na codificação binária, são utilizados valores binários para representar os indivíduos. Essa codificação é empregada, principalmente, quando os problemas envolvem variáveis discretas. Por outro lado, se as variáveis são contínuas, o processo de conversão é necessário. Esse processo de conversão por ser complexo, dependendo do número de bits envolvidos. E, se uma quantidade grande de bits é usada no processo de conversão, apesar de melhorar a precisão dos resultados obtidos pelo AG, o processo de conversão torna-se mais complexo [21]. Por isso, para casos que envolvem variáveis reais, como os que serão apresentados neste trabalho, a codificação real é mais indicada, o que dispensa o processo de conversão.

### Seleção

O método de seleção corresponde em selecionar os melhores indivíduos da população para gerar os descendentes para a geração seguinte. Neste contexto, o processo de seleção tenta copiar as soluções mais aptas e eliminar as soluções com menos aptidão, mantendo o tamanho da população

constante. A seleção é realizada através dos seguintes passos [15]:

1. Identificar as soluções mais aptas (boas soluções) em uma população;
2. Realizar múltiplas cópias das soluções mais aptas;
3. Eliminar soluções de baixa aptidão (soluções ruins), permitindo que várias cópias das soluções mais aptas possam ser inseridas na população.

Depois da realização dos passos acima, as melhores soluções serão submetidas aos operadores genéticos de cruzamento e mutação.

Existem várias maneiras de realizar a seleção, onde os métodos mais comuns são a seleção por torneio, seleção proporcional e seleção por *ranking*.

Neste trabalho foi usado o método de torneio. Nesse método um torneio é jogado entre as duas possíveis soluções (indivíduos) e a melhor solução é escolhida e adicionada à lista de soluções. As duas soluções retornam para a população e podem ser selecionadas novamente. Outras duas soluções são selecionadas e a melhor é adicionada à lista de soluções. Esse procedimento segue, mas cada solução só pode participar de dois torneios. A melhor de todas as soluções ganhará nos dois torneios e terá duas cópias na nova população. De maneira similar, a pior solução perderá os dois torneios e será eliminada da população. Dessa maneira, cada indivíduo terá zero, uma ou duas cópias na nova população [15].

A seleção por torneio tem melhor ou equivalente propriedades de convergência e complexidade computacional quando comparada com outros operadores de seleção existentes na literatura [15].

Para exemplificar o método de seleção por torneio, a Fig. 3.2 mostra quatro diferentes torneios jogados entre os membros da população (cada um joga duas vezes). Em cada torneio, os participantes são escolhidos aleatoriamente. Observe que o melhor indivíduo sempre ganha em cada torneio e forma a lista de soluções.

### **Cruzamento**

O operador de cruzamento faz parte do mecanismo de reprodução do AG utilizado para criar novas soluções a partir de outras existentes. O cruzamento é aplicado à população de indivíduos resultante do processo de seleção.

Como na *seleção*, existem vários operadores de cruzamento na literatura de AG, na maioria deles dois indivíduos de uma geração são escolhidos aleatoriamente e novos indivíduos são criados a partir da troca do material genético. Os descendentes serão diferentes de seus pais, mas com características genéticas de ambos os genitores.

Um método simples de cruzamento, aplicado para AG com codificação binária, é realizado entre dois indivíduos, onde é feita uma escolha, aleatória, de **um ponto** de cruzamento, onde a

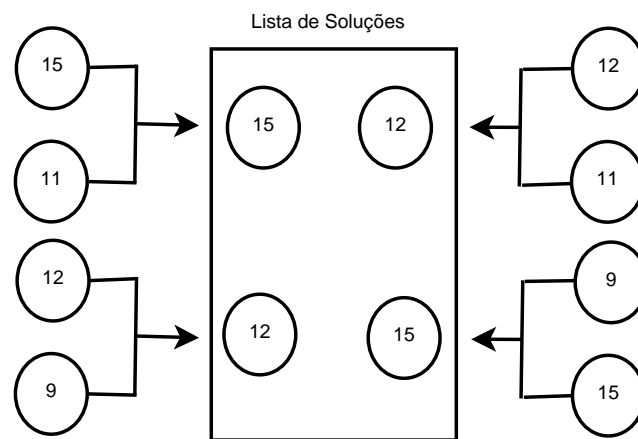


Fig. 3.2: Exemplo de torneio entre 4 jogadores. Os elementos de dentro do retângulo são as soluções que têm maior aptidão em cada comparação e formam a nova população de indivíduos.

partir desse ponto ambos os indivíduos são comutados e é feita uma permuta de valores, como exemplificado na Fig. 3.3 [15, 21].

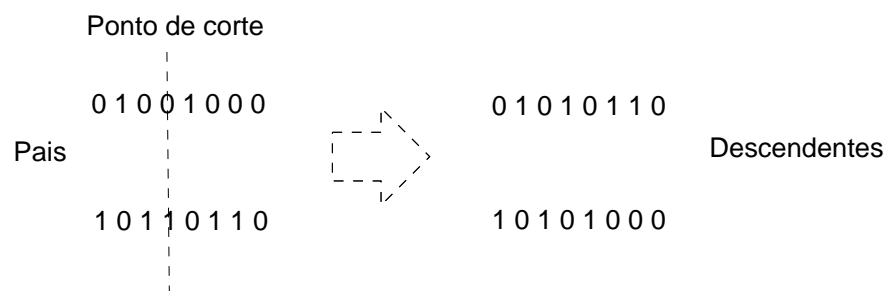


Fig. 3.3: Operador de cruzamento de um ponto.

Neste trabalho o processo de codificação é real e a técnica de cruzamento adotada é o SBX (*Simulated Binary Crossover*), que trabalha com o cruzamento de dois pais e a partir deles são gerados dois filhos. O operador SBX tem seu princípio de funcionamento é baseado no *cruzamento de um ponto* usado na codificação binária [23].

No SBX os filhos são criados a partir de uma distribuição de probabilidade que depende das localizações dos pais. Um fator de espalhamento ( $\beta_i$ ) é definido como a razão da diferença absoluta entre os cromossomos filhos e os pais, e é dada por

$$\beta_i = \left| \frac{f_1 - f_2}{p_1 - p_2} \right|, \quad (3.1)$$

onde,  $f_1$  e  $f_2$  são os descendentes (os filhos) e  $p_1$  e  $p_2$  são os pais.

Os filhos são criados de acordo com uma distribuição de probabilidade, a qual é demonstrada em [23] que depende da localização dos pais e é dada por:

$$\mathcal{P}(\beta_i) = \begin{cases} 0,5(\eta_c + 1)\beta_i^{\eta_c}, & \text{se } \beta_i \leq 1; \\ 0,5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{se } \beta_i > 1, \end{cases} \quad (3.2)$$

onde  $\eta_c$  é um número real não-negativo. Grandes valores de  $\eta_c$  implicam em altas probabilidades dos filhos serem gerados próximos aos pais, por outro lado, valores pequenos de  $\eta_c$  indicam que os filhos são gerados distantes dos pais, com alta probabilidade. Em geral, uma faixa apropriada para  $\eta_c$  está entre 2 e 5 [23].

Calculando a distribuição de probabilidade associada à função densidade dada pela Eq. (3.2) e igualando a  $u_i$ , onde  $u_i$  é um número aleatório entre 0 e 1, tem-se  $\beta_{qi}$  dada por:

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{se } u_i \leq 0,5; \\ \left[ \frac{1}{2(1-u_i)} \right]^{\frac{1}{\eta_c+1}}, & \text{se } u_i > 0,5. \end{cases} \quad (3.3)$$

Depois de obter  $\beta_{qi}$  da distribuição de probabilidade, os filhos são calculados por

$$\begin{aligned} f_1 &= 0,5 [(1 + \beta_{qi})p_1 + (1 - \beta_{qi})p_2], \\ f_2 &= 0,5 [(1 - \beta_{qi})p_1 + (1 + \beta_{qi})p_2]. \end{aligned} \quad (3.4)$$

Assim, o procedimento para calcular filhos ( $f_1$  e  $f_2$ ) a partir dos pais ( $p_1$  e  $p_2$ ) é:

1. Escolha um número aleatório  $u_i \in [0, 1]$ ;
2. Calcule  $\beta_{qi}$  usando a Eq. (3.3);
3. Calcule os filhos usando Eq. (3.4).

Um aspecto relevante desse operador de cruzamento é que para um valor fixo de  $\eta_c$ , os filhos tem um espalhamento que é proporcional ao dos pais, o que matematicamente pode ser observado pela manipulação da Eq. (3.4), e resulta em

$$(f_2 - f_1) = \beta_{qi}(p_2 - p_1). \quad (3.5)$$

Essencialmente o operador de cruzamento SBX tem duas propriedades:

1. A diferença entre os filhos é proporcional à dos pais;
2. As soluções para os filhos são mais prováveis de serem escolhidas próximas as dos pais do que distantes deles.

De maneira geral, o operador de cruzamento tem a capacidade de criar novos indivíduos que exploram o espaço de soluções do problema. Quando os novos indivíduos são criados, é possível

que os mesmos possam ser qualitativamente inferiores que aos pais, e por isso, o processo de seleção fica encarregado de removê-los ao longo das gerações, novas e melhores soluções persistirão no lugar das soluções de baixa aptidão garantindo um bom desempenho no processo de busca. Porém, é importante que os cromossomos filhos sejam diferentes dos pais para que seja preservada a diversidade das populações, isto é, o quão diferentes os indivíduos são uns dos outros. A diversidade pode impedir a convergência prematura a um ótimo local.

### Mutação

Assim como o operador de cruzamento, o operador de mutação é também responsável pelo aspecto de busca da solução pelo AG. A mutação é um operador que tem por objetivo aumentar a diversidade na população, assim como o cruzamento. Esse operador impede que o algoritmo tenha convergência prematura, evitando que a busca pelas soluções ótimas fique presa em regiões com soluções de mínimos ou máximos locais (dependendo da aplicação) [15].

No modo binário, o operador de mutação bit-a-bit muda o bit 1 por bit 0, e vice-versa, com uma probabilidade de mutação. Cada gene de um cromossomo recebe um valor de probabilidade (de maneira geral essa probabilidade assume pequenos valores) o qual irá determinar se o bit será modificado ou não, em seguida, é sorteado um número entre 0 e 1, se o valor sorteado for menor que a probabilidade, então a mutação é realizada. Como os valores de probabilidade são pequenos, esse processo resulta em pequenas alterações no cromossomo.

Já na codificação real, para se obter pequenas alterações no cromossomo, os procedimentos de mutação usam desvios que seguem distribuições probabilísticas variadas. Neste trabalho, o operador de mutação adotado foi a *mutação polinomial* e, assim como o operador SBX, usa uma distribuição polinomial de probabilidade. Para realizar a mutação é definido um *fator de perturbação*  $\delta$  definido por [24]:

$$\delta = \frac{f_i - p_i}{p_i^S - p_i^I}, \quad (3.6)$$

onde,  $p_i^S$  e  $p_i^I$  representa o limite superior e inferior do espaço de soluções do problema, respectivamente,  $f_i$  é o cromossomo resultante do processo mutação obtido a partir do cromossomo original  $p_i$ .

O valor da mutação é calculada com uma distribuição de probabilidade que depende do fator de perturbação  $\delta$ , dada por:

$$\mathcal{P}(\delta_i) = 0,5(\eta_m + 1)(1 - |\delta|)^{\eta_m}, \quad (3.7)$$

onde,  $\eta_m$  é o índice de distribuição, o qual controla o formato da distribuição.

A distribuição de probabilidade acima é válida para  $\delta \in [-1, 1]$ . Para criar um valor mutado, um número aleatório  $u_i$  é criado na faixa de  $[0, 1]$ . Assim, pode ser calculado o fator de perturbação  $\delta$  correspondente a  $u_i$  usando a distribuição de probabilidade acima:

$$\bar{\delta} = \begin{cases} (2u_i)^{\frac{1}{\eta_m+1}} - 1, & \text{se } u_i < 0,5; \\ 1 - [2(1 - u_i)]^{\frac{1}{\eta_m+1}}, & \text{se } u_i \geq 0,5. \end{cases} \quad (3.8)$$

Por fim, o valor mutado é calculado por:

$$f_i = p_i + (p_i^S - p_i^I)\bar{\delta}_i, \quad (3.9)$$

### 3.2.2 Parâmetros do AG

Existem alguns parâmetros, os quais, implicam no funcionamento do AG, tais como, o tamanho da população e as probabilidades de cruzamento e mutação.

#### Tamanho da População

O tamanho da população estabelece o número de indivíduos em uma população, e por isso, afeta o desempenho dos AGs, pois define a quantidade de chamadas à função objetivo por geração do AG.

Uma população muito grande permitiria uma maior exploração do espaço de soluções, porém com uma convergência mais lenta. Uma população muito pequena possibilitaria uma pequena cobertura do espaço busca, e por isso, corre-se o risco do algoritmo convergir prematuramente e não atingir o resultado esperado [15].

#### Probabilidade de Cruzamento

A probabilidade de cruzamento afeta a velocidade com que novas estruturas são introduzidas na população. Quanto maior for essa probabilidade, mais rapidamente novas estruturas serão introduzidas no processo de busca e, também, poderá aumentar a perturbação dos bons cromossomos, ocasionando suas perdas [25]. Com um valor baixo de probabilidade de cruzamento, o algoritmo pode tornar-se muito lento. Geralmente, a probabilidade de cruzamento é configurada antes da execução do AG e mantida constante ao longo das gerações, porém algumas implementações usam métodos dinâmicos para ajuste desse parâmetro [26].

#### Probabilidade de Mutação

A probabilidade de mutação estabelece qual a taxa em que uma mutação ocorrerá no indivíduo. Com probabilidade altas de mutação, a busca tende a se transformar em uma busca estritamente aleatória, possibilitando que boas soluções sejam perdidas [25]. Por outro lado, com valores muito baixos, a possibilidade de bons indivíduos serem introduzidos na população diminui.

De maneira geral, esses parâmetros genéticos são determinados de acordo com o problema em questão, ou seja, de forma empírica. Porém, existem estudos sobre mecanismos adaptativos e autoadaptativos, os quais não serão abordados nesta tese.

### 3.3 Elitist Nondominated Sorting Genetic Algorithm - NSGA-II

Elitist Nondominated Sorting Genetic Algorithm (NSGA-II) é um algoritmo evolucionário multiobjetivo <sup>2</sup> (multi-objective evolutionary algorithm - MOEA) baseado em AG, capaz de encontrar soluções bem espalhadas (com boa diversidade).

São características do NSGA-II [27]:

- Usa o conceito de ordenamento por não dominância em AGs, através da comparação dos indivíduos das soluções, gerando uma classificação dos mesmos;
- Antes de realizar cruzamento e mutação, usa o operador de seleção por torneio de multidão para preservar a diversidade entre as soluções, de forma a proporcionar um bom espalhamento das soluções;
- O NSGA-II usa um mecanismo de elitismo para preservar as melhores soluções encontradas, pois é realizada a comparação da população atual com as melhores soluções encontradas anteriormente.

De maneira geral o NSGA-II, inicia com uma população aleatória, onde terá seus elementos classificados, e os pais serão selecionados, os quais geram os filhos a partir dos operadores de cruzamento SBX e a mutação polinomial. Por fim, a população atual juntamente com os filhos gerados são unidos e passam por um processo de seleção, onde os melhores indivíduos são selecionados para formar a população final. Este processo se repete até alcançar o número de gerações proposto.

---

<sup>2</sup>Em problemas de otimização com um objetivo ou mono-objetivo, a meta é achar uma única solução ótima global. Um problema multiobjetivo a otimização trata de um conjunto de soluções.



# Capítulo 4

## Geração de Pontos de Operação com Diversidade usando Algoritmo Evolucionário

### 4.1 Introdução

Existem vários algoritmos de balanceamento de espectro, na literatura, tais com os apresentados em [2, 8, 9, 28, 29], que são usados para minimizar os efeitos que degradam a transmissão de dados através de pares trançados. Esses algoritmos, em geral, utilizam métodos que otimizam as densidades espectrais de potência dos modems através de uma otimização mono-objetivo, ou seja, calculam apenas um ponto de operação específico. Neste trabalho é apresentado um método para encontrar um conjunto de pontos de operação para sistemas DSL. A solução que será apresentada poderá ser útil para o planejamento de redes DSL para, por exemplo, ajudar as operadoras da rede a avaliarem a capacidade do sistema DSL e criar um plano para melhorar o uso do sistema.

Os conjuntos de pontos de operação que serão encontrados com o método proposto são denominados de soluções *não dominadas* ou soluções de *Pareto* (se uma solução  $x_1$  domina uma outra  $x_2$  significa que  $x_1$  é melhor que  $x_2$  ou que  $x_1$  é não dominada por  $x_2$ ) e são pontos distribuídos com *diversidade* [15]. Uma solução com diversidade ou solução diversa é aquela que tem um conjunto que apresenta uma variedade de valores bem diversa, dentro dos limites possíveis. A diversidade é importante porque, em casos extremos, todos os elementos de uma dada solução poderiam ser constituída de um conjunto de pontos não dominados e muito próximos um do outro. E, para o problema proposto, essa solução não seria um bom resultado, como no exemplo apresentado na Fig. 4.1(a), onde os pontos não estão bem distribuídos, ou seja, apresentam baixa diversidade, comparada com a diversidade dos pontos apresentados no exemplo da Fig. 4.1(b), que podem ser considerados como um conjunto com boa diversidade<sup>1</sup>.

---

<sup>1</sup>Os valores dos pontos apresentados nas Fig. 4.1(a) e Fig. 4.1(b) foram criados por método de tentativa e erro.

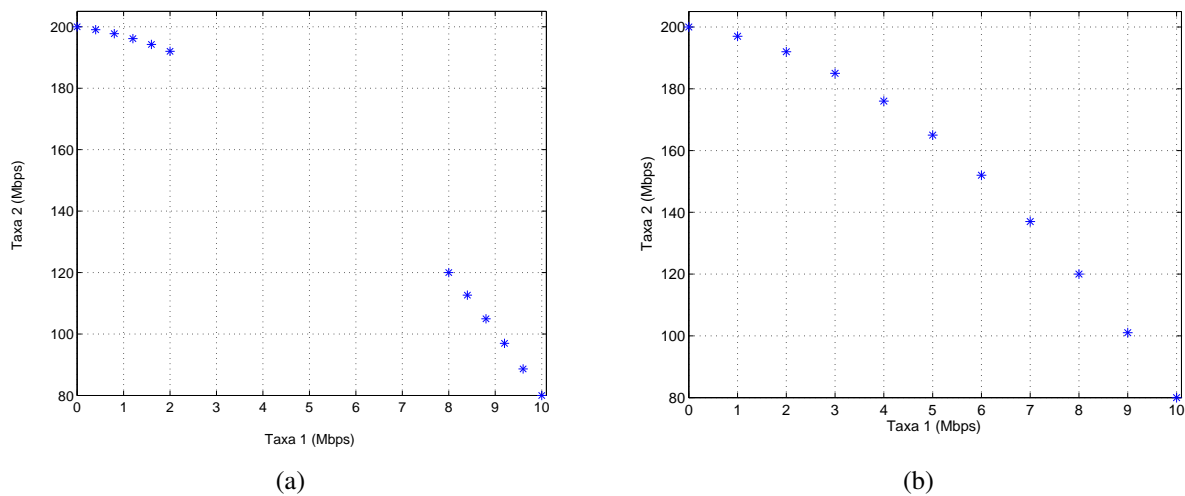


Fig. 4.1: Exemplo de dois conjuntos de solução para exemplificar diversidade, sendo (a) solução não dominada com pouca diversidade (b) solução não dominada com boa diversidade.

O problema em encontrar uma solução com diversidade será chamado de balanceamento de espectro com solução diversa (DSSB - *diverse set spectrum balancing*). A solução para DSSB é implementada com um algoritmo genético (AG) evolucionário e baseada em uma otimização multi-objetivo [15].

As próximas seções deste capítulo seguem a seguinte organização, a Seção 4.2 apresenta a formulação multiobjetivo e fornece um visão geral do problema de DSSB e na Seção 4.3 é apresentado o método proposto para solucionar o problema DSSB.

## 4.2 Descrição do Problema

Nesta seção será descrito o problema DSSB. Primeiro, será redefinido o problema de balanceamento de espectro de acordo com a formulação multiobjetivo. Essa formulação será derivada da otimização mono-objetivo (ou de objetivo único) a qual pode-se encontrar em diversos artigos na literatura, tais como, os apresentados em [2, 8, 9, 28, 29].

### 4.2.1 Balanceamento de Espectro Multiobjetivo

Para desenvolver a solução para o problema DSSB admite-se que a modulação usada para DSL é a DMT (multi-ton discreto - *discrete multitone*) [1]. A densidade espectral de potência (PSD) de todos os usuários é organizada em uma matriz  $\mathbf{S} = \{s_k^n\}$  de dimensão  $N \times K$ , onde  $N$  é o número de usuários,  $K$  é o número de subcanais (tons) e cada elemento  $s_k^n$  representa a PSD do usuário  $n$  no tom  $k$ . Os ganhos dos canais (direto ou de *crosstalk*) são representados por uma matriz tridimensional  $\bar{\mathbf{H}}$  com dimensão  $N \times N \times K$ . Para cada tom  $k$ , a matriz  $\{h_{i,j}^k\}$  corresponde

a uma função de transferência do canal do usuário  $i$  para  $j$ . Para determinar  $\bar{\mathbf{H}}$  em cenários práticos, existem procedimentos para estimar o canal direto  $h_{i,i}^k, \forall k$  referente ao  $i$ -ésimo usuário [30, 31] e o canal FEXT (*far-end crosstalk*)  $h_{i,j}^k, \forall k, i \neq j$  [32]. Assume-se que é usado o sistema de duplexação por divisão de frequência (FDD), onde as transmissões em *downstream* e *upstream* não se sobrepõem, pois estão em bandas diferentes e, dessa maneira, o NEXT (*near-end crosstalk*) pode ser omitido [1].

Outras fontes de ruído, tais com, ruído de fundo, interferência das radiofrequências [16] e *alien crosstalk*, são descritas por uma matriz  $\mathbf{N} = \{\sigma_k^n\}$  de dimensão  $N \times K$ . O elemento  $\sigma_k^n$  é o valor da PSD de ruído no receptor  $n$  e no tom  $k$ .

Então, as informações de um canal (IC) são descritas por  $(\mathbf{N}, \bar{\mathbf{H}})$ . Um *ponto de operação*  $(\mathbf{S}, \mathbf{N}, \bar{\mathbf{H}})$  é completamente descrito pelas PSDs de todos os usuários  $\mathbf{S}$  e pelas ICs de uma dada rede. Dado  $(\mathbf{S}, \mathbf{N}, \bar{\mathbf{H}})$  pode-se calcular a taxa  $r_n = \mathcal{R}(\mathbf{S}, \mathbf{N}, \bar{\mathbf{H}})$  e a potência total transmitida  $p_n = \mathcal{P}(\mathbf{S}, \mathbf{N}, \bar{\mathbf{H}})$  dos  $n$ -ésimo usuário, onde  $\mathcal{R}$  e  $\mathcal{P}$  são as funções para cálculo das taxas e das potências totais, respectivamente. Por exemplo, a taxa de bit do  $n$ -ésimo usuário é

$$r_n = f_s \sum_{k=1}^K \log_2 \left( 1 + \frac{h_{n,n}^k s_n^k}{\sigma_n^k + \sum_{\substack{m=1 \\ m \neq n}}^N h_{m,n}^k s_m^k} \right) \quad (4.1)$$

bits por segundo (bps), onde  $f_s$  é a taxa de símbolo (tipicamente  $f_s = 4$  kHz nos sistemas DSL).

Será usado, por conveniência, a notação  $\bar{\mathbf{S}}_Z = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_Z\}$  o que representa um conjunto de  $Z$  matrizes de PSDs de dimensão  $N \times K$ , onde cada matriz representa um ponto de operação para os modems. Com a representação  $\bar{\mathbf{S}}_\infty$  significa que  $Z \rightarrow \infty$  ou  $Z$  é desconhecido.

O problema de *balanceamento de espectro multiobjetivo* é um pouco diferente do problema convencional de balanceamento de espectro mono-objetivo, apresentado em [6]. A versão multiobjetivo corresponde a uma busca a um conjunto de PSDs ótimas de Pareto  $\bar{\mathbf{S}}_\infty^\dagger$  de acordo com:

$$\begin{aligned} \bar{\mathbf{S}}_\infty^\dagger = \arg \max_{\mathbf{S}} \quad & \mathcal{O}_i(\mathbf{S}, \mathbf{N}, \bar{\mathbf{H}}), \quad i = 1, \dots, L \\ \text{tal que} \quad & q_j(\mathbf{S}) \leq 0, \quad j = 1, \dots, Q \\ & t_m(\mathbf{S}) = 0, \quad m = 1, \dots, T \\ & 0 \leq s_k^n \leq M_k, \quad k = 1, \dots, K \end{aligned} \quad (4.2)$$

onde  $\mathcal{O}_i$  (os argumentos são omitidos por simplicidade) é a  $i$ -ésima função objetivo. A função  $q_j$  e  $t_m$  são, respectivamente, as restrições de desigualdade e igualdade impostas a solução (por exemplo, para permitir a imposição da potência máxima ou da mínima taxa de bits por usuário). O valor  $M_k$  corresponde a mascarar PSD limite, tipicamente imposta pelo padrão DSL e pelas limitações de hardware.

Cada função objetivo  $\mathcal{O}_i$  poderia ser eventualmente maximizada (no caso de maximização de taxa de dados -  $R_n$ ) ou minimizada (para o caso de minimização potência total transmitida), onde a minimização da função  $\mathcal{O}_i$  equivale a maximização de  $-\mathcal{O}_i$ .

Definições adicionais são úteis para discutir a Eq. (4.2). Uma solução  $\mathbf{S}$  é possível se as restrições da Eq. (4.2) são satisfeitas. O conjunto de todas as soluções possíveis é a *região factível* do *espaço de busca*  $S$ . O vetor  $\mathbf{O} = (\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_L)$  de dimensão  $L$  com os valores de todos as funções objetivo é localizado em um espaço multi-dimensional chamado *espaço objetivo*. Esta é a fundamental distinção com relação ao problema de otimização mono-objetivo, o qual tem o espaço objetivo unidimensional. Então, em vez de buscar uma solução única para o problema, como ocorre nos problemas de otimização mono-objetivo de SB, a solução para o problema de SB multiobjetivo é um conjunto ( $\bar{\mathbf{S}}_\infty^\dagger$ ) dos *Pontos de Pareto*.

As soluções da Eq. (4.2) são chamadas *Ótimas de Pareto* porque para melhorar um objetivo de um conjunto de várias soluções objetivo, pelo menos um outro objetivo torna-se pior [15], essas soluções ótimas formam o que é conhecido como *fronteira de Pareto*. Formalmente, o conjunto  $\mathbf{S}^*$  é uma solução ótima de Pareto se não existe outro conjunto  $\mathbf{S}$  que domine  $\mathbf{S}^*$ , isto é,  $\mathcal{O}_i(\mathbf{S}) \leq \mathcal{O}_i(\mathbf{S}^*) \forall i = 1, \dots, L$ , e que a solução  $\mathbf{S}^*$  é estritamente melhor que  $\mathbf{S}$  em pelo menos um objetivo, isto é,  $\mathcal{O}_j(\mathbf{S}) < \mathcal{O}_j(\mathbf{S}^*)$ , para pelo menos um  $j = 1, \dots, L$ .

Daqui em diante e sem perda de generalidade, o foco será a versão do problema de SB relacionado com taxa adaptativa (RA - *rate-adaptive*), onde o objetivo é maximizar a taxa de transmissão de bits de dados [6]. E, a solução proposta a seguir não fica restrita ao problema de taxa adaptativa mas, é válida também, para o problema de minimização de potência (margem fixa) [29].

Um ponto de operação ótimo de Pareto na versão de taxa adaptativa da Eq. (4.2) precisa estar na fronteira de Pareto correspondente a  $L = N$  e  $\mathcal{O}_n = R_n$ . Além disso, existem  $Q = N$  desigualdades  $q_n = p_n - P_n^{\max} \leq 0, \forall n$ , as quais, limitam a potência por usuário  $p_n$  a um valor máximo  $P_n^{\max}$ . Não existem nenhuma restrição de igualdade ( $T = 0$ ) e nem de limitação da máscara da PSD, isto é,  $M_k = \infty$ . De maneira geral, a solução para SB multiobjetivo para a versão de taxa adaptativa é:

$$\begin{aligned} \bar{\mathbf{S}}_\infty^\dagger = \arg \max_{\mathbf{S}} \quad & R_n \quad , n = 1, \dots, N \\ \text{tal que} \quad & p_n \leq P_n^{\max} \quad , n = 1, \dots, N. \end{aligned} \quad (4.3)$$

A frente de Pareto resultante da otimização da Eq. (4.3) é chamada de *região de taxa*, uma vez que caracteriza todas as combinações de taxas de dados ótimas entre os usuários.

### 4.2.2 Análise do Problema de Balanceamento de Espectro Multiobjetivo

Assuma que  $\mathcal{D}$  seja a notação para diversidade e o interesse está em obter os  $Z$  pontos de operação, e o problema de DSSB corresponde em encontrar o melhor  $\bar{\mathbf{S}}_Z^\dagger$  de acordo com:

$$\bar{\mathbf{S}}_Z^\dagger = \arg \max_{\bar{\mathbf{S}}_Z} \mathcal{D}(\bar{\mathbf{S}}_Z), \quad (4.4)$$

onde  $\bar{\mathbf{S}}_Z$  contém somente soluções ótimas Pareto de acordo com a Eq. (4.2) ou, de maneira alternativa, de acordo com o caso especial da Eq. (4.3).

Existem várias maneiras para avaliar a diversidade de uma solução ótima de Pareto. Uma técnica relevante é o *spacing* [15]. Para simplificar sua descrição, considere que  $\mathcal{O}_i^z = \mathcal{O}_i(\mathbf{S}_z, \mathbf{N}, \bar{\mathbf{H}})$  representa o valor da  $i$ -ésima função objetivo para a matriz  $\mathbf{S}_z$  de PSDs. A métrica de diversidade *spacing* é dada por

$$\mathcal{D}(\bar{\mathbf{S}}_Z) = \left( \frac{1}{Z} \sum_{z=1}^Z (d_z - \bar{d})^2 \right)^{-1/2}, \quad (4.5)$$

onde

$$d_z = \min_{\substack{m=1, \dots, Z \\ m \neq z}} \sum_{i=1}^L |\mathcal{O}_i^z - \mathcal{O}_i^m|, \quad (4.6)$$

e

$$\bar{d} = \left( \sum_{z=1}^Z d_z \right) / Z. \quad (4.7)$$

Essa métrica corresponde ao inverso do desvio padrão de  $d_z$ , e  $d_z$  representa o menor valor da soma das diferenças absolutas entre dois pontos da solução no espaço objetivo. Em outras palavras,  $d_z$  retorna um grande valor quando as soluções não estão próximas. Intuitivamente pode-se concluir que um conjunto  $\bar{\mathbf{S}}_Z$  é diverso quando a solução tem valores similares de  $d_z$ . Em um caso extremo, se todas as soluções  $Z$  tem os mesmos valores  $d_z$ , o desvio padrão correspondente a essa solução seria zero resultando em diversidade infinita ( $\mathcal{D}(\bar{\mathbf{S}}_Z) = \infty$ ). A seguir é apresentado um exemplo numérico para analisar a métrica *spacing*.

A Tabela 4.1 lista dois diferentes conjuntos de soluções  $\bar{\mathbf{S}}_Z^a$  e  $\bar{\mathbf{S}}_Z^b$  os quais serão comparados. Ambos assumem um espaço objetivo de duas dimensões ( $L = 2$ ) e tem  $Z = 4$  soluções. Essas soluções são representadas graficamente na Fig. 4.2. Assumindo o primeiro conjunto de soluções  $\bar{\mathbf{S}}_Z^a$  na Tabela 4.1 e o objetivo é obter  $d_1, d_2, d_3$  e  $d_4$ . Para determinar o valor de  $d_1$  a partir da Eq. (4.6) tem que minimizar a diferença acumulada, assim o primeiro candidato é  $m = 2$  o qual resulta em  $|0 - 0,51| + |1 - 0,85| = 0,66$ . Para  $m = 3$  o valor é  $0,88 + 0,53 = 1,41$  e para  $m = 4$  é 2. Assim,  $d_1 = 0,66$  devido ser o valor mínimo comparado com os outros valores (1,41 e 2). De maneira similar determina-se  $d_2 = 0,66$ ,  $d_3 = 0,59$  e  $d_4 = 0,59$ . O desvio padrão dos valores de  $d_z$  vale  $\sigma = 0,035$ , e que resulta em  $\mathcal{D}(\bar{\mathbf{S}}_Z^a) = 1/\sigma \approx 28,6$ . Para o segundo

conjunto de soluções tem-se os valores  $d_z = 0, 17, 0, 17, 0, 1, 0, 1$ , o quais resultam em  $\mathcal{D}(\bar{\mathbf{S}}_Z^b) = 1/0,035 \approx 28,6$ . Como as duas soluções resultam nos mesmos valores de  $\mathcal{D}(\bar{\mathbf{S}}_Z)$ , e pela Fig. 4.2 é possível perceber que a segunda solução (Fig. 4.2(b)) não tem o mesmo espalhamento do que a primeira solução ((Fig. 4.2(a)), então pode-se concluir que a métrica *spacing* não considera o espalhamento das soluções, ou seja, a diversidade. Assim, para avaliar os resultados do método proposto não é adequado, dada a necessidade de avaliar a diversidade pois, para o planejamento de uma rede DSL, a operadora desse sistema poderia precisar conhecer os pontos de operação possíveis para determinado conjunto de assinantes e para isso o desejável seria ter pontos de operação com diversidade, caso contrário, o planejamento ficará muito restrito.

Apesar de existirem outras métricas que tentam medir diversidade [15], cada uma das métricas estudadas precisam de informações para realizarem seus cálculos, tais com, deve-se ter uma região ótima para verificar se os dados obtidos estão próximos dela ou identificar quem são os pontos extremos das curvas, esses requisitos parecem simples para muitos problemas, porém para o problema proposto foi extremamente difícil obtê-las, principalmente quando o problema trata de casos com  $N > 2$ , por isso realizou-se a implementação de uma nova métrica [33, 34] que atendesse a necessidade de avaliação dos resultados obtidos, a qual será discutida em seguida.

Tabela 4.1: Dois conjuntos diferentes de soluções  $\bar{\mathbf{S}}_Z^a$  e  $\bar{\mathbf{S}}_Z^b$  para  $L = 2$  e  $Z = 4$ . As soluções são mostradas nos gráficos da Fig. 4.2.

	$\bar{\mathbf{S}}_Z^a$		$\bar{\mathbf{S}}_Z^b$	
$z$	$O_1$	$O_2$	$O_1$	$O_2$
1	0	1	0,57	0,82
2	0,51	0,85	0,67	0,75
3	0,88	0,47	0,78	0,63
4	1	0	0,82	0,57

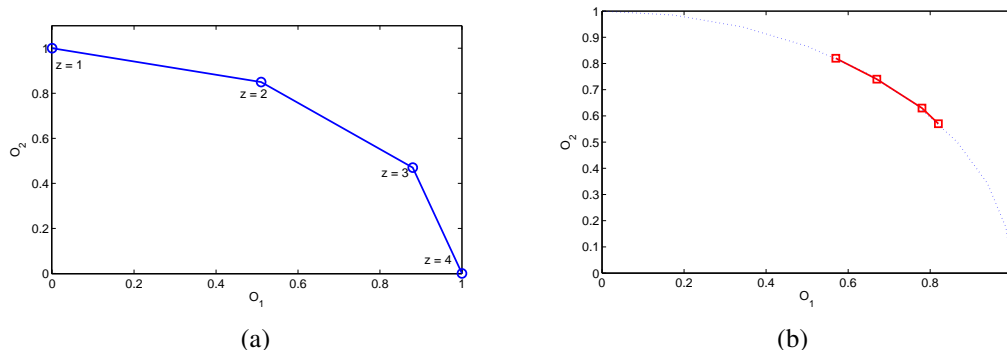


Fig. 4.2: Exemplo de dois conjuntos de soluções  $\bar{\mathbf{S}}_Z^a$  e  $\bar{\mathbf{S}}_Z^b$  (veja valores na Tabela 4.1) que tem o mesmo valor (28,6) de diversidade de acordo com a métrica *spacing* da Eq. (4.5).

A definição de diversidade utilizada neste trabalho é baseada na geometria computacional [35]

e que pode ser explicada com os seguinte elementos. Assume-se que  $Z$  pontos são os vértices de um polítopo convexo  $L$ -dimensional, o qual tem *facetas* de dimensão  $L-1$ . Considerando que as facetas, por si só, são polítopos, cujas facetas são arestas de dimensão  $L-2$  (também chamadas de *subfacetas*) do polítopo original. As arestas são novamente ditas polítopos cujas facetas são de dimensão  $L-3$  do polítopo original, e assim por diante. Por exemplo, quando  $L = 2$ , como na Fig. 4.2, uma faceta é um segmento de linha (será apresentado, mas adiante no Capítulo 5, para o caso de  $L = 3$  na Fig. 5.6, onde uma faceta é um polígono).

O método proposto calcula as áreas das facetas  $A_i, i = 1, \dots, U$  dos polítopos usando o algoritmo Quickhull [36] e, em seguida, o desvio padrão  $\sigma_A = \sqrt{(1/U) \sum_{i=1}^U (A_i - \mu_A)^2}$  dessas áreas, onde  $\mu_A = (1/U) \sum_{i=1}^U A_i$  é a média. A *diversidade do polítopo* é então calculada como

$$\mathcal{D}(\bar{\mathbf{S}}_Z) = 1/\sigma_A. \quad (4.8)$$

Outras heurísticas foram avaliadas para expressar a diversidade das áreas, porém a Eq. (4.8) é mais simples e conduz a bons resultados. O método proposto é similar ao da Eq. (4.5) que usa o inverso do desvio padrão, porém em vez de utilizar  $d_z$  usa a área das facetas  $A_i$ . A Eq. (4.5) é mais simples de calcular que a Eq. (4.8) porém, considerando o papel de  $d_z$  e  $A_i$  no problema de geometria, observa-se que as áreas das facetas ( $A_i$ ) conduzem a uma definição mais adequada de diversidade para o problema de DSSB.

Existem ainda dois importantes detalhes a fazer sobre Eq. (4.8) que a fazem trabalhar eficientemente para o problema de DSSB. O primeiro é a inclusão de *pontos de formação* de facetas. Por exemplo, no caso de  $\bar{\mathbf{S}}_Z^b$  na Fig. 4.2(b), a faceta poderia conectar os pontos  $z = 1$  ao ponto  $z = 4$  o que resultaria no cálculo da área de  $A = \sqrt{(0,57 - 0,82)^2 + (0,82 - 0,57)^2} = 0,3536$  (no caso bidimensional, a área da faceta corresponde ao comprimento do segmento de reta que liga os dois pontos), esse valor de  $A = 0,3536$  seria o maior segmento de reta da distribuição, sendo desproporcional aos demais, lembrando que ele corresponde ao primeiro ponto ligado direto ao último ponto, esse cálculo seria necessário para fechar a região da envoltória complexa. Para evitar este tipo de faceta e para promover um prolongamento dos pontos ao longo do espaço objetivo,  $L + 1$  pontos de formação são sempre adicionados aos  $Z$  pontos em  $\bar{\mathbf{S}}_Z$  antes de obter as envoltórias convexas (*convex hull*). O primeiro ponto a ser localizado está na origem  $(0, 0, \dots, 0)$  e os outros tem somente o  $i$ -ésimo diferente de zero, onde  $i = 1, \dots, L$ . Por exemplo, quando resolve-se a Eq. (4.3) o ponto de formação  $(r_1^{\max}, 0, 0, \dots, 0)$  é incluído, onde  $r_1^{\max}$  é a taxa de dados obtida pelo usuário 1 quando todos os outros usuários estão inativos. A taxa de dados do segundo usuário ( $r_2^{\max}$ ) determina outro ponto de formação  $(0, r_2^{\max}, 0, \dots, 0)$  e assim por diante. Depois de calcular a região adequada de convexidade com os pontos de formação, todos as áreas das facetas, as quais têm seus vértices como sendo os pontos de formação são excluídos do cálculo do  $\sigma'_A$ .

Um exemplo usando  $\bar{\mathbf{S}}_Z^b$  na Fig. 4.2(b) é útil para o entendimento do cálculo da *diversidade* usando a Eq. (4.8) e dos pontos de formação. Considera-se que  $r_1^{\max} = r_2^{\max} = 1$ , tal que,

os três pontos de formação sejam  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$ . Assim, todos os sete pontos (os quatro originais de  $\bar{\mathbf{S}}_Z^b$  e os três pontos de formação) estão na envoltória convexa do novo politopo. A inclusão dos pontos de formação aumenta o número de facetas de quatro para sete e suas áreas são: 1, 1, 0, 5977, 0, 5977, 0, 1221, 0, 1628 e 0, 0721. Todos os vértices das primeiras duas facetas (os quais são iguais a 1) são pontos de formação e essas facetas devem ser excluídas. Por fim, o desvio padrão das cinco<sup>2</sup> facetas restantes vale  $\sigma_A = 0,2363$ , o qual conduz a  $\mathcal{D}(\bar{\mathbf{S}}_Z^b) = 4,23$ , usando a Eq. (4.8).

O segundo detalhe serve para fazer a *medida de diversidade* adequada para fins práticos de acordo com o problema de DSSB, ele estabelece que é necessário excluir da solução  $\bar{\mathbf{S}}_Z$  os pontos que não estão na fronteira de Pareto, ou seja, aqueles pontos que não estão na envoltória convexa dos polítopos. Isto é importante devido existir a necessidade de se usar algoritmos de balanceamento de espectro sub-ótimos, como será discutido na seção 4.3, e esses algoritmos podem retornar um conjunto  $\bar{\mathbf{S}}_Z$  que tem somente  $C < Z$  soluções na frente de Pareto.

De acordo com a Eq. (4.2), se uma solução não faz parte de fronteira de Pareto, ela não deverá fazer parte do conjunto  $\bar{\mathbf{S}}_Z$ . Contudo, isto é uma consequência em se adotar algoritmos SB sub-ótimos para a redução do custo computacional. Por causa disso, quando usado em problema de DSSB, o método da Eq. (4.8) deverá ser modificado para penalizar a existência de  $Z - C$  soluções que não estão na fronteira de Pareto. Algumas heurísticas foram testadas, porém uma simples obtém um bom resultado: primeiro obtém-se  $\sigma'_A$ , o desvio padrão das áreas correspondentes as facetas dos vértices  $C$  que estão na região de convexidade (isto é, são descartadas as solução em  $\bar{\mathbf{S}}_Z$  que não são parte da região convexa), e divide-se  $\sigma'_A$  pela fração  $C/Z$  da solução de Pareto. De acordo com esse procedimento, o valor de  $\sigma_A$  para ser usado na Eq. (4.8) é obtido por

$$\sigma_A = \left( \frac{Z}{C} \right) \sigma'_A. \quad (4.9)$$

Quanto menor a fração  $C/Z$ , a diversidade segundo a Eq. (4.8) será penalizada. Por exemplo, assuma que os pontos  $(0, 3, 0, 3)$  não pertençam à região de convexidade mas são incluídos em  $\bar{\mathbf{S}}_Z^b$  na Fig. 4.2 para criar um novo conjunto, que usando resultado anteriormente calculado, terá diversidade  $\mathcal{D}(\bar{\mathbf{S}}_Z^c) = (4/5)4,23 = 3,386$ . No caso de  $\bar{\mathbf{S}}_Z^a$ ,  $C = Z = 4$ , as áreas são 0,5316, 0,4851, 0,5304 e  $\sigma_A = \sigma'_A = 0,0216$ , por isso  $\mathcal{D}(\bar{\mathbf{S}}_Z^a) = 46,2$ , o qual é maior que  $\mathcal{D}(\bar{\mathbf{S}}_Z^b)$ , como desejado.

A próxima seção foca na apresentação do método proposto e a Eq. (4.8) será usada no Capítulo 5 para avaliá-lo.

<sup>2</sup>Originalmente, eram quatro pontos e três facetas (ou, nesse caso, segmentos) e, os pontos  $(1, 0)$  e  $(0, 1)$  não foram excluídos, isso para poder promover um prolongamento dos pontos ao longo do espaço objetivo, como citado anteriormente.



## 4.3 O Método Proposto para Solucionar o Problema de DSSB

O problema de DSSB apresentado na equação Eq. (4.4) é um de solução complicada e não é viável a utilização de um método de tentativa e erro (força bruta), pois seria necessário um alto custo computacional, como será detalhado na sequência. Neste trabalho, um algoritmo evolucionário multiobjetivo (MOEA - *multi-objective evolutionary algorithm*) será usado para encontrar uma solução alternativa para o problema [33, 34].

Uma das grandes vantagens em se usar algoritmos evolucionários está no ganho em flexibilidade e adaptabilidade para resolver uma tarefa, combinado com um desempenho robusto (embora isso dependa da classe de problema e das características de procura globais) [37]. As técnicas evolucionárias são úteis para resolver difíceis problemas de otimização, porém elas tem baixa convergência quando comparado com algoritmos que se beneficiam do conhecimento sobre um problema específico.

Para resolver o problema de DSSB dado pela Eq. (4.4), uma busca baseada em computação evolucionária parece muito difícil porém, este trabalho defende a adoção de uma *aproximação híbrida*, onde o algoritmo evolucionário direciona a busca para um solução ótima de Pareto e para um conjunto diverso de pontos, enquanto uma busca local é baseada em um Algoritmo de SB que fornece as PSDs em resposta à solicitação do MOEA.

As características dos procedimentos globais, tais como otimização e custo computacional, dependem das combinações dos algoritmos. Neste trabalho são explorados as seguintes combinações, a qual é chamada de algoritmo *diverseSB*: O MOEA é o *non-dominated sorting genetic algorithm II* (NSGA-II) [27] e para o algoritmo de SB assume-se a formulação de soma ponderada (*weighed-sum*), tais com *iterative SB* (ISB) [8] e *Successive Convex Approximation for Low Complexity* (SCALE) [9].

A próxima seção descreve a motivação para usar algoritmos de SB a partir de soma ponderada no contexto dos algoritmos de taxa adaptativas e então, o algoritmo proposto é discutido.

### 4.3.1 Algoritmo de Balanceamento de Espectro para Soluções Multiobjetivo

Em vez de solucionar o problema multiobjetivo Eq. (4.2), os algoritmos de balanceamento de espectro solucionam uma versão mono-objetivo (ou de objetivo único) do problema. Porém, uma aproximação baseada no fato de que uma otimização multiobjetivo pode ser convertida em problemas mono-objetivos usando uma *função de objetivo agregado* - FOA [15] será considerada com solução, pois a FOA combina todas as funções objetivos em uma simples forma funcional. Uma alternativa possível é a “soma linear de pesos” dos objetivos ou, simplesmente, *soma ponderada*, a qual é largamente adotada em SB (veja exemplos em [17]).

A soma ponderada estabelece que um peso escalar  $\omega_i$  pode ser especificado para otimizar

cada objetivo  $\mathcal{O}_i$ , e então a soma ponderada dos objetivos é combinada em uma função que pode ser otimizada por alguma otimizador mono-objetivo. A solução obtida dependerá dos valores relativos dos pesos especificados, os quais priorizam os objetivos.

Os algoritmos convencionais de SB de taxa adaptativa usando soma ponderada [17, 2, 38, 28, 9] adotam  $\mathbf{O} = \omega_1 r_1 + \dots + \omega_N r_N$ , como a FOA, e solucionam o problema equacionado como:

$$\begin{aligned} \mathbf{S}^* &= \arg \max_{\mathbf{S}} \sum_{n=1}^N \omega_n r_n \\ \text{tal que } p_n &\leq P_n^{\max}, n = 1, \dots, N, \end{aligned} \quad (4.10)$$

onde  $\mathbf{S}^*$  é uma simples matriz de  $N \times K$  com PSDs (correspondendo a um ponto de operação). É possível que  $\mathbf{S}^*$  não esteja na fronteira de Pareto de acordo com a Eq. (4.3), isto é, pode ser que  $\mathbf{S}^* \notin \bar{\mathbf{S}}_{\infty}^{\dagger}$ . A razão para isso é que os algoritmos SB não podem garantir a obtenção de soluções ótimas, nem a convergência dos resultados, apesar de alcançarem bons resultados na prática. O algoritmo ótimo de SB (OSB - *optimal spectrum balancing*) [19] resulta em ótimos valores porém, não é uma solução prática dada o seu elevado custo computacional.

O proposto *diverseSB* assume que se tem a disponibilidade de um algoritmo de SB de taxa adaptativa com soma ponderada que tente resolver Eq. (4.10), fornecendo as PSDs (eventualmente sub-ótimo) para um ponto de operação, dada a informação do canal e  $N$  pesos. Na prática, uma solução pode ser alcançada com  $N - 1$  pesos, pois a soma de todos os pesos pode ser igual a um ( $\sum_{n=1}^N \omega_n = 1$ ) ou um outro valor.

Uma solução direta para encontrar os pontos de operação, de acordo com Eq. (4.10), seria variar os valores dos  $N$  pesos ( $\omega$ ), e a cada combinação dos pesos gerados, passá-los para um algoritmo de balanceamento de espectro de taxa adaptativa que seja formulado com soma ponderada para encontrar pontos de operações distintos, formando assim, uma região de taxa.

Ocorre que, o mapeamento entre os pesos e funções objetivos é tipicamente não linear (como poderá ser comprovado com os resultados das simulações realizadas) e não é trivial encontrar um conjunto diverso constituído de soluções ótimas pela discretização do espaço  $\mathbb{R}^N$  (cada um dos  $N$  pesos pode assumir valores dentro de um intervalo, o qual é discretizado e o intervalo pode estar entre  $[0, 1]$ ). Os eixos<sup>3</sup> podem ser discretizados de maneira uniforme ou não uniforme.

Uma discretização linear ou uniforme consiste dos produtos Cartesianos dos  $N$  eixos, discretizados de acordo com  $\{0, \Delta, 2\Delta, \dots, (G - 1)\Delta\}$ , onde  $G$  é o número de pontos em cada eixo,  $\Delta$  é o tamanho do passo. Considerando que  $\omega$  assume valores no intervalo  $[0, 1]$ , então  $\Delta = 1/G$ .

Uma discretização não uniforme também pode ser usada através da discretização dos eixos de acordo com uma escala logarítmica.

Contudo, o custo computacional para obter um conjunto diverso através dessas discretizações é muito alto, principalmente para casos com 3 ou mais dimensões. Por exemplo, como será detalhado

<sup>3</sup>É chamado de eixo a faixa de valores que cada peso ( $\omega$ ) pode assumir.

depois, uma implementação com discretização uniforme para  $N = 25$  usuários poderia necessitar de 27 dias usando um computador pessoal atual.<sup>4</sup> Um método alternativo é descrito na sequência.

### 4.3.2 Algoritmo Proposto: *diverseSB*

Esta subseção descreve uma solução para o problema de DSSB. O método proposto ameniza o custo computacional exponencial para discretização dos eixos, conforme apresentado na subseção anterior, usando uma aproximação híbrida, na qual o algoritmo NSGA-II conduz à busca para encontrar uma solução ótima de Pareto e um conjunto de pontos diversos. Enquanto, uma busca local é baseada no algoritmo de SB de taxa adaptativa e com soma ponderada, tais como SCALE e ISB, para calcular as PSDs para um conjunto de pesos.

O algoritmo NSGA-II mantém uma população de indivíduos  $e$ , seguindo os princípios da evolução natural, o otimizador aperfeiçoa iterativamente a população, gerando novos indivíduos de acordo os mecanismos de *mutation*, *crossover* e *elitism* [15].

No algoritmo evolucionário, cada indivíduo (também chamado *cromossomo*)  $\mathbf{x}$  representa um conjunto de  $N$  pesos  $\omega_n$  na Eq. (4.10), com  $\omega_n \in [0, 1]$ . O tamanho da população é  $Z$ , o qual representa o número de pontos de operações definidos pelo usuário. No processo de otimização é realizada uma busca para encontrar as soluções que estejam na frente de Pareto. Para tal, é necessário calcular o vetor  $\mathbf{O} = (\omega_1 r_1, \omega_2 r_2, \dots, \omega_N r_N)$  com todas as funções objetivos em Eq. (4.10). Esse cálculo requer a chamada ao algoritmo SB, o qual recebe os pesos representados por  $\mathbf{x}$  e as informações do canal  $(\mathbf{N}, \bar{\mathbf{H}})$  como entrada e calcula as PSDs de saída  $(\mathbf{S})$ . Este procedimento corresponde em obter-se um ponto de operação, ou seja é uma solução mono-objetivo. Para a solução multi-objetivo, com vários pontos de operação, após os cálculos de cada pontos de operação pelo algoritmo de SB, o algoritmo evolucionário fica encarregado de receber os pontos de operação, formando a população de indivíduos, avaliá-los e melhorá-los a cada geração, através do uso dos operadores genéticos, até chegar a convergência ou até alcançar um número máximo de iterações. O resultado final é um conjunto  $\bar{\mathbf{S}}_Z$  de  $Z$  pontos de operação. A Fig. 4.3 apresenta um fluxograma do algoritmo *diverseSB*.

Como discutido previamente, a Eq. (4.8) é usada para avaliar a solução final, e que essa métrica não consta na iteração do algoritmo *diverseSB*. Isso ocorre para que seja evitado o custo computacional do cálculo do *convex hull*. Outra razão é que um MOEA convencional somente requer uma métrica para diversidade individual (a distância de multidão é usada neste trabalho) e poderia ser necessário realizar modificações para levar em conta a diversidade global do conjunto de soluções.

O algoritmo inicia com a criação de uma população aleatória  $\mathbf{P}_1$  (onde o subscrito denota a contagem da geração) com  $Z$  indivíduos, cada um codificado com um conjunto de  $N$  pesos  $\omega_n \in$

<sup>4</sup>Essa estimativa assume o algoritmo SCALE e uma discretização com  $G = 20$  pontos por usuário. Reduzindo a resolução para  $G = 10$  ainda corresponde a aproximadamente 40 horas de processamento. As simulações foram realizadas usando um computador com processador Core 2 Duo de 2 GHz e com 2 GB de RAM.

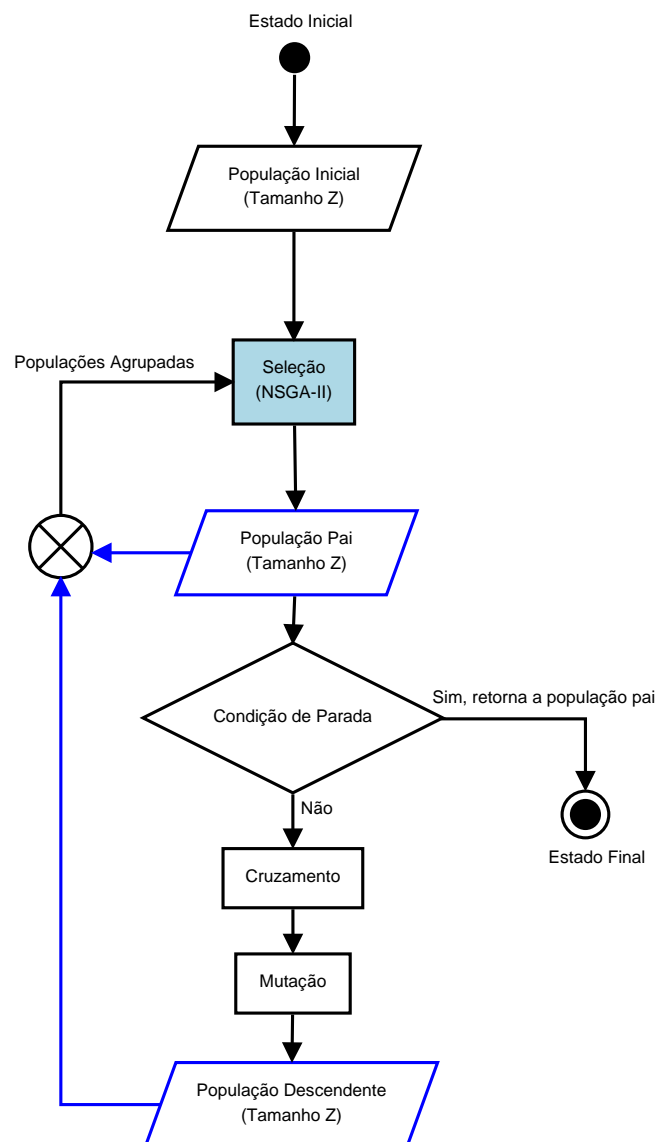


Fig. 4.3: Fluxograma do método proposto para solucionar o problema de DSSB.

$[0, 1]$ . Esses pesos podem ser obtidos, por exemplo, por uma distribuição uniforme. Então, a primeira população  $\mathbf{P}_1$  é submetida aos operadores de *seleção*, *cruzamento* e *mutação* para gerar os primeiros descendentes  $\mathbf{P}_2$  (a segunda geração). Depois disso, realiza-se o elitismo, onde  $Z$  indivíduos são selecionados, a partir da união das gerações anteriores ( $\mathbf{P}_1 \cup \mathbf{P}_2$ ), para formar a próxima geração  $\mathbf{P}_3$ , para isso, é considerada como métrica a dominância e a diversidade para a seleção dos indivíduos da nova geração (mais detalhes sobre o conceito de elitismo, dominância e diversidade em *algoritmos genéticos* podem ser encontrados, por exemplo, em [15, 27]). Esse processo é repetido até que um critério de parada seja encontrado, como exemplo, um número máximo de gerações  $I_{\max}$ . Uma descrição desses procedimentos é apresentada no algoritmo da Fig. 4.4.

Deve ser observado que a medida de diversidade discutida no parágrafo anterior objetiva ajudar a selecionar os melhores indivíduos que formarão a população da próxima geração, e assim obter

**Entrada:**

- 1 a)  $Z$  - número desejado de pontos de operações;
- 2 b)  $sba$  - um algoritmo de balanceamento de espectro;
- 3 c)  $moea$  - um algoritmo evolucionário multiobjetivo;
- 4 d) Conjunto de funções objetivo e restrições;
- 5 e)  $(\mathbf{N}, \bar{\mathbf{H}})$  - informação do canal.

**Saída:**  $\bar{\mathbf{S}}_Z^\dagger$  - PSDs

- 6 Cria uma população inicial com  $\mathbf{P}_1$  com  $Z$  indivíduos.
- 7  $t \leftarrow 1$
- 8 **repita**
  - 9 Execute  $sba$  para cada indivíduo  $\mathbf{x}_i$  para obter a respectiva respectiva matriz  $N \times K$  com PSDs  $\mathbf{S}_i$ ;
  - 10 Calcule a função objetivo para todos os indivíduos;
  - 11 Execute  $moea$  para aperfeiçoar a atual população  $\mathbf{P}_t$ , criando uma nova população  $\mathbf{P}_{t+1}$  de  $Z$  indivíduos, com função objetivo e diversidade melhorados.
  - 12  $t \leftarrow t + 1$
- 13 **até** *convergência*;
- 14 Converter a população final  $\mathbf{P}^\dagger$  dos pesos em um conjunto de matrizes PSDs  $\bar{\mathbf{S}}_Z^\dagger$ .
- 15 **retorna**  $\bar{\mathbf{S}}_Z^\dagger$ ;

Fig. 4.4: Algoritmo *diverseSB* Proposto.

$\mathbf{S}$  (a matriz de PSD), enquanto que as medidas diversidade do politopo discutida na Seção 4.2.2 são usadas para avaliar o conjunto inteiro de pontos e não um indivíduo da população, como acontece com o conceito de diversidade discutida no algoritmo genético. A medida de diversidade individual usada no algoritmo da Fig. 4.4 foi a *distância de multidão* [39, 27]. A distância de multidão é calculada pelo perímetro do maior cubóide que inclui cada ponto do espaço objetivo sem incluir qualquer outro ponto [15]. Depois de normalizada a distância de multidão para cada solução não-dominada, elas são classificadas em ordem ascendente e usadas no passo 11 do algoritmo apresentado na Fig. 4.4. O Capítulo 5 são apresentados os resultados obtidos com método proposto e uma comparação com as técnicas de obtenção de pontos de operação usando discretização uniforme e não-uniforme.

# Capítulo 5

## Resultados Obtidos

Este Capítulo apresenta os resultados obtidos pelas simulações realizadas, através dos quais é possível avaliar o desempenho do método proposto *diverseSB* e, compará-los com dois métodos básicos, os quais correspondem em adotar uma discretização uniforme, não-uniforme (logarítmica e aleatório) para a variação da faixa valores de pesos ( $\omega_n$ ) dispostas entre  $[0, 1]$ .

As simulações usam cenário DSL, tomando como base um cenário geral conforme a Fig. 5.1. Esse cenário consiste de um escritório central (EC) e três terminais remotos (TR), e para as simulações será considerada a transmissão na direção *downstream*. As simulações são organizadas em: simulações com 2 usuários (envolvendo o EC e TR3), simulações com 3 usuários (envolvendo EC, TR1 e TR3), simulações com 4 usuários (EC, TR1, TR2 e TR3), simulações com 5 e 6 usuários pelo acréscimo de mais linhas DSL ao EC.

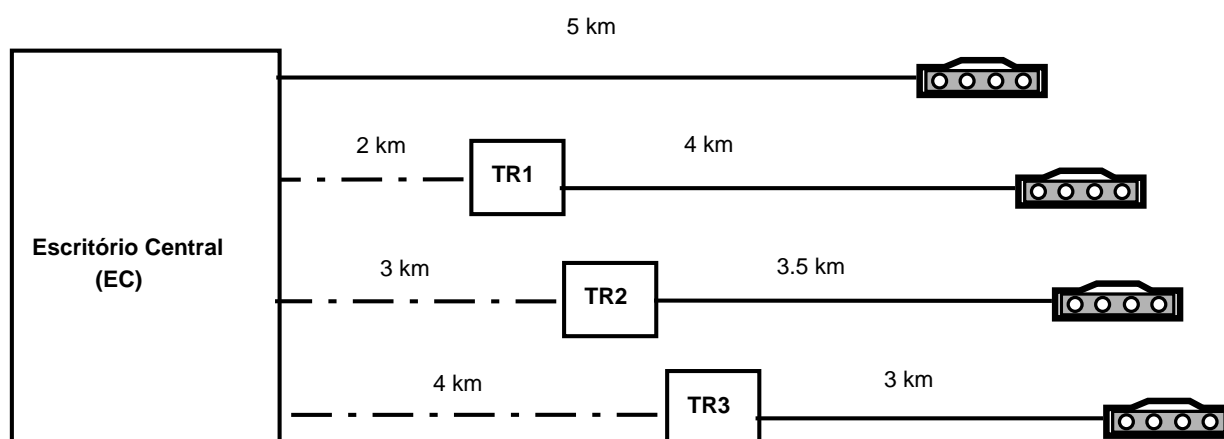


Fig. 5.1: Cenário DSL geral com 4-usuários.

Considera-se que todos os modems DSL adotam a tecnologia ADSL. O modelo de cabo adotado para todas as linhas durante as simulações foi ANSI TP2 24 AWG. No algoritmo *diverseSB* foram usados como parâmetros do NSGA-II: probabilidade de cruzamento de 0,7; probabilidade de

mutação de 0,3; os tamanhos da população e da geração foram atribuídos de acordo com as simulações realizadas, pois quanto maior o número de linhas DSL, mais pontos são necessário para compor a região de taxa, não tendo uma forma padrão para especificar a quantidade de pontos que devem ser escolhidos para uma dada quantidade de linhas. O algoritmo de SB adotado nas simulações foi SCALE [9]. O software Qhull [40] foi utilizado para o cálculo das áreas das facetas na aplicação da métrica apresenta na Eq. (4.8).

Para a comparação do custo computacional entre o método proposto e os métodos de discretização uniforme, logarítmica e aleatório será considerado o número de vezes que o algoritmo de SB será executado ( $num\_SB$ ). Esta métrica de avaliação foi proposta dado que o custo computacional de outras rotinas do *diverseSB* são bem menores quando comparados com a rotina do algoritmo de SB, por exemplo, a escolha dos pesos demanda muito menos tempo (um custo computacional menor) do que o SB algoritmo. Excluindo o tempo investido em  $num\_SB$ , o custo computacional adicional proveniente do MOEA é maior que o uso dos métodos de discretização dos pesos, porém esse custo adicional é muito menor que o custo necessário para a execução total do algoritmo proposto, o qual depende do número de execuções do algoritmo SB. Então, a complexidade do algoritmo *diverseSB* proposto pode ser aproximada por  $\mathcal{O}(num\_SB \times SB_{complexity})$ , onde  $SB_{complexity}$  representa a complexidade do algoritmo de SB adotado. Na Tab. 2.1 são apresentas as complexidades de alguns algoritmos de SB.

Apesar de ambos, os algoritmos de MOEA e os de discretização, terem custo computacional que dependem linearmente do número  $num\_SB$  de execuções dos algoritmos SB, eles diferem nos parâmetros dos quais  $num\_SB$  depende. Quando usam uma discretização  $N$ -dimensional e assumindo o mesmo número de pontos  $G$  em todos os eixos Cartesianos, o número de  $num\_SB = G^N$ . Para o MOEA,  $num\_SB$  dependendo do tamanho da população  $Z$  e do número máximo  $I_{max}$  de iterações e é dado por  $num\_SB = I_{max}Z$ . Para proporcionar uma comparação justa, as simulações comparam algoritmos com o mesmo número  $num\_SB$ , tal que o tempo de execuções sejam equivalentes. A próxima subseção apresenta os resultados para  $N = 2$  usuários.

## 5.1 Simulações com 2-usuários

O cenário usado para a simulação com 2-usuários, obtém-se da Fig. 5.1 e, é apresentado na Fig. 5.2 por conveniência.

As simulações realizadas para o cenário da Fig. 5.2 usam o mesmo número de  $num\_SB$  ( $S = 144$ ) e o mesmo número de soluções finais ( $Z = 12$ ), ou seja, cada método realiza 144 execuções do algoritmo SB e no final são obtidos 12 pontos de operação.

Para as discretizações uniforme, logarítmica e aleatória foi usado  $G = 12$  que, como são dois usuários, tem-se  $12 \times 12 = 144$  combinações para os pesos ( $\omega_n$ ). Como são 144 valores de  $\omega_n$ , são gerados 144 pontos de operação pelos métodos de discretização uniforme, logarítmica e aleatória.

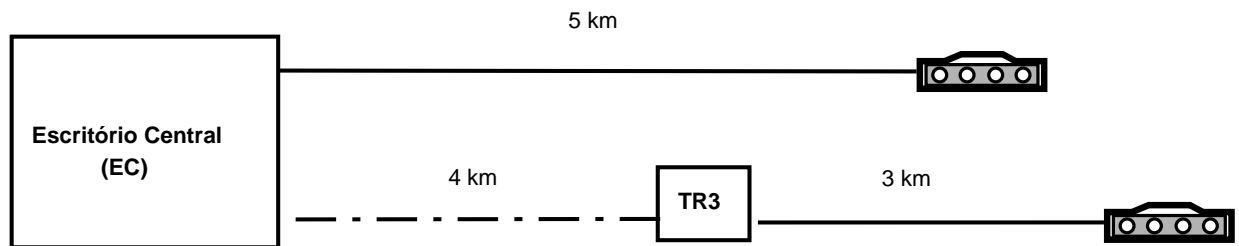


Fig. 5.2: Cenário DSL com 2-usuários ( $N = L = 2$ ).

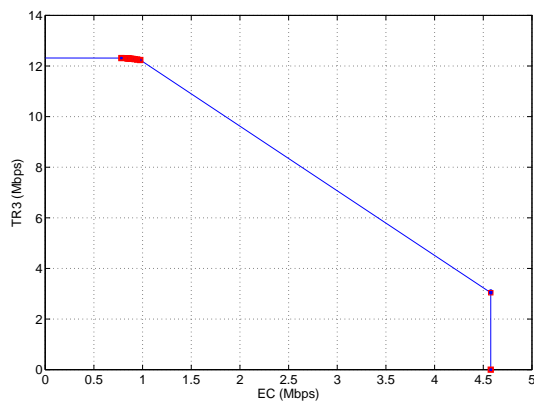
Então, para se ter uma comparação justa dos resultados obtidos por esses métodos com o método proposto, do total de 144 pontos de operação obtidos, um conjunto diverso de 12 pontos de operação foi selecionado usando distância de multidão, a mesma medida usada para determinar a diversidade de um indivíduo no *diverseSB*. Para o método proposto *diverseSB*, as simulações foram realizadas com populações de tamanho  $Z = 12$  indivíduos (ou pontos de operação) e critério para  $I_{\max} = 12$  gerações, o que também resulta em 144 *num\_SB*.

Os resultados são apresentados na Fig. 5.3. Através de uma inspeção visual da figura em questão, pode-se notar que o método proposto supera os outros dois métodos de discretização com relação a diversidade (lembrando que como todos os métodos usam o mesmo número de *num\_SB*, isso implica que os métodos, aproximadamente, tem o mesmo custo computacional). Os valores de diversidade calculados pela Eq. (4.8) conduzem à mesma conclusão que a da inspeção visual, cujos valores de diversidade calculados foram de 1,03, 0,23, 0,58 e 0,28, para o método proposto e para as discretizações uniforme, logarítmica e aleatória, respectivamente.

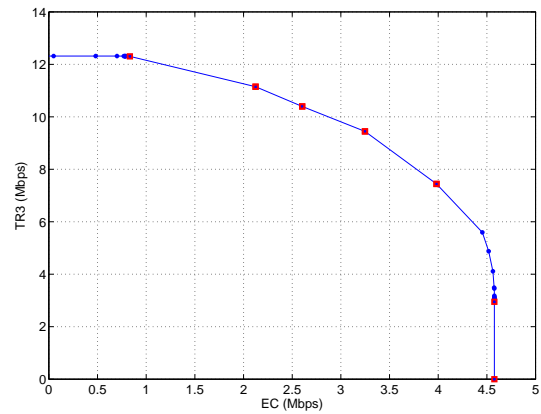
A Fig. 5.3 indica que o método proposto é capaz de fornecer uma alta diversidade com o mesmo custo computacional, comparado com os métodos de discretização. Esses resultados foram obtidos para um número específico de *num\_SB*.

E, é de interesse também observar como ficam os resultados das simulações quando o número de *num\_SB* é variado. Para esse fim, foram realizadas simulações onde os resultados de diversidade estão apresentados graficamente na Fig. 5.4. A Fig. 5.4 foi obtida variando *num\_SB* na abscissa e usando o mesmo cenário com 2-usuários apresentado na Fig. 5.2. Para o método proposto, foi fixado o tamanho da população em  $Z = 20$  e foi medida a diversidade para cada nova geração. Para os métodos de discretização uniforme, logarítmico e aleatório os resultados foram gerados variando-se uniformemente  $G$  de 10 até 35, com passo de 5. A Fig. 5.4 mostra que o método proposto *diverseSB* obteve um desempenho superior em comparação com os outros métodos de discretização. Vale ressaltar que os métodos uniforme e logaritmo tendem a convergir a medida que o número de *num\_SB* vai aumentando, o que faz sentido, pois a partir de um determinado valor de *num\_SB* a região de taxa vai ser, praticamente, toda explorada. O método de discretização aleatório obteve os melhores resultados em comparação aos métodos de discretização uniforme e logarítmico, porém

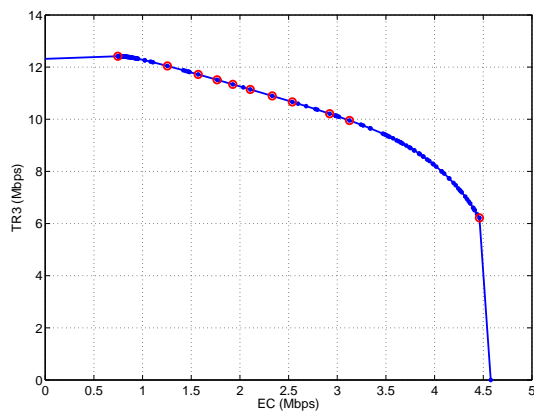




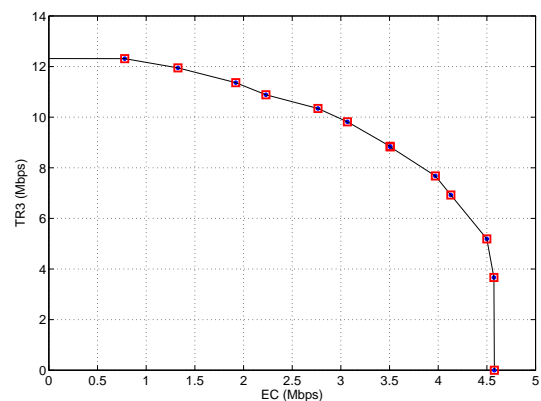
(a) Discretização Uniforme



(b) Discretização Logarítmica



(c) Discretização Aleatória



(d) Método Proposto

Fig. 5.3: Pontos de operação para o cenário de 2 usuários usando (a) discretização uniforme, (b) discretização logarítmica, (c) discretização aleatória e (d) método proposto (*diverseSB*). Todos os quatro métodos realizam 144 *num\_SB*, de onde  $Z = 12$  pontos são selecionados (alguns pontos não podem ser distinguíveis visualmente - estão muito próximos ou se sobrepõem), tal com em (d).

dada sua natureza aleatória pode-se realizar a mesmas simulações e obter melhores ou piores resultados, o que o torna um método não muito confiável. O método proposto *diverseSB* usa de valores probabilidade para ficar ajustando os valores dos pesos, por isso dificilmente duas simulações gerem resultados idênticos, mas a diferença é que em cada geração os bons resultados sempre serão preservados, devido a característica de elitismo do NSGA-II, e os operadores genéticos sempre buscarão encontrar melhores indivíduos e dessa forma, o conjunto de pesos em cada geração é atualizado, diferente dos métodos de discretização onde o resultado é gerado com um conjunto fixo de valores de pesos  $\omega$ .

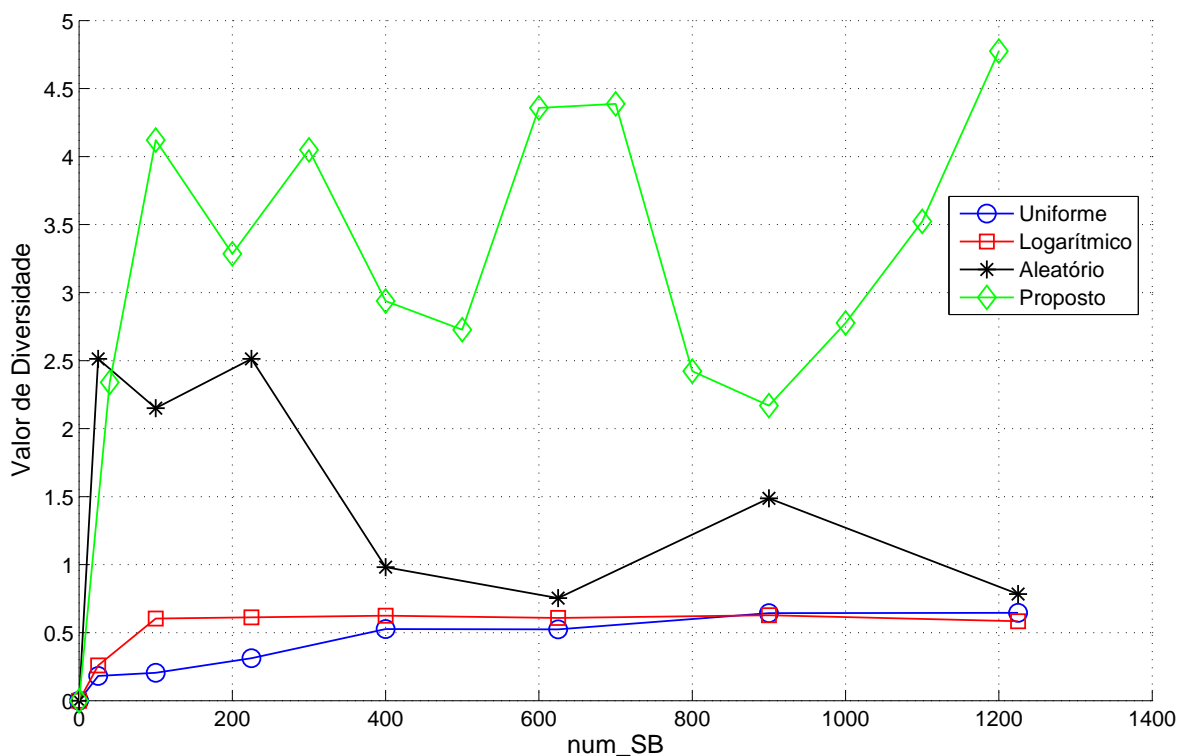


Fig. 5.4: Diversidade para diferentes valores de  $num\_SB$ . Na figura são identificados os métodos de discretizações: *Uniforme*, *Logaritmo*, *Aleatório* e o método proposto *diverseSB* como *Proposto*.

## 5.2 Simulações com 3-usuários

O cenário para a simulação com 3-usuários é apresentado na Fig. 5.5. Para realizar as simulações com os métodos de discretização, foi adotado  $G = 10$  pontos, o qual corresponde a 1.000  $num\_SB$ . Para o proposto método *diverseSB*, o tamanho adotado para a população foi de 100 indivíduos, com 10 gerações (o que resulta em 1.000  $num\_SB$ ). Os resultados são mostrados na Fig. 5.6.

As diversidades calculadas com a métrica Eq. (4.8) são iguais a 0,47, 0,3, 0,15 e 0,36 para o método proposto e para os métodos de discretização uniforme, logarítmica e aleatória, respectivamente. Pontos para formação de algumas facetas foram adicionados para facilitar a visualização da Fig. 5.6.

Novamente, com os resultados obtidos de diversidade e pela observação da Fig. 5.6, o método proposto *diverseSB* supera os resultados obtidos pelos outros métodos de discretização.

## 5.3 Simulações para cenários com mais de 3 usuários

Esta subseção apresenta os resultados obtidos com as simulações para os casos com 4, 5 e 6 usuários ( $N = 4, 5$  e 6).

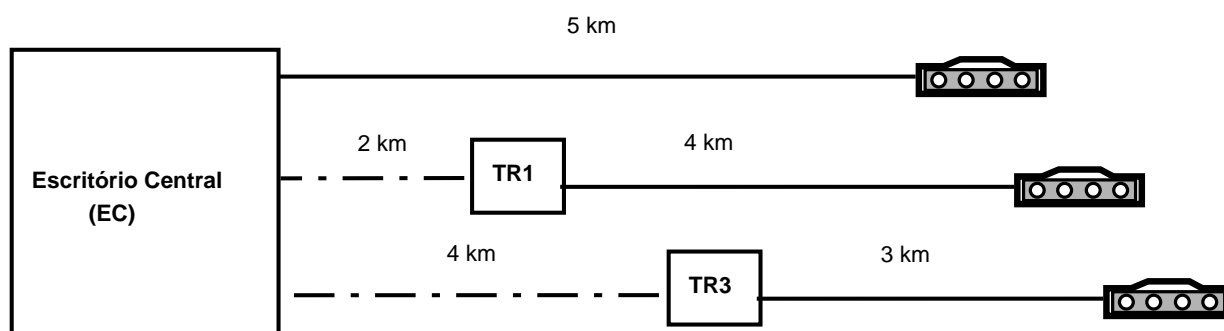
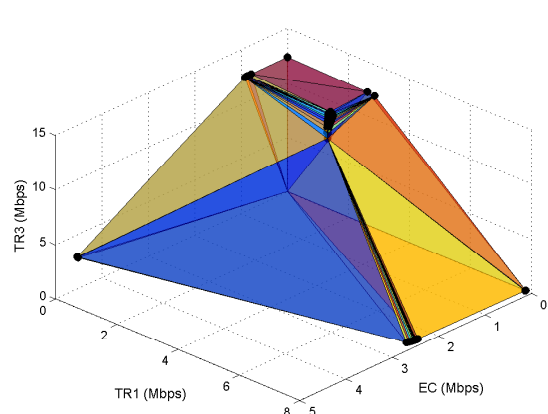
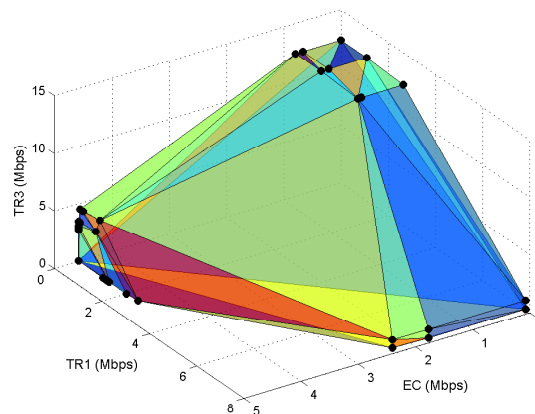


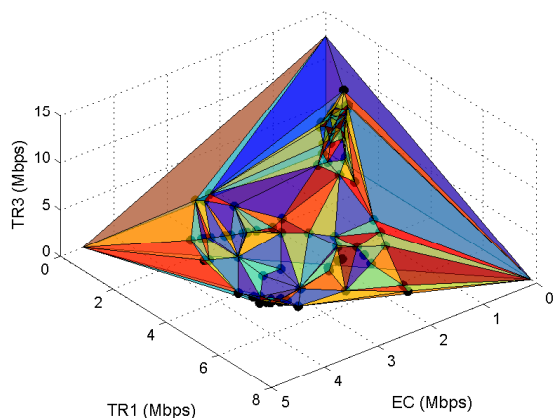
Fig. 5.5: Cenário DSL com 3-usuários.



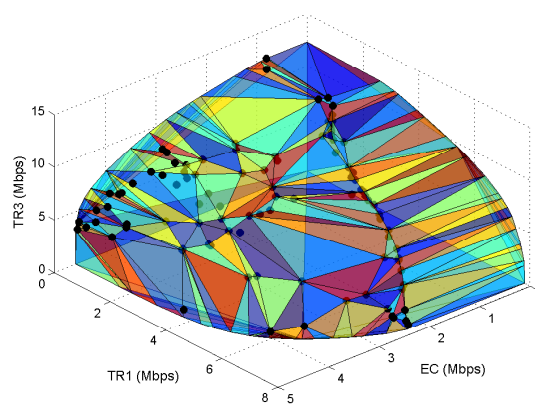
(a) Discretização Uniforme



(b) Discretização Logarítmica



(c) Discretização Aleatória



(d) Método Proposto

Fig. 5.6: OS resultados para as simulações com 3-usuários, (a) discretização uniforme, (b) discretização logarítmica, (c) discretização aleatória e (d) método proposto *diverseSB*. Todos os quatro métodos usaram 1.000  $num\_SB$  e forneceram  $Z = 100$  pontos de operação.

A primeira simulação é para 4 usuários, o cenário é apresentado na Fig. 5.1. O método proposto *diverseSB* usa 10.000 *num\_SB* (500 indivíduos e 20 gerações). As simulações com os métodos de discretização foram ajustas para o mesmo número de *num\_SB* e com  $G = 10$ . A diversidade calculado com a Eq. (4.8) para o método proposto e os métodos com as discretizações uniforme, logarítmica e aleatória foram 0,27, 0,10, 0,18 e 0,21, respectivamente.

De maneira similar, simulações para os cenários com 5 e 6 usuários foram realizadas, para os cenários de simulações foram adicionadas mais linhas ligadas ao EC da Fig. 5.1. As simulações realizadas para os métodos de discretização foram ajustas com  $G = 4$ , o que conduz a 1.024 e 4.096 *num\_SB*, respectivamente. O método proposto foi configurado com  $Z = 16$  indivíduos e 64 gerações, resultando em 1.024 *num\_SB*, para as simulações com 5 usuário e, com  $Z = 32$  indivíduos e 128 gerações (4.096 *num\_SB*) para o cenário com 6 usuários.

A diversidade calculada para os resultados da simulação do cenário com 5 usuários usando a Eq. (4.8) foram 0,0056, 0,0039, 0,0042 0,004 para o método proposto e para as discretizações uniforme, logarítmica e aleatória, respectivamente.

Para o cenário com 6 usuários, os resultados de diversidade forma 0,022, 0,01, 0,0019 e 0,001 para o método proposto e para as discretizações uniforme, logarítmica e aleatória, respectivamente.

A Fig. 5.7 apresenta um gráfico que apresenta um sumário dos resultados obtidos para  $N$  igual a 2, ..., 6, onde pretende-se facilitar a análise comparativa dos métodos apresentados. Por questões de conveniência, os valores exibidos de diversidade foram todos normalizados pelo método proposto. Pode-se observar que como indicado na figura, para esses casos, o método proposto sempre apresenta uma diversidade superior aos das obtidas pelos métodos de discretização uniforme, logarítmica e aleatória.

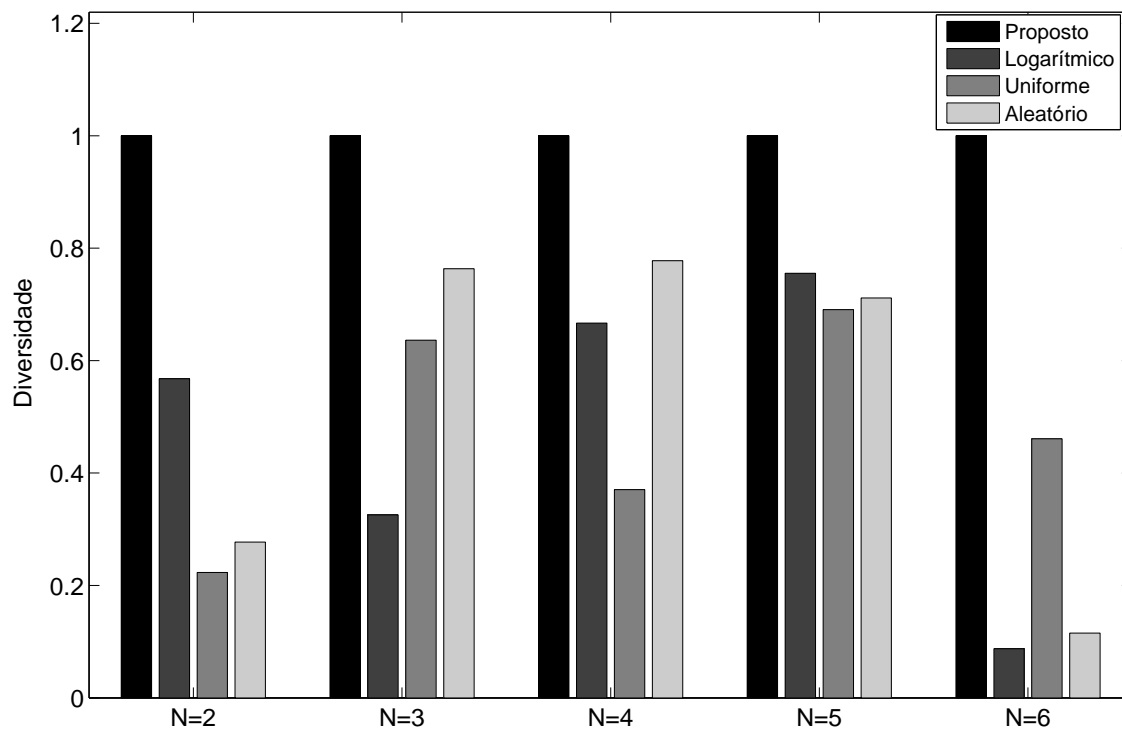


Fig. 5.7: Sumário dos resultados de diversidade obtidos com as simulações apresentadas, onde os valores foram normalizados pelos resultados do método proposto.

# Capítulo 6

## Conclusão

Neste trabalho são apresentados dois métodos, um para determinar um conjunto de pontos de operação denominado de *diverseSB* e o segundo é um método de avaliação da diversidade de soluções não-dominadas obtidas a partir do primeiro método.

Baseado na geometria computacional, foi proposto um método para avaliar diversidade das soluções, onde a partir dos desvios padrões das áreas das facetas compostas por pontos pertencentes as soluções, é possível avaliar a diversidade da soluções final. Esse método uso o algoritmo Qhull disponível em [40] para realizar o cálculo das áreas das facetas e a partir dos valores das áreas determinou-se o desvio padrão delas e, quanto menor o valor do desvio padrão melhor é a solução, ou seja a solução tem melhor diversidade.

O método *diverseSB* foi usado para resolver o problemas de balanceamento de espectro com conjunto de soluções diversas (DSSB), aplicado às linhas digitais de assinante (DSL), utilizando um processo híbrido composto de um algoritmo evolucionário multiobjetivo (MOEA), especificamente, o algoritmo genético por ordenamento por não-dominância (NSGA-II), baseado em uma otimização multiobjetivo e que usa um algoritmo de balanceamento de espectro.

Os resultados obtidos por simulações mostraram que, para uma dada diversidade, o custo computacional para determinar os pontos de operação com diversidade usando o algoritmo *diverseSB* proposto é muito menor que os métodos de busca usando discretização uniforme e não-uniforme. A complexidade do algoritmo *diverseSB* seria a complexidade do algoritmo NSGA-II mais  $I$  (número de gerações) vezes a complexidade do algoritmo de balanceamento de espectro. O NSGA-II tem complexidade de  $\mathcal{O}(FZ^2)$ , onde  $F$  é o número de funções objetivos, para este trabalho  $F = 1$ , e  $Z$  é o tamanho da população. Já, a complexidade do algoritmo de balanceamento de espectro depende de qual algoritmo será adotado, por exemplo, usando o SCALE, que é um algoritmo sub-ótimo e de baixa complexidade, sua complexidade é de  $\mathcal{O}(NK \log K)$ , onde  $N$  é o número de usuário DSL e  $K$  é o número de tons, para ADSL  $K = 256$  e para VDSL  $K = 4096$ . Portanto, comparando as complexidades do NSGA-II e do SCALE, nota-se que como a complexidade do NSGA-II é muito menor que a do SCALE. Assim, o algoritmo *diverseSB* proposto tem complexidade próxima a do SCALE,

e generalizando, a complexidade do *diverseSB* aproxima-se da complexidade do algoritmo de balanceamento do espectro adotado multiplicada pelo número de vezes que foi executado ( $num\_SB$ ), ou seja,  $\mathcal{O}(num\_SB \times SB_{complexity})$ , onde  $SB_{complexity}$  refere-se a complexidade do algoritmo de balanceamento de espectro adotado na simulação. Veja a Tabela 2.1 os valores dos mais populares algoritmos de balanceamento de espectro.

Já a complexidade dos métodos discretização de uniforme e não-uniforme de  $\omega$  (usado na soma ponderada) depende de  $G$  (o número de pontos de discretização) e  $N$  (número de usuários), segundo a relação exponencial  $G^N$ , multiplicada pela complexidade do algoritmo de balanceamento de espectro adotado, o que torna o procedimento de discretização inviável.

Considerando a análise anterior, quando se faz a comparação do desempenho, entre os métodos *diverseSB* e o de discretização, de maneira que esses métodos sejam forçados a terem o mesmo custo computacional, a diversidade obtida pelo *diverseSB* é muito superior.

O método proposto *diverseSB* como utiliza algoritmo evolucionário para resolver o problema multiobjetivo, que consiste em encontrar um conjunto de pontos de operação, os valores de pesos ( $\omega$ ) adotados pelos algoritmos SB são ajustados a cada geração, de tal forma que possam melhorar a população de pontos, por isso espera-se que o método proposto tenha maior probabilidade para encontrar os melhores resultados, quando comparados com os métodos de discretização, o quais dependem da escolha de um conjunto fixo de valores de pesos.

Com o algoritmo *diverseSB* proposto é possível fornecer um mecanismo para as Concessionárias de Telefonia, as quais são os provedores do serviço DSL, realizarem o planejamento da oferta do serviço DSL, através da avaliação da capacidade da rede e criação de planos para melhorar seu uso.

### **Sugestões para outros trabalhos:**

Uma das dificuldades encontradas pelo desenvolvimento do método proposto é o da elevação do custo computacional a medida que o número de usuário aumenta, pois exige uma população de indivíduos maior para compor a região de taxa. Dessa forma, sugere-se como um trabalho futuro o uso de algoritmo genético para o desenvolvimento de um algoritmo de balanceamento de espectro mono-objetivo, assim espera-se que o custo computacional desse algoritmo seja inferior comparado com os algoritmos de balanceamento de espectro existentes, permitir que o algoritmo *diverseSB* proposto use apenas algoritmo evolucionário e tente reduzir o custo computacional para uma rede com muitos modems DSL.

Como as probabilidades de cruzamento e de mutação usados no algoritmo genético, tiveram seus valores fixos para todas as simulações. Uma combinação ideal de todos estes parâmetros permitirá ao algoritmo genético obter a maior probabilidade de sucesso. Portanto, propõe-se a realização de um trabalho no sentido de investigar a utilização de mecanismos adaptativos e autoadaptativos para ajustes dos parâmetros do algoritmo genético.

Outra sugestão para dar continuidade a este trabalho, seria implementar um algoritmo para solucionar DSSB usando otimização por enxame de partícula (PSO) e poderia se comparar o seu desempenho com o método proposto que utiliza o NSGA-II, apesar da existência de trabalhos que mostram que o NSGA-II tem um desempenho superior ao PSO [26]. Vale ressaltar que foi publicado na literatura um algoritmo mono-objetivo implementado com PSO [14], nesse trabalho os autores afirmam que o método usando PSO garante rápida convergência com poucas iterações e resolve o problema de maximização de taxa de maneira eficiente.



# Referências Bibliográficas

- [1] T. Starr, J. M. Cioffi, and P. J. Silverman. *Understanding Digital Subscriber Line Technology*. Prentice-Hall, 1999.
- [2] W. Yu, G. Ginis, and J. M. Cioffi. Distributed multiuser power control for digital subscriber lines. *IEEE Journal on Selected Areas in Communication*, volume 20:p. 1105–1115, June 2002.
- [3] E. Van Bogaert, T. Bostoen, J. Verlinden, and R. Suci. Dynamic spectrum management in practice. *Technology White Paper - Alcatel*, 2004.
- [4] K.J. Kerpez, D.L. Waring, S. Galli, J. Dixon, and P. Madon. Advanced DSL management. *IEEE Communications Magazine*, volume 41(9):p. 116– 123, September 2003.
- [5] Etienne Van den Bogaert, Tom Bostoen, Jan Verlinden, R. Cendrillon, and Marc Moonen. ADSL transceivers applying DSM and their nonstationary noise robustness. *EURASIP Journal on Applied Signal Processing*, Article ID 67686, 2006.
- [6] T. Starr, M. Sorbara, J. M. Cioffi, and P. J. Silverman. *DSL Advances*. Prentice-Hall, 2003.
- [7] R. Cendrillon, Wei Yu, Marc Moonen, Jan Verlinden, and Tom Bostoen. Optimal multiuser spectrum balancing for digital subscriber lines. *IEEE Transactions on Communications*, 54(5):922–933, May 2006.
- [8] R. Cendrillon and M. Moonen. Iterative spectrum balancing for digital subscriber lines. *IEEE International Conference on Communications, ICC '05*, 3:1937–1941, May 2005.
- [9] John Papandriopoulos and Jamie S. Evans. Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks. In *IEEE International Conference on Communications*, volume 7, pages 3270 – 3275, June 2006.
- [10] R. Cendrillon, J. Huang, Mung Chiang, and Marc Moonen. Autonomous spectrum balancing for digital subscriber lines. *IEEE Transactions on Signal Processing*, 55(8):4241–4257, August 2007.

- [11] G. Ginis and J.M. Cioffi. Vecteded transmission for digital subscriber line systems. *IEEE Communications Magazine*, volume 20(5):p. 1085–1104, june 2002.
- [12] Rodrigo Moraes, Aldebaro Klautau, J. Rius, B. Dortschy, and R. Zampolo. Optimal solution for the fixed margin problem in digital subscriber lines. In *International Symposium on Communications, Control and Signal Processing*, pages 1395–1399, 2008.
- [13] Kalyanmoy Deb. Single and multi-objective optimization using evolutionary computation. KanGAL Report No. 2004002, <http://www.iitk.ac.in/kangal/papers/2004002.pdf>, 2000.
- [14] Meiqin Tang, C. Long, and X. Guan. An Optimal Spectrum-balancing Algorithm for Digital Subscriber Lines Based on Particle Swarm Optimization. *Intenational Journal of Communication Systems*, 2008.
- [15] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.
- [16] Philip Golden, Herve Dedieu, and Krista Jacobsen. *Fundamentals of DSL Technology*. Auerbach Publications, Taylor & Francis Group, 2006.
- [17] R. Cendrillon. *Multi-user Signal and Spectra Co-Ordination for Digital Subscriber Lines*. PhD thesis, Katholieke Universiteit Leuven, December 2004.
- [18] R. Cendrillon, M. Moonen, W. Yu, J. Verlinden, and T. Bostoen. On the Optimality of Iterative Waterfilling. In *TIE1.4/2003-325, San Diego, California, USA*, 2003.
- [19] R. Cendrillon, Marc Moonen, Jan Verliden, Tom Bostoen, and Wei Yu. Optimal Multi-user Spectrum Management for Digital Subscriber Lines. In *IEEE International Conference on Communications (ICC)*, volume 1, pages 1–5, 2004.
- [20] Jianwei Huang, R. Cendrillon, and Mung Chiang. Autonomous spectrum balancing (ASB) for frequency selective interference channels. In *IEEE International Symposium on Information Theory (ISIT), Seattle, Washington, USA*, July 2006.
- [21] R. Linden. *Algoritmos Genéticos*. Brasport, 2008.
- [22] Melanie Mitchell. *A introduction to Genetic Algorithms*. Massachusetts Institute of Technology, 1997.
- [23] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. Technical report, Indian Institute of Thecnology, 1994.
- [24] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

- [25] Kim F. Man, K. S. Tang, and S. Kwong. *Genetic algorithms: concepts and designs*. Springer, 1<sup>a</sup> ed. edition, 1999.
- [26] Marco José de Sousa. *Síntese de grades de Bragg em fibra: técnicas de aceleração e codificação para algoritmos evolucionários*. PhD thesis, UFPA, 2008.
- [27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182 – 197, April 2002.
- [28] Rodrigo Moraes, Aldebaro Klautau, B. Dortschy, and J. Rius. Semi-Blind Power Allocation for Digital Subscriber Lines. In *IEEE International Conference on Communications, ICC '08*, pages 1420–1425, 2008.
- [29] M. Monteiro, N. Lindqvist, and A. Klautau. Spectrum balancing algorithms for power minimization in DSL networks. In *IEEE International Conference on Communications, ICC '09*, pages 1–5, June 2009.
- [30] S. Galli and K. Kerpez. Improved algorithms for single-ended loop make-up identification. In *IEEE International Conference on Communications ICC'06*, volume 1, pages 67–72, 2006.
- [31] Claudomiro S. Sales, Roberto Menezes, Fredrik Lindqvist, Joao Costa, Aldebaro Klautau, Klas Ericson, Jaume Rius I Riu, and Per Ola Borjesson. Line topology identification using multi-objective evolutionary computation. *IEEE Transactions on Instrumentation and Measurement*, 2009.
- [32] Fredrik Lindqvist, Neiva Lindqvist, Boris Dortschy, Per Ödling, Per Ola Börjesson, Klas Ericson, and Evaldo Pelaes. Crosstalk channel estimation via standardized two-port measurements. *EURASIP Journal on Advances in Signal Processing*, 2008. Article ID 916865, 14 pages, 2008.
- [33] J. C. Bezerra, A. Kautau, M. Monteiro, E. Pelaes, E. Medeiros, and B. Dortschy. An evolutionary algorithm for improved diversity in dsl spectrum balancing solutions. *EURASIP Journal on Advances in Signal Processing*, v. 2010, p. 1-12,, 2010.
- [34] J. C. Bezerra, A. Kautau, M. Monteiro, E. Medeiros, and B. Dortschy. Operating points for spectrum management in digital subscriber lines. *Patente Registrada na World Intellectual Property Organization (WIPO), Pub. No.: WO/2010/138043, International Application No.: PCT/SE2009/050631, Data da Publicação: 02/12/2010*.
- [35] J. Goodman and J. O'Rourke, editors. *The Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, 2nd edition, 2004.

- [36] C. Barber, D. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22:469–483, Dec. 1996.
- [37] T. Bäck, U. Hammel, and H. Schwefel. Evolutionary computation: Comments on the history and current state. volume 1, 1997.
- [38] R. Lui and W. Yu. Low-complexity near-optimal spectrum balancing for digital subscriber lines. In *IEEE International Conference on Communications, ICC '05*, volume 3, pages 1947–1951, May 2005.
- [39] Ching-Shih Tsou, Shih-Chia Chang, and Po-Wu Lai. *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, chapter Using Crowding Distance to Improve Multi-Objective PSO with Local Search. Itech Education and Publishing, 2007.
- [40] Qhull. Disponível em: [www.qhull.org](http://www.qhull.org), Último Acesso: 29/12/2010.

## Trabalhos Publicados Pelo Autor

1. J. C. Bezerra; A. Kautau; M. Monteiro; E. Pelaes; E. Medeiros; B. Dortschy. “An Evolutionary Algorithm for Improved Diversity in DSL Spectrum Balancing Solutions”. *EURASIP Journal on Advances in Signal Processing*, v. 2010, p. 1-12, 2010.
2. J. C. Bezerra; A. Kautau; M. Monteiro; E. Medeiros; B. Dortschy. “Operating Points for Spectrum Management in Digital Subscriber Lines”. *Patente Registrada na World Intellectual Property Organization (WIPO)*, Pub. No.: WO/2010/138043, International Application No.: PCT/SE2009/050631, Data da Publicação: 02/12/2010.
3. BEZERRA, J. C. ; GOMES, Ana Claudia ; KLAUTAU, Aldebaro ; PELAES, E. G. ; DORTSCHY, Boris. “The Influence of DSM on Legacy xDSL Services. In: 5th International Information and Telecommunication Technologies Symposium” (I2TS, 2006), Cuiabá. The Influence of DSM on Legacy xDSL Services, 2006.
4. J. C. Bezerra; E. Pelaes; A. Kautau; M. Monteiro; B. Dortschy; C. Muto; R. Moraes “Power Spectrum Balancing Solution for Digital Subscriber Lines”. *International Information and Telecommunication Technologies (I2TS’2008)*, Foz do Iguaçu, Brasil, 2008.
5. J. C. Bezerra; S. C. de Souza Filho ; R. B. Vaz “Gerenciamento de Espectro Multi-usuário no sistema DSL”. In: *17º Simpósio Internacional de Iniciação Científica da Universidade de São Paulo*, São Carlos, 2007.
6. J. C. Bezerra; D. O. de Souza; O. das N. Oliveira. “Análise da Técnica de Multiplexação Ortogonal por Divisão em Frequência (OFDM)”. *14º Simpósio Internacional de Iniciação Científica da Universidade de São Paulo*, São Paulo, 2006
7. J. C. Bezerra; H. J. C. Braga; N. S. L. Tembra; J. M. Gonçalves. “Gerenciamento Dinâmico do Espectro para Transmissão de Dados usando Par-Trançado”. *13º Simpósio Internacional de Iniciação Científica da USP*, 2005, São Carlos.

# Apêndice A

## Algoritmos DSM

Neste apêndice são apresentados os algoritmos DSM descritos no Capítulo 2.

### A.1 Algoritmo Iterative Water-Filling - IWF

Tabela A.1 apresenta o algoritmo IWF [17].

Tabela A.1: Algorithm 1 - Iterative Waterfilling.

<p><b>Repetis</b> <b>Para</b> cada usuário <math>n = 1, \dots, N</math> <b>Repetir</b> <math>s_k^n = s_k^n(\lambda_n), \forall k</math>, found using (2.8) <b>Se</b> <math>f_s \sum_k b_k^n &lt; R_n^{target}</math> e <math>\sum_k s_k^n \leq P_n</math>, então aumente <math>\lambda_n</math>, senão deiminua <math>\lambda_n</math> <b>Até convergir</b> <b>end</b> <b>Até convergência de Taxa</b></p>
--

### A.2 Optimal Spectrum Balancing - OSB

#### A.2.1 OSB para o caso de 2 usuário

No formato de algoritmo, o OSB teria quatro funções. A Tabela A.2 reproduz o algoritmo apresentado em [17].

O algoritmo proposto apresenta três loops. O loop externo varia  $w$  para encontrar o valor que alcança a melhor relação entre as taxas dos dois usuário, portanto maximizando a taxa do usuário 2 enquanto o usuário 1 alcança sua taxa alvo. E variando  $w$  de 0 a 1 permite que todos os pontos ótimos de operação na região de taxa sejam encontrados.

Tabela A.2: Algoritmo 2 - OSB para 2 users.

**Função Principal****for**  $w = 0, \dots, 1$ 

$$[s_1, s_2] = \text{optimize\_} \lambda_1(w)$$

**Fim****Função**  $[s_1, s_2] = \text{optimize\_} \lambda_1(w)$ 

$$\lambda_1^{max} = 1, \lambda_1^{min} = 0$$

**Enquanto**  $\sum_k s_k^1 > P_1$ 

$$\lambda_1^{max} = 2\lambda_1^{max}$$

$$[s_1, s_2] = \text{optimize\_} \lambda_2(w, \lambda_1^{max})$$

**Fim****repetir**

$$\lambda_1 = \frac{\lambda_1^{max} + \lambda_1^{min}}{2}$$

$$[s_1, s_2] = \text{optimize\_} \lambda_2(w, \lambda_1)$$

Se  $\sum_k s_k^1 > P_1$ , então  $\lambda_1^{min} = \lambda_1$ , senão  $\lambda_1^{max} = \lambda_1$ **Até convergir****Função**  $[s_1, s_2] = \text{optimize\_} \lambda_2(w, \lambda_1)$ 

$$\lambda_2^{max} = 1, \lambda_2^{min} = 0$$

**Enquanto**  $\sum_k s_k^2 > P_2$ 

$$\lambda_2^{max} = 2\lambda_2^{max}$$

$$[s_1, s_2] = \text{optimize\_} s(w, \lambda_1, \lambda_2^{max})$$

**Fim****repetir**

$$\lambda_2 = \frac{\lambda_2^{max} + \lambda_2^{min}}{2}$$

$$[s_1, s_2] = \text{optimize\_} s(w, \lambda_1, \lambda_2)$$

Se  $\sum_k s_k^2 > P_2$ , então  $\lambda_2^{min} = \lambda_2$ , senão  $\lambda_2^{max} = \lambda_2$ **Até convergir****Função**  $[s_1, s_2] = \text{optimize\_} s(w, \lambda_1, \lambda_2)$ **Para**  $k = 1, \dots, K$ 

$$[s_k^1, s_k^2] = \arg \max_{s_k^1, s_k^2} L_k(s_k^1, s_k^2, w, \lambda_1, \lambda_2)$$

tal que,  $0 \leq s_k^n \leq s_k^{n,max}, \forall n$ 

(resolvido com busca exaustiva bi-dimensional)

**Fim**

Um loop intermediário é usado para procurar  $\lambda_1$  e um loop externo para procurar por  $\lambda_2$ , e o método da Bisseção é usado em cada uma dessas buscas. Com um grande valor para  $\lambda_n$ , uma pequena quantidade de potência será alocada para o usuário  $n$ . Por outro lado, um  $\lambda_n$  próximo de zero conduz a uma alocação de uma quantidade maior de potência. A instrução “until convergence” significa que ambos  $\lambda_n = 0$  ou  $\sum_k s_k^n = P_n$  sejam alcançados.

### A.2.2 OSB para o caso de $N$ usuários

O OSB para  $N$  usuário é apresentado na Tabela A.3 [17].

Tabela A.3: Algoritmo 3 - OSB ( $N$  usuários).

<p><b>repetir</b>  <b>Para</b> cada <math>k: s_k^1, \dots, s_k^N = \operatorname{argmax}_{s_k^1, \dots, s_k^N} L_k</math>          (Resolvido com busca N-dimensional)  <b>Para</b> cada <math>n: w_n = [w_n + \epsilon(R_n^{target} - f_s \sum_k b_k^n)]^+;</math>  <b>Para</b> cada <math>n: \lambda_n = [\lambda_n + \epsilon(\sum_k s_k^n - P_n)]^+;</math>  <b>Até convergir</b></p>
---

## A.3 Iterative Spectrum Balancing - ISB

O sumário dos estágios do algoritmo ISB é apresentado na Tabela A.4 [8]. No loop interno é executado uma busca unidimensional exaustiva e se repete até que o usuário  $n$  alcance sua taxa alvo e sua potência disponível. A próxima interação é aplicada através do loop externo para cada usuário  $n$ , o qual termina quando as PSD's de todos os usuário converge.

Tabela A.4: Algoritmo 4 - ISB.

<p><b>repetir</b>  <b>for</b> <math>n = 1, \dots, N</math>  <b>repetir</b>  <b>Para</b> cada <math>k</math>: fixe <math>s_k^m \forall m \neq n</math>, então  <math>s_k^n = \operatorname{argmax}_{s_k^n} L_k;</math>          (resolvido com busca exaustiva unidimensional)  <math>w_n = [w_n + \epsilon(R_n^{target} - \sum_k b_k^n)]^+;</math>  <math>\lambda_n = [\lambda_n + \epsilon(\sum_k s_k^n - P_n)]^+;</math>  <b>Até convergir</b>  <b>Fim</b>  <b>Até convergir</b></p>
--

## A.4 Autonomous Spectrum Balancing - ASB

O algoritmo do ASB é apresentado na Tabela A.5.



Tabela A.5: Algoritmo ASB-I.

<p><b>Repetir</b>  <b>Para</b> cada usuário <math>n = 1, \dots, N</math>  <b>Repetir</b>  <b>Para</b> cada usuário <math>k = 1, \dots, K</math>  <math>s_k^{n,I} = \arg \max_{s_k^n \geq 0} L_k^n</math>  <math>\lambda^n = [\lambda^n + \epsilon_\lambda (\sum_k s_k^{n,I} - P^n)]^+;</math>  <math>w^n = [w^n + \epsilon_w (R^{n,target} - \sum_k s_k^{n,I})]^+;</math>  <b>Até a convergência de taxa</b>  fim  <b>Até a convergência de taxa</b></p>
--

## A.5 Algoritmo Successive Convex Approximation for Low Complexity - SCALE

O algoritmo SCALE em forma completa é apresentado na Tabela A.6. O algoritmo consiste de 3 laços, onde o mais interno realiza a alocação de potência e é de responsabilidade dos modems. O laço intermediário é de convergência de PSDs. O processo de repete até convergência. Vale ressaltar que o  $\alpha_n^k$  deve ser apenas calculado se o resto da divisão inteira do contador de iterações por um número  $D > 1$  for zero. Mais detalhes do algoritmo SCALE podem ser encontrados em [9].

Tabela A.6: Algoritmo Successive Convex Approximation for Low Complexity (SCALE)

<p><b>Repetir</b>  <math>\alpha_n^k = 1 \quad \forall n, k</math>  Definir contador <math>c = 1</math>  <b>Repetir</b>  <b>Se</b> <math>\text{mod}(c, D) = 0</math>,      Calcular <math>\alpha_n^k \quad \forall n, k</math>      Calcular <math>\Phi_n^k</math> e <math>\Psi_n^k \quad \forall n, k</math>      <b>Repetir</b>          <math>p_n^k = \frac{w_n \alpha_n^k}{\lambda_n + \Psi_n^k} \quad \forall n, k</math>          <math>\lambda_n = \max(\lambda_n + \epsilon_p (\sum_k p_n^k - P_n^{max}), 0) \quad \forall n</math>      <b>até a convergência de potências</b>      <math>c = c + 1</math>  <b>até a convergência de PSDs</b>      <math>w_n = \max(w_n + \epsilon_r (R_n^{alvo} - f_s \sum_k b_n^k), 0) \quad \forall n &gt; 1</math>      <math>w_1 = 1 - \sum_2^N w_n</math>  <b>até a convergência de taxas</b></p>
---