

ALGORITMOS HEURÍSTICOS APLICADOS À DETECÇÃO MULTIUSUÁRIO DS/CDMA

Fernando Ciriaco, Taufik Abrão e Paul Jean E. Jeszensky

Resumo - Neste trabalho são analisados e comparados nove algoritmos heurísticos baseados na teoria da evolução genética e de busca local aplicados ao problema da detecção multiusuário (MuD-*Multiuser Detection*) para sistemas DS/CDMA (*Direct Sequence/Code Division Multiple Access*) em canais com desvanecimento Rayleigh Plano. A eficiência destes algoritmos é comparada através do compromisso desempenho *versus* complexidade computacional, sendo esta complexidade expressa em termos do número de operações necessárias para se alcançar o desempenho do detector de máxima verossimilhança (ML - *Maximum Likelihood*). Analisa-se também a perda de desempenho quando há ocorrência de erros na estimativa do canal e na condição de sistemas sobrecarregados. São três as principais contribuições deste trabalho: a) aplicação de algoritmos de busca local, do algoritmo de programação evolucionária e do algoritmo de programação evolucionária com clonagem ao problema MuD em canais com desvanecimento Rayleigh; b) comparação sistemática de nove algoritmos heurísticos sob uma mesma base de simulação e de análise de desempenho e c) análise de complexidade mais precisa que as encontradas atualmente na literatura para o problema MuD, expressando-a em termos do número de operações computacionais.

Palavras-chave: Detecção multiusuário, algoritmos evolucionários, algoritmos de busca local, complexidade computacional, detector de máxima verossimilhança.

Abstract - The characteristics of nine heuristic algorithms based on genetic evolution theory and local search, applied to a DS/CDMA (*Direct Sequence/Code Division Multiple Access*) multi-user detection problem in Flat Rayleigh fading channel are introduced. The algorithms comparisons through the performance versus computational complexity tradeoff are carried out. The computational complexity is determined in terms of the number of operations to reach the Maximum Likelihood (ML) performance. The effects on the performance of the errors in the estimates of the channel coefficients are also considered, as well as the performance degradation due to overload. There are three contributions in this work: a) a compared analysis of the evolutionary, evolutionary with cloning and local search algorithms applied to MuD detector in Flat Rayleigh channels; b) a systematic comparison of these algorithms in terms of the same simulation scenario and performance analysis base; c) a more fair and precise complexity analysis for the MuD detector in terms of the

number of the computational operations.

Keywords: Multiuser detection, evolutionary algorithms, local search algorithms, computational complexity, maximum likelihood detector.

1. INTRODUÇÃO

O sinal recebido por um detector convencional (*Conventional Detector*), constituído de um banco de filtros casados (*MFB-Matched Filter Bank*) em um sistema DS-CDMA síncrono em canal Rayleigh Plano não pode ser recuperado de forma ótima, pois é afetado pela interferência de múltiplo acesso (*MAI-Multiple Access Interference*) e pelo efeito *near-far*, resultando em um sistema cuja capacidade está bem abaixo da capacidade do canal [1, 2].

Nas últimas duas décadas, uma grande variedade de detectores multiusuário foram propostos na literatura no intuito de melhorar o desempenho obtido com o detector convencional. O melhor desempenho é alcançado com o detector de máxima verossimilhança. Este detector procura estimar o vetor transmitido de forma a maximizar a probabilidade deste ter sido enviado, dado que um sinal foi recebido, onde o sinal processado estende-se por toda a mensagem, com a hipótese de todas as mensagens transmitidas serem equiprováveis. Porém, o detector ML possui uma complexidade computacional que cresce exponencialmente com o número de usuários, tornando-o inviável para implementação [2]. Faz-se necessário, portanto, investigações na área de detecção multiusuário sub-ótimas que atendam aos critérios de alto desempenho e baixa complexidade.

Na literatura, apesar de existirem vários trabalhos empregando procedimentos aproximativos para a detecção multiusuário, a maioria das investigações está restrita a canais AWGN (*Additive White Gaussian Noise*) síncronos [3, 4, 5, 6, 7]. Até o momento somente trabalhos que analisam o desempenho do algoritmo genético aplicado à detecção multiusuário em canais com desvanecimento estão disponíveis na literatura [8].

Este trabalho faz uma comparação sistemática entre nove algoritmos heurísticos baseados na teoria da evolução genética e de busca local, aplicáveis ao problema da detecção multiusuário em canal com desvanecimento Rayleigh Plano.

Adicionalmente, a comparação da complexidade computacional destes nove algoritmos é realizada de forma inédita, sendo expressa através do número de operações computacionais que cada receptor necessita para a demodulação da informação.

Este trabalho divide-se em 7 seções. Inicialmente, a seção 2 trata da descrição do modelo matemático de um sistema DS/CDMA em canal Rayleigh Plano síncrono. Na seção

Fernando Ciriaco e Taufik Abrão, Universidade Estadual de Londrina, Departamento de Engenharia Elétrica, Londrina, Brasil. E-mails: fernandociriaco@onda.com.br, taufik@uel.br.
Paul Jean E. Jeszensky, Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Telecomunicações e Controle, São Paulo, Brasil. E-mail: pjj@lcs.poli.usp.br.

3 é descrito o problema combinatório para a detecção multiusuário ótima, evidenciando a aplicabilidade de métodos combinatórios sub-ótimos. Na seção 4 são apresentadas as características e a descrição dos principais algoritmos heurísticos evolucionários e de busca local. Na seção 5 são apresentados resultados numéricos para diferentes condições de operação do sistema via simulação Monte-Carlo. Na seção 6 são determinadas as expressões analíticas gerais para a complexidade computacional dos algoritmos, obtendo-se o número de operações para cada detector, em função dos parâmetros específicos de cada algoritmo. Por fim, na seção 8 são apresentadas as principais conclusões deste estudo.

2. MODELO DO SISTEMA

Em um sistema DS-CDMA com modulação BPSK (*Binary Phase-Shift Keying*) em canal com desvanecimento, o sinal que chega ao receptor pode ser descrito, em banda base, por:

$$r(t) = \sum_{k=1}^K A_k b_k q_k(t - \tau_k) * h(t) + n(t) \quad (1)$$

onde K é o número de usuários ativos no sistema, $b_k \in \{-1, +1\}$ é o bit de informação transmitido, q_k é a seqüência de assinatura binária bipolarizada atribuída ao k -ésimo usuário. $A_k = \sqrt{E_{b_k}/T_b}$ é a amplitude do sinal transmitido pelo k -ésimo usuário, onde T_b é o período de bit e E_{b_k} a energia de bit do k -ésimo usuário, $t \in [0, T_b]$, τ_k é o atraso inter-usuários. $h(t)$ é a resposta impulsiva do canal e $n(t)$ representa o AWGN com densidade espectral de potência bilateral igual a $N_0/2$.

Considerando um sistema síncrono, $\tau_k = 0 \forall k$, e um canal não seletivo em frequência, a resposta impulsiva, $h(t)$, pode ser caracterizada por um canal Rayleigh Plano, podendo ser reescrita como:

$$h(t) = c_k \delta(t) = \beta_k e^{j\phi_k} \delta(t) \quad (2)$$

onde c_k indica o coeficiente complexo do canal para o k -ésimo usuário. β_k denota o módulo de c_k com uma distribuição Rayleigh e ϕ_k a fase de c_k com uma distribuição uniforme no intervalo $[0, 2\pi)$; dessa forma, pode-se reescrever a equação (1) como:

$$r(t) = \sum_{k=1}^K A_k b_k c_k q_k(t) + n(t) \quad (3)$$

O sinal na saída de um banco de filtros casados (CD) pode ser expresso como:

$$y_k = \int_0^{T_b} r(t) q_k(t) dt = A_k b_k c_k + \sum_{j \neq k} A_j b_j c_j \lambda_{k,j} + n_k \quad (4)$$

onde n_k é o ruído AWGN filtrado para o k -ésimo usuário e $\lambda_{k,j}$ denota o k, j -ésimo elemento da matriz de correlação \mathbf{R} , dado por [2]:

$$\lambda_{k,j} = \frac{1}{N} \int_0^{T_b} q_k(t) q_j(t) dt \quad (5)$$

onde $N = T_b/T_c$ é o ganho de processamento e T_c é o período de chip.

Reescrevendo a equação (4), utilizando notação vetorial, tem-se:

$$\mathbf{y} = \mathbf{R} \mathbf{A} \mathbf{c} \mathbf{b} + \mathbf{n} \quad (6)$$

onde $\mathbf{y} = [y_1, y_2, \dots, y_K]^T$, $\mathbf{A} = \text{diag}[A_1, A_2, \dots, A_K]$, $\mathbf{c} = \text{diag}[c_1, c_2, \dots, c_K]$, $\mathbf{b} = [b_1, b_2, \dots, b_K]^T$, $\mathbf{n} = [n_1, n_2, \dots, n_K]^T$ e $\text{diag}[\cdot]$ é o operador diagonal.

3. DESCRIÇÃO DO PROBLEMA

Em [2] foi mostrado que uma solução ótima para recuperar os bits de informação de (3) consiste em estimar a informação transmitida utilizando a saída de um detector de máxima verossimilhança:

$$\hat{\mathbf{b}} = \arg \left\{ \max_{\mathbf{b} \in \{\pm 1\}^K} \left[2\Re \{ \mathbf{y}^T \mathbf{c}^H \mathbf{A} \mathbf{b} \} - \mathbf{b}^T \mathbf{c} \mathbf{A} \mathbf{R} \mathbf{A} \mathbf{c}^H \mathbf{b} \right] \right\} \quad (7)$$

onde $(\cdot)^H$ indica operador hermitiano¹ e $\Re\{\cdot\}$ o operador parte real.

O detector multiusuário ótimo consiste na busca do melhor vetor de bits de dados em um conjunto com todas as possibilidades, ou seja, 2^K vetores de bits candidatos. O conjunto com todas as possibilidades cresce exponencialmente com o número de usuários, sendo conhecido na literatura como um problema de combinação NP-completo [9], no qual os algoritmos tradicionais de otimização combinatória são ineficientes.

Métodos heurísticos aplicados a esse tipo de problema têm se tornado atraentes, pois permitem obter soluções sub-ótimas, porém muito próximas da solução ótima, para problemas de combinação em um curto tempo e espaço de busca, diminuindo assim a complexidade computacional em relação aos métodos exaustivos.

Sob a restrição de um espaço de busca, todos os algoritmos heurísticos procuram melhores soluções seguindo uma função objetivo, capaz de quantificar a tendência de melhoria na direção da solução ótima. Esta função é denominada função custo (*fitness value*) e no contexto da detecção multiusuário é dada pela função de verossimilhança:

$$f(\mathbf{s}) = 2\Re \{ \mathbf{y}^T \mathbf{c}^H \mathbf{A} \mathbf{s} \} - \mathbf{s}^T \mathbf{c} \mathbf{A} \mathbf{R} \mathbf{A} \mathbf{c}^H \mathbf{s} \quad (8)$$

onde \mathbf{s} é um vetor de bits candidatos de dimensão idêntica à \mathbf{b} .

Cada algoritmo heurístico aplicado ao problema da detecção multiusuário busca maximizar a equação (8) através de um vetor de bits, \mathbf{s} , cujo desempenho médio correspondente aproxima-se daquele obtido por um detector ML.

O esquema síncrono de transmissão-recepção adotado neste trabalho é mostrado na figura 1. O receptor é constituído de um banco de filtros casados, seguido de um algoritmo heurístico com o intuito de maximizar o desempenho do sistema DS/CDMA em termos de taxa de erro de bits (*BER*).

¹ Equivalente às operações matriciais transposição e conjugação.

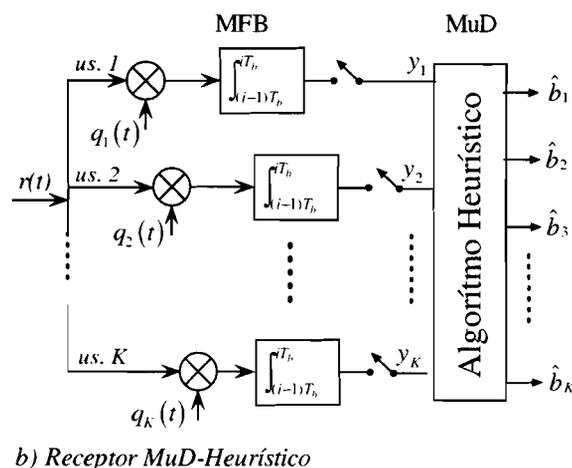
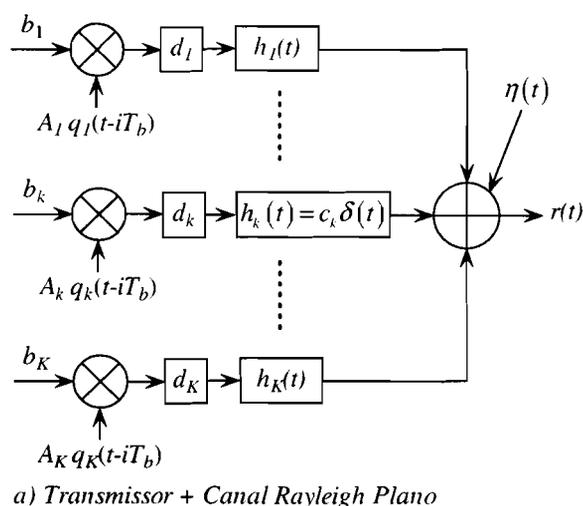


Figura 1. (a) Modelo de transmissão e (b) recepção utilizados na modelagem do sistema.

4. ALGORITMOS HEURÍSTICOS

Em comparação com as técnicas exatas, os algoritmos heurísticos não garantem encontrar uma solução ótima após atingirem um critério de parada; mas estes têm demonstrado alta eficiência em problemas de larga combinação para casos práticos, além de poderem ser modificados facilmente, adaptando-se ao problema analisado [3, 10].

Algoritmos heurísticos podem ser divididos em duas grandes classes: os analógicos e os estocásticos. Na primeira classe encontram-se os algoritmos genéticos [11, 12], os de programação evolucionária [13] e Clonalg [14]. Na segunda classe encontram-se o l-opt LS e k-opt LS [10, 15], o recozimento simulado [16, 17, 18], busca Tabu clássica [19] e busca Tabu reativa [20].

4.1 ALGORITMOS EVOLUCIONÁRIOS

Os algoritmos evolucionários constituem métodos de busca não determinísticos baseados em mecanismos de seleção e evolução natural seguindo a teoria da evolução das espécies de Darwin [21]. Esta teoria descreve o comporta-

mento complexo que ocorre na interação entre os diversos níveis biológicos: a célula, o organismo, o indivíduo e a população.

A mais extensa coleção de teorias evolucionárias aceita é chamada de paradigma neo-Darwiniano, que explica a história da vida pela ação de processos físicos e operadores genéticos em populações ou espécies. Estes processos são conhecidos por reprodução, perturbação, competição e seleção.

Para cada problema deve-se adaptar a nomenclatura inerente a cada algoritmo evolucionário, pois originalmente os algoritmos foram propostos a partir de uma interpretação biológica. No contexto da detecção multiusuário, para o algoritmo Genético e Programação Evolucionária um vetor de bits candidato deve ser interpretado como um indivíduo e cada bit deste vetor como um gene. Para o algoritmo Clonalg, um vetor de bits candidato é interpretado como um anticorpo e os parâmetros do sistema necessários para o cálculo da equação (8) são interpretados como os antígenos.

Em qualquer algoritmo evolucionário, as estratégias utilizadas para a diversificação do universo de busca são baseadas no processo de perturbação, que ocorre na etapa de reprodução das células. As perturbações são exercidas pelos operadores genéticos, sendo as mais conhecidas denominadas mutação e crossover. O critério de mutação, ilustrado na figura 2, consiste na troca de um ou mais genes escolhidos aleatoriamente por outros que possuem características diferentes.



Figura 2. Operador mutação.

No caso de alfabeto binário bipolarizado, -1 e +1, quando a perturbação for pequena, não haverá mudança (mutação) do gene [22, 23, 24]. No entanto, quando a perturbação for grande o suficiente para trocar o sinal do gene, ocorrerá mutação. Esta perturbação pode ser obtida através de uma distribuição estatística específica. Neste trabalho adotou-se uma distribuição Gaussiana:

$$\text{novoindividuo} = \text{sign}(\text{genitor} + \mathcal{N}(0, \sigma^2)) \quad (9)$$

onde $\mathcal{N}(0, \sigma^2)$ representa uma variável aleatória Gaussiana com média 0 e desvio padrão σ . Este desvio padrão está intimamente ligado à taxa média de mutação. No caso binário bipolarizado, o desvio padrão representará uma taxa média de mutação, seguindo o comportamento mostrado na figura 3.

Para cada algoritmo evolucionário escolheu-se um desvio padrão que correspondesse a uma taxa média de mutação mais apropriada. Fica evidente que a mutação pode ocorrer em um ou mais pontos do vetor de bits candidato, ou ainda não ocorrer em nenhum, devido à característica estocástica do processo.

O critério de crossover utiliza dois indivíduos genitores, com o objetivo de formar novos indivíduos a partir da troca de

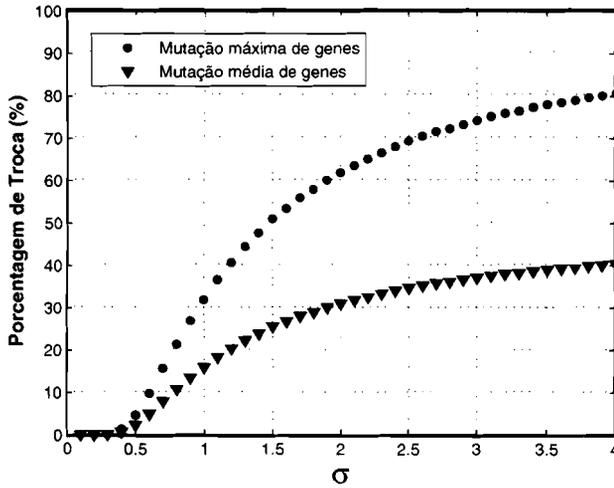


Figura 3. Porcentagem de troca dos genes em relação ao desvio padrão.

genes entre os indivíduos genitores. Neste trabalho, adotou-se *crossover* do tipo 1 ponto (*single-point crossover*) [11, 12], sendo este ponto escolhido aleatoriamente; um exemplo é mostrado na figura 4.

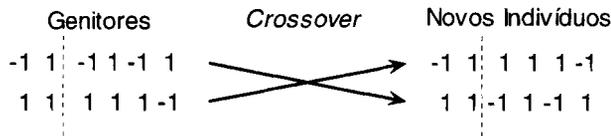


Figura 4. Operador *single-point crossover*.

O operador genético² proposto neste trabalho consiste em clonar (replicar) os indivíduos que possuem melhores genes, no intuito de produzir uma população altamente evoluída na geração posterior. Outra estratégia de intensificação utilizada consiste na tentativa de quantificar o número necessário de mutações que devem ocorrer na etapa de reprodução de forma que um novo indivíduo apresente um melhor conjunto de genes que seu genitor [23]. Melhorias no conjunto de genes representam incrementos no valor da função custo (8). Esta estratégia visa adaptar a matriz de desvio padrão, σ , do algoritmo de programação evolucionária através dos parâmetros E_b/N_0 (razão entre a energia de bit e a densidade espectral de potência do ruído, N_0), razão *near-far* (NFR-Near Far Ratio) e o próprio valor da função custo.

A razão NFR quantifica as disparidades de potência dos diversos usuários em um ambiente com interferência de múltiplo acesso. Desigualdades de potência fazem com que usuários com reduzidas potências no receptor sejam “sufocados” pelos usuários que chegam com maiores energias [2, 25].

Nos algoritmos evolucionários descritos neste trabalho, g indica a geração atual, G o número total de gerações e p o tamanho da população.

²Em muitos contextos denominado estratégia de intensificação.

4.1.1 PROGRAMAÇÃO EVOLUCIONÁRIA (EP)

O algoritmo heurístico programação evolucionária (EP - *Evolutionary Programming*) descreve as interações biológicas que ocorrem entre os indivíduos de uma população, utilizando apenas o operador genético mutação na etapa de diversificação do universo de busca. Neste caso, este operador atua no sentido de alterar uma característica fenotípica do genitor durante a etapa de transferência do material genético para um novo indivíduo (processo de reprodução).

Por utilizar apenas o operador de mutação como critério de diversificação e nenhum critério de intensificação, este algoritmo possui a menor complexidade computacional por geração dentre os evolucionários [13, 22]. Mas sua convergência é mais lenta, sendo necessárias gerações extras para atingir o desempenho desejado. Descreve-se a seguir o algoritmo EP aplicado à detecção multiusuário:

1. Utiliza-se a saída do detector convencional como o vetor de genes do indivíduo inicial.

$$s_1 = \hat{b}_{CD} = \text{sign}(y) \quad (10)$$

2. Os outros vetores de genes dos indivíduos que formam a população S da 1ª geração são determinados por um processo aleatório. Para $j = 2, \dots, p$:

$$s_j = U\{-1, +1\}^K \quad (11)$$

onde $U\{-1, +1\}^K$ é um vetor coluna binário obtido aleatoriamente e de tamanho K . A matriz população S , contendo genes iniciais, possui K linhas e p colunas (genitores) e pode ser expressa por:

$$S = [s_1, \dots, s_p] \quad (12)$$

3. Para cada $s_j, j = 1, \dots, p$, calcula-se o valor da função custo, equação (8).
4. Para $g = 1, 2, \dots, G$:

- (a) Para todos os indivíduos de S (colunas) é introduzida a mutação no intuito de se criar novos indivíduos que possuam melhores genes. Para $i = 1, \dots, K$ e $j = 1, \dots, p$:

$$S_{i,j+p} = \text{sign}\{S_{i,j} + \mathcal{N}(0, \sigma_{i,j})\} \quad (13)$$

onde $S_{i,j}$ representa o i -ésimo gene do j -ésimo indivíduo, $\mathcal{N}(0, \sigma_{i,j})$ representa uma variável aleatória Gaussiana com média 0 e desvio padrão $\sigma_{i,j}$. Este desvio padrão é escolhido de forma que a troca de genes (mutação) ocorra com uma dada probabilidade de mutação p_m .

- (b) Para cada $s_{j+p}, j = 1, \dots, p$, é calculado o valor da função custo, equação (8).
- (c) A população S , composta agora pelos $2p$ indivíduos é ordenada de forma decrescente em relação ao valor obtido pela função custo cujo

vetor linha de dimensão $1 \times 2p$ é $\mathbf{f} = [f(s_1), f(s_2), \dots, f(s_{2p})]$. Os melhores p indivíduos, ou seja, que possuem maiores valores para a função custo, são selecionados para compor a base da nova geração.

- (d) incrementa-se g : retorna-se à etapa 4a) até que o número de gerações g atinja um valor G pré-estabelecido.

5. Toma-se o indivíduo s_1 como sendo o vetor de saída do algoritmo.

O algoritmo EP deve ser iniciado com quatro parâmetros: tamanho da população p , percentagem de mutação p_m , número total de gerações G e vetor de genes inicial s_1 .

4.1.2 PROGRAMAÇÃO EVOLUCIONÁRIA COM CLONAGEM (EP-C)

Este algoritmo é baseado na mesma teoria do algoritmo EP, porém aqui são aplicadas as duas estratégias de intensificação descritas na seção 4.1, ou seja, a clonagem e a utilização da matriz de desvio padrão adaptativa.

Na obtenção da matriz de desvio padrão adaptativa, analisou-se a percentagem de troca dos genes na etapa de reprodução da g -ésima geração do algoritmo EP em função do desvio padrão, a qual indicou um comportamento acentuadamente crescente na faixa de $\sigma \in [0, 5; 2]$, figura 3. Otimizações feitas a seguir levaram em consideração este intervalo. Considerou-se esta faixa para σ , uma vez que, valores muito abaixo ou muito acima desta faixa representam uma chance remota de troca de genes ou uma chance elevada de troca, respectivamente. No último caso, é computacionalmente menos complexo gerar aleatoriamente um novo indivíduo candidato. No primeiro caso, não há evolução (geração sem evolução). Assim, adotando-se $\sigma \in [0, 5; 2]$, verifica-se que a faixa de troca de pico estará confinada ao intervalo $\approx [5\%: 60\%]$. Efetivamente, a probabilidade média de troca do gene (p_m) na g -ésima geração estará restrita à faixa de $\approx [2, 5\%: 30\%]$, garantindo um compromisso entre variabilidade (entropia) e evolução.

Em função dos parâmetros E_b/N_0 , NFR e função custo, equação (8), obteve-se uma expressão para o desvio padrão do processo EP-C referente à probabilidade de mudança de genes do indivíduo candidato na g -ésima geração [23]:

$$\sigma_{i,j} = \sqrt{\frac{4}{E_b/N_0}} - \sqrt{\frac{NFR(i)}{32}} + \frac{\max[\varphi] - \varphi(j)}{\max[\varphi]} \quad (14)$$

onde os índices i e j referem-se às linhas e às colunas da matriz de desvio padrão, que refletem os genes (i -ésima linha) e os indivíduos candidatos (j -ésima coluna) da matriz \mathbf{S} , E_b/N_0 expressa a relação sinal-ruído de operação do sistema (referência), com $E_b = A_{ref}^2 \cdot T_b$; potências recebidas abaixo ou acima da potência de referência são computadas através da razão *near-far*: $NFR(i) = (A_i/A_{ref})^2$; $\varphi(j) = f(s_j) - \min[\mathbf{f}]$, onde $\min[\cdot]$ e $\max[\cdot]$ representam o operador valor mínimo e máximo, respectivamente e $\varphi = [\varphi(1), \varphi(2), \dots, \varphi(2p)]$.

O primeiro termo em (14) é devido ao efeito da relação entre energia de bit e densidade espectral de potência do ruído, sendo neste caso um escalar, idêntico para todos os elementos da matriz desvio padrão (σ). O segundo termo é devido à razão *near-far*. Quanto maior a razão *near-far*, menor será o desvio padrão. A razão *near-far* é incluída em cada gene da i -ésima linha, isto é, resultará em valor idêntico para todas as gerações do EP-C dado que a potência recebida de cada usuário é constante dentro do intervalo de otimização. O último termo é devido ao efeito da função custo. Este efeito terá impacto sobre cada indivíduo (j -ésima coluna), implicando em aumento do desvio padrão toda vez que o valor da função custo do j -ésimo indivíduo for menor que o maior valor da função custo obtido.

Apenas o terceiro termo da matriz de desvio padrão é calculado a cada nova geração, pois os valores da função custo são diferentes a cada geração.

Para que $\sigma \in [0, 5; 2]$, as seguintes condições foram impostas:

Se $\sigma_{i,j} < 0, 5$, adota-se $\sigma_{i,j} = 0, 5$.

Se $\sigma_{i,j} > 2$, adota-se $\sigma_{i,j} = 2$.

com $i = 1, \dots, K$ e $j = 1, \dots, 2p$.

O operador elitismo [11, 12], largamente utilizado na literatura, possui características semelhantes às apresentadas pelo operador clonagem. No entanto, enquanto que no operador elitismo simplesmente o melhor ou melhores indivíduos são selecionados para compor a base da próxima geração, no operador clonagem ocorre a multiplicação (replicação) destes melhores indivíduos. Outra diferença refere-se à etapa na qual os dois operadores são aplicados; o operador elitismo é aplicável somente na etapa de reposição em qualquer algoritmo evolucionário [11, 12], enquanto que o operador clonagem é aplicável somente na etapa de reprodução [14, 23, 24].

A estratégia de clonagem substitui a etapa 4c) do algoritmo EP pela seguinte etapa:

4. c) A população \mathbf{S} , composta agora pelos $2p$ indivíduos é ordenada de forma decrescente em relação ao valor obtido pela função custo cujo vetor linha de dimensão $1 \times 2p$ é $\mathbf{f} = [f(s_1), f(s_2), \dots, f(s_{2p})]$. Deve-se escolher apenas uma percentagem de indivíduos para que sejam mantidos na próxima geração. Para $j = 1, \dots, p/I_C$:

$$\mathbf{S} = \Omega \{s_j\} \quad (15)$$

onde I_C é o índice de clonagem e $\Omega \{\cdot\}$ é o operador de clonagem que cria I_C cópias de cada indivíduo. Deve-se empregar um índice de clonagem cuja divisão p/I_C seja um número inteiro. O índice de clonagem é dado por:

$$I_C = \frac{p \cdot i\%}{100} \quad (16)$$

onde $i\%$ é a percentagem de seleção dos indivíduos que possuem maior função custo.

As modificações introduzidas no algoritmo EP-C visam melhorar a taxa de convergência do EP convencional, diminuindo o número de gerações necessárias para se alcançar o desempenho ML.

O algoritmo EP-C deve ser iniciado com seis parâmetros: percentagem de seleção dos indivíduos $i\%$, tamanho da

população p , número total de gerações G , vetor de genes inicial s_1 , NFR e relação sinal-ruído de operação do sistema E_b/N_0 .

4.1.3 ALGORITMO GENÉTICO

A teoria do algoritmo genético (GA - *Genetic Algorithm*) é similar à do algoritmo EP, baseando-se na evolução natural. No entanto, este algoritmo procura descrever as interações biológicas existentes entre as células de um organismo, alterando o genótipo na etapa de transferência do material genético. Neste caso, por se tratar de reprodução de células, os operadores genéticos mutação e *crossover* são utilizados como estratégias de diversificação [8, 26].

O GA utiliza o operador *crossover* como seu principal operador genético no intuito de trazer variabilidade de busca sem perder as características de adaptação adquiridas. A mutação não é considerada essencial, pois em uma população real a taxa de mutação é baixa, fazendo com que a mutação seja um mecanismo secundário no processo de adaptação dos algoritmos genéticos.

O GA é idêntico ao algoritmo EP, porém incluindo o critério de *crossover*, antes da etapa 4a, sendo descrito por:

4. a') É introduzido o efeito da recombinação genética, que cria dois novos indivíduos combinando-se sub-partes de dois indivíduos genitores (com probabilidade de *crossover* = p_c). A matriz de genitores é dada por (12). Assim, a matriz de novos indivíduos é criada trocando-se genes dos indivíduos genitores a partir de pontos de *crossover*. Neste trabalho, os pontos de *crossover* são gerados aleatoriamente.

O GA é iniciado com cinco parâmetros: percentagem de *crossover* p_c , percentagem de mutação p_m , tamanho da população p , número total de gerações G , vetor de genes inicial s_1 .

4.1.4 ALGORITMO DE CLONAGEM SELETIVA (CLONALG)

Este algoritmo foi proposto em [14] sendo baseado na resposta imunológica de um sistema na presença de um antígeno. Quando um antígeno entra em contato com um organismo este produz diversos tipos de anticorpos no intuito de combatê-lo. Apenas os anticorpos que tiverem maior afinidade com o antígeno são reproduzidos. Quando este antígeno é novamente colocado em contato com o organismo, a reação a ele se torna mais rápida, pois este é reconhecido e apenas os melhores anticorpos são utilizados. Quanto mais vezes o organismo for colocado em contato com o antígeno, mais rápida se torna a resposta a ele.

O algoritmo Clonalg utiliza o operador mutação como critério de diversificação e o operador clonagem como critério de intensificação. Descreve-se a seguir o algoritmo Clonalg aplicado à detecção multiusuário:

1. Atribui-se ao vetor de genes do anticorpo inicial a saída do detector convencional, equação (10).

2. Os outros vetores de genes dos anticorpos, que formam a população da 1ª geração, são determinados a partir de um processo aleatório, definido conforme a equação (11).

3. Para $g = 1, 2, \dots, G$:

- (a) Para cada $s_j, j = 1, \dots, p$, calcula-se o valor da função custo, equação (8). Reordena-se a população segundo valores decrescentes da função custo e apenas os n melhores anticorpos são selecionados.

- (b) Utiliza-se o operador clonagem, criando $N_C(j) = \text{round}\left(\frac{v \cdot p}{j}\right)$ cópias de cada anticorpo (incluso a semente), onde $N_C = \sum_{j=1}^n N_C(j)$ é o número total

de anticorpos clonados a cada geração, v é um fator multiplicativo que estabelece um compromisso entre complexidade e convergência e $\text{round}(\cdot)$ é o operador que retorna o inteiro mais próximo. Com isso, os anticorpos são clonados em quantidades diferentes, dependendo da classificação em relação ao valor da função custo [14].

- (c) Para cada $s_j, j = 1, \dots, N_C$, é utilizado o operador mutação, visando diversificar a busca. O operador mutação é introduzido segundo uma função baseada na afinidade do anticorpo com o antígeno, dada por:

$$\zeta_j = 100 \cdot \exp(-\rho f_j) \quad (17)$$

onde ζ_j é a percentagem de mutação, $\rho > 0$ controla o decaimento da função e $f_j = \frac{f(s_j)}{f_{max}}$ é a afinidade do anticorpo ao antígeno, com $f_{max} = \max_{j=1, \dots, N_C} [f(s_j)]$.

- (d) Para cada $s_j, j = 1, \dots, N_C$, é calculado o valor da função custo, equação (8).

- (e) A população agora composta pelos N_C indivíduos é ordenada de forma decrescente em relação ao valor obtido pela função custo, $f = [f(s_1), f(s_2), \dots, f(s_{N_C})]$. Os melhores n anticorpos, ou seja, os que possuem maiores valores para a função custo são selecionados para compor a base da nova geração.

- (f) Para $j = n + 1, \dots, p$ são gerados anticorpos determinados pelo processo aleatório definido pela equação (11). A população completa dos anticorpos é composta pelos n anticorpos selecionados no passo anterior (3e) e pelos $p - n$ anticorpos gerados nesta etapa. A matriz de anticorpos possui K linhas e p colunas (anticorpos), sendo expressa também pela equação (12).

- (g) Incrementa-se g ; retorna-se à etapa 3 até que o número de gerações g atinja o valor G pre-estabelecido.

4. Toma-se o anticorpo s_1 como sendo o vetor de saída do algoritmo.

O algoritmo Clonalg é iniciado com seis parâmetros: constante de decaimento ρ , fator responsável pelo compromisso complexidade-convergência v , número de anticorpos clonados por geração n , tamanho da população p , número total de gerações G , além do vetor de genes inicial s_1 .

4.2 ALGORITMOS DE BUSCA LOCAL

O problema da detecção multiusuário também pode ser resolvido utilizando algoritmos heurísticos de busca local [15]. Estes algoritmos de busca determinística procuram por uma solução ao redor de uma vizinhança [3, 15]. Esta vizinhança é baseada na distância de Hamming.

Entre os algoritmos de Busca Local existem duas estratégias de deslocamento no interior do espaço de busca: a estratégia de deslocamento pelo maior ganho e a estratégia de deslocamento pelo primeiro ganho. A primeira utiliza a melhor solução de uma vizinhança como entrada da próxima iteração. A outra estratégia utiliza a primeira solução que apresentar um maior valor que a solução atual. Neste trabalho, utilizou-se a estratégia de deslocamento pelo maior ganho.

Na descrição dos algoritmos de busca local, m indica a iteração atual, M , o número total de iterações e o operador (\cdot) indica o maior valor ou o vetor associado ao maior valor em todos os (\cdot) .

4.2.1 BUSCA LOCAL 1-OPT (1-OPT LS)

O algoritmo 1-opt LS (*1-optimum Local Search*) é um algoritmo de busca local que procura por uma solução ao redor da estimativa inicial cujos vetores de bits candidatos são compostos por todas as possíveis soluções com distância de Hamming igual a 1.

Visando a redução de retornos a máximos locais, propôs-se aqui um procedimento adicional à estratégia de deslocamento do algoritmo 1-opt LS padrão, identificado na etapa 3d do algoritmo a seguir. No contexto da detecção multiusuário, este procedimento adicional consiste em não retornar à última solução visitada e tida como a melhor solução, tentando fazer com que o algoritmo não retorne ao último máximo local. Apesar desta estratégia melhorar a taxa de escape de máximos locais, esse procedimento adicional não garante que o algoritmo atinja o máximo global. Descreve-se a seguir o algoritmo 1-opt LS aplicado à detecção multiusuário:

1. Utiliza-se a saída do detector convencional como solução inicial, conforme a equação (10).

2. Calcula-se a energia inicial do sistema através da função custo, equação (8). Inicializa-se:

$$s_{best} \leftarrow s_1 \quad (18a)$$

$$f_{best} \leftarrow f(s_1) \quad (18b)$$

$$\tilde{s}_1 \leftarrow s_1 \quad (18c)$$

3. Para $m = 1, 2, \dots, M_t$:

(a) Encontram-se as K possíveis soluções na vizinhança $V(\tilde{s}_m)$ cuja distância de Hamming é igual a 1, sendo \tilde{s}_m a solução corrente na iteração m .

$$V(\tilde{s}_m) = \left\{ s_j \in \{-1, 1\}^K \text{ tal que } \|s_j - \tilde{s}_m\| = 1 \right\} \quad (19)$$

onde $j = 1, \dots, K$.

(b) Utilizando a equação (8), calcula-se a energia do sistema para cada uma das soluções possíveis.

(c) Encontra-se o ganho Δe para as energias calculadas:

$$\Delta e(j) = f(s_j) - f(s_{best}) \quad (20)$$

(d) Determina-se qual o maior ganho de energia e qual é a solução correspondente (corrente).

Se $\underline{\Delta e} > 0$:

$$\tilde{s}_{m+1} \leftarrow \underline{s} \quad (21a)$$

$$s_{best} \leftarrow \underline{s} \quad (21b)$$

Se $\underline{\Delta e} \leq 0$:

$$\tilde{s}_{m+1} \leftarrow \underline{s}^* \quad (22)$$

onde o vetor solução corrente $\underline{s}^* \in V(\tilde{s}_m)$ e $\underline{s}^* \neq \tilde{s}_{m-1}$, com $m > 1$; na primeira iteração ($m = 1$), $\underline{s}^* = \underline{s}$.

(e) Incrementa-se m : retorna-se à etapa 3 até que o número de iterações m atinja o valor M_t pré-estabelecido.

4. Toma-se o vetor s_{best} como o vetor de saída do algoritmo.

O algoritmo 1-opt LS deve ser iniciado com os parâmetros número total de iterações M_t e o vetor de genes inicial s_1 .

4.2.2 BUSCA LOCAL K-OPT (K-OPT LS)

O algoritmo k -opt LS (*k-optimum Local Search*) também é um algoritmo de busca local baseado na mesma estratégia que o algoritmo 1-opt LS. Porém, este algoritmo procura soluções em uma vizinhança maior, cuja distância de Hamming é menor ou igual a k . Portanto, para o algoritmo k -opt LS é necessário apenas substituir a etapa 3a) do algoritmo 1-opt LS por:

3) Para $m = 1, 2, \dots, M_t$:

(a) Encontram-se todas as $Q = \sum_{i=1}^k \frac{K!}{(K-i)!i!}$ soluções candidatas possíveis na vizinhança $V(\tilde{s}_m)$ cuja distância de Hamming é menor ou igual a k .

$$V(\tilde{s}_m) = \left\{ s_j \in \{-1, 1\}^K \text{ tal que } \|s_j - \tilde{s}_m\| \leq k \right\} \quad (23)$$

onde $j = 1, \dots, Q$.

O algoritmo k -opt LS deve ser iniciado com os seguintes parâmetros: $k \geq 2$, número total de iterações M_t e vetor de genes inicial s_1 .

4.2.3 RECOZIMENTO SIMULADO (SA)

O conceito do algoritmo recozimento simulado (SA - *Simulated Annealing*) está associado a um princípio da termodinâmica, onde um sólido aquecido a uma temperatura muito alta e depois resfriado gradualmente, tenderá a se solidificar de modo a formar uma estrutura de menor energia possível [3, 16, 17].

Para escapar de soluções locais, o SA utiliza uma função probabilidade de aceitação $z(m)$ na m -ésima iteração, proporcional à temperatura $T(m)$, podendo então aceitar uma solução particular que possui um menor valor de energia em relação à solução corrente. Isto possibilita que o algoritmo escape de uma região de máximo local e procure pelo máximo global em outras regiões. Ao longo das iterações, a temperatura sofre decréscimos baseados geralmente na função de distribuição de Boltzman.

O SA utilizado é baseado na busca local do 1-opt LS, procurando uma melhor solução na vizinhança cuja distância de Hamming é igual a 1, sendo chamado de SA-LS [18]. Descreve-se a seguir o algoritmo SA-LS:

1. Utiliza-se a saída do detector convencional como a solução inicial, conforme a equação (10).
2. Calcula-se a energia inicial do sistema através da função custo, equação (8). Inicializa-se:

$$s_{best} \leftarrow s_1 \quad (24a)$$

$$f_{best} \leftarrow f(s_1) \quad (24b)$$

$$\tilde{s}_m \leftarrow s_1 \quad (24c)$$

3. Para $m = 1, 2, \dots, M_t$:
 - (a) Encontram-se as K possíveis soluções na vizinhança $V(\tilde{s}_m)$ cuja distância de Hamming é igual a 1, conforme a equação (19).
 - (b) Calcula-se a energia do sistema para cada uma das soluções possíveis através da função custo, equação (8).
 - (c) Se $f(\underline{s}) > f_{best}$:

$$s_{best} \leftarrow \underline{s} \quad (25a)$$

$$f_{best} \leftarrow f(\underline{s}) \quad (25b)$$

- (d) Calcula-se a variação de energia de modo similar à equação (20):

$$\Delta e(j) = f(s_j) - f(\tilde{s}_m) \quad (26)$$

Se $\Delta e > 0$:

$$\tilde{s}_{m+1} \leftarrow \underline{s} \quad (27a)$$

$$f(\tilde{s}_{m+1}) \leftarrow f(\underline{s}) \quad (27b)$$

Se $\Delta e \leq 0$, gera-se um número aleatório com distribuição uniforme $\nu \in [0, 1]$;

Se $\nu < z(m)$,

executa-se (27);

Caso contrário, mantém-se a solução corrente:

$$\tilde{s}_{m+1} \leftarrow \tilde{s}_m \quad (28a)$$

$$f(\tilde{s}_{m+1}) \leftarrow f(\tilde{s}_m) \quad (28b)$$

onde $z(m)$ é o critério de aceitação.

4. incrementa-se m ; retorna-se à etapa 3 até que o número de iterações m atinja o valor M_t pré-estabelecido.
5. Toma-se o vetor s_{best} como o vetor de saída do algoritmo.

O critério da probabilidade de aceitação $z(m)$ é inspirado na termodinâmica, onde a distribuição de Boltzman é usualmente utilizada:

$$z(m) = \exp\left[-\frac{\Delta e}{T(m)}\right] \quad (29)$$

onde $T(m)$ é a temperatura na iteração m . Geralmente, a temperatura inicial possui um valor alto, $T(0) > 0$, sendo gradativamente reduzida através do processo de resfriamento. Existem vários métodos para o processo de resfriamento. Um dos mais eficientes, utilizado em vários problemas de otimização, consiste em:

1. Inicializar a temperatura $T(0)$.
2. Manter a temperatura constante por \mathcal{L} iterações consecutivas.
3. A cada série de \mathcal{L} iterações, diminuir a temperatura através de um fator fixo $\gamma \in [0, 1]$. A cada iteração, a temperatura é expressa por:

$$T(m) = \gamma^x T(0) \quad (30)$$

onde $x = \lceil \frac{m}{\mathcal{L}} \rceil$ e o operador $\lceil \lambda \rceil$ indica o maior inteiro contido em λ . Portanto, o algoritmo SA deve ser iniciado com cinco parâmetros: temperatura inicial $T(0)$, taxa de resfriamento γ , tamanho da série (platô) \mathcal{L} , número total de iterações M_t , além do vetor de genes inicial s_1 .

4.2.4 BUSCA TABU DE PERÍODO CURTO (STTS)

O algoritmo busca tabu de período curto (STTS - *Short Term Tabu Search*) é baseado no modo determinístico do funcionamento de uma memória. A memória é implementada através da gravação de características de deslocamento de soluções previamente visitadas [3, 19, 27, 28]. Esta é descrita pela lista Tabu, a qual é formada pelo passado recente de busca, sendo chamada de efeito de memória de período curto (*Short Term*). Estas características de deslocamento são proibidas pela lista Tabu por um certo número de iterações, determinado pela variável P . Isto ajuda a evitar retornos para máximos locais, promovendo uma diversificação na busca de soluções. Descreve-se a seguir o algoritmo STTS.

1. Utiliza-se a saída do detector convencional como solução inicial, equação (10). A lista Tabu, T_{LIST} , está vazia.
2. Calcula-se a energia inicial do sistema através da função custo, equação (8). Inicializa-se:

$$s_{best} \leftarrow s_1 \quad (31a)$$

$$f_{best} \leftarrow f(s_1) \quad (31b)$$

$$\tilde{s}_m \leftarrow s_1 \quad (31c)$$

$$f_m \leftarrow -\infty \quad (31d)$$

3. Para $m = 1, 2, \dots, M_t$;

(a) Para $j = 1, \dots, K$:

- Cria-se um vetor característica (w_j); este vetor de dimensão $K \times 1$ contém $K - 1$ valores iguais a 1, sendo o j -ésimo valor igual a -1 .
- Visando a obtenção de todos vetores com distância de Hamming igual a 1 em relação à solução corrente, como definido em (19), realiza-se a troca de um bit no vetor \tilde{s}_m , multiplicando cada elemento pelo seu respectivo valor no vetor característica, ou seja:

$$s_j \leftarrow \tilde{s}_m \cdot w_j \quad (32)$$

Retorna-se ao início desta etapa até que $j = K$.

(b) Calcula-se a energia atual para todos os s_j vetores candidatos através da equação (8).

(c) Se $\underline{w} \in T_{LIST}$:

$$rpt \leftarrow 1 \quad (33)$$

Caso contrário:

$$\tilde{s}_{m+1} \leftarrow \underline{s} \quad (34a)$$

$$f_{m+1} \leftarrow f(\underline{s}) \quad (34b)$$

A variável rpt auxilia o programador na determinação do período de proibição P da lista Tabu. O valor de P deve ser definido conforme a frequência com que a variável rpt é ativada. Visando garantir diversificação ao processo de busca, o programador deve aumentar o período de proibição P toda vez que a variável rpt for ativada frequentemente. Por outro lado, para permitir intensificação ao processo de busca, o programador deve garantir uma redução no período de proibição P quando rpt for ativada raramente.

(d) Se $f(\underline{s}) > f_{best}$ (critério de aspiração):

$$s_{best} \leftarrow \underline{s} \quad (35a)$$

$$f_{best} \leftarrow f(\underline{s}) \quad (35b)$$

Assim, caso $f(\underline{s}) > f_{best}$, sendo portanto, a melhor solução visitada até o momento, deve-se utilizar o princípio da intensificação. Este princípio visa identificar regiões atrativas, examinando todos os possíveis deslocamentos da solução; ou seja, na próxima iteração é desconsiderada a etapa 3c), sendo esta nova solução aceita independentemente da lista Tabu e do critério de aspiração.

(e) Transfere-se a característica de deslocamento da melhor solução para a lista Tabu, ou seja:

$$T_{LIST} \leftarrow \underline{w} \quad (36)$$

Se a lista Tabu tiver completado P períodos, ou seja, se T_{LIST} conter P vetores características, esta é esvaziada:

$$T_{LIST} \leftarrow [] \quad (37a)$$

$$rpt \leftarrow 0 \quad (37b)$$

onde $[]$ significa conjunto vazio.

(f) incrementa-se m ; retorna-se ao início da etapa 3 até que o número de iterações m atinja o valor M_t pré-estabelecido.

4. Toma-se o vetor s_{best} como o vetor de saída do algoritmo.

O critério de aspiração utilizado neste trabalho aceita uma nova solução, independentemente da lista Tabu, quando esta possuir um valor de energia maior que todas as soluções já visitadas anteriormente.

O algoritmo STTS deve ser iniciado com os seguintes parâmetros: período de proibição P , número total de iterações M_t e vetor de genes inicial s_1 .

4.2.5 BUSCA TABU REATIVA (RTS)

O algoritmo busca tabu reativa (RTS - *Reactive Tabu Search*) combina o efeito de período curto com outro efeito de memória para evitar retornos aos máximos locais e garantir uma busca eficiente. Este efeito é conhecido como memória de período longo (*Long Term*), o qual alterna entre as fases de intensificação e diversificação da busca [3, 20].

A lista Tabu do algoritmo STTS é implementada utilizando o efeito de período curto (*short term*). Mas isto não garante escapar de retornos aos máximos locais. Adicionalmente, a escolha de um período de proibição (P) fixo, adequado para cada problema, torna-se uma tarefa difícil, pois um período pequeno é insuficiente para evitar retornos a máximos locais e um período grande demais reduz a quantidade de deslocamentos possíveis, acarretando uma busca ineficiente.

A memória de período longo do RTS é constituída pelo efeito de memória de período curto do algoritmo STTS, adaptando-se o período de proibição durante a busca, admitindo-se que este período, $P(m)$, assuma diferentes valores a cada iteração.

O algoritmo RTS é idêntico ao STTS, exceto que a etapa 3e) deve ser substituída por:

3. e) Transfere-se a característica da melhor solução para a lista Tabu, ou seja:

$$T_{LIST} \leftarrow \underline{w} \quad (38)$$

Caso $rpt = 1$:

$$P(m+1) = P(m) + stp \quad (39)$$

Caso $|f(\mathbf{s}) - f_{best}| < \psi f_{best}$:

$$P(m+1) = \max[(P(m) - stp) ; P(0)] \quad (40)$$

Se a lista Tabu tiver completado $P(m)$ períodos, ou seja, se T_{LIST} conter $P(m)$ vetores características, esta é esvaziada:

$$T_{LIST} \leftarrow [] \quad (41a)$$

$$rpt \leftarrow 0 \quad (41b)$$

O período de proibição é inicializado com um valor pequeno, $P(0)$, sendo alterado conforme a ocorrência de repetições. Tanto no STTS quanto no RTS, a variável rpt auxilia no controle do período de proibição da lista Tabu. No algoritmo RTS, o período de proibição é determinado diretamente da variável repetição rpt . Desta forma, toda vez que uma repetição é encontrada, encoraja-se a diversificação através do incremento do período $P(m)$ em passos de stp . Para que essa diversificação não assuma valores muito altos após algumas iterações, reduz-se o período $P(m)$ em passos de stp quando $|f(\mathbf{s}) - f_{best}| < \psi f_{best}$, onde $0 < \psi < 1$ é uma constante que controla a redução do período $P(m)$. Em [3] foi obtido experimentalmente um valor robusto, $\psi = 0,3$, considerando o problema da detecção multiusuário em canal AWGN síncrono.

O algoritmo RTS deve ser iniciado com os seguintes parâmetros: período de proibição inicial $P(0)$, constante que controla a redução do período de proibição ψ e o número total de iterações M_t , além do vetor de genes inicial \mathbf{s}_1 .

5. RESULTADOS NUMÉRICOS

Nesta seção são apresentados resultados de desempenho para o sistema DS/CDMA, utilizando os algoritmos heurísticos evolucionários e de busca local no processo de detecção multiusuário. Os seguintes parâmetros foram considerados: seqüências aleatórias de comprimento $N = 32$, número de usuários $K = 12, 24, 32, 48$ e 64 , resultando em um carregamento $L = K/N = 37,5\%, 75\%, 100\%, 150\%$ e 200% respectivamente; região de relação sinal-ruído média ($E_b/N_0 = 15\text{dB}$); cenários com controle perfeito de potência e com disparidades na faixa de $NFR = 15$ a 30dB .

Para os parâmetros do canal, adotou-se o modelo de *Jakes* modificado [29], com frequência da portadora $f_c = 2\text{GHz}$, número de osciladores $N_d = 36$ e velocidades de deslocamento dos móveis uniformemente distribuídas entre 0 e 120km/h .

Em todas as simulações Monte Carlo adotou-se um número mínimo de erros/ponto³ = 100 . Para efeito de comparação foram incluídos os desempenhos dos detectores convencional (CD), de máxima verossimilhança (ML) e o limite quando há apenas um único usuário ativo no sistema (*SuB-Single user Bound*) com modulação BPSK e canal Rayleigh Plano [31].

No intuito de se realizar uma comparação justa entre os algoritmos, buscou-se valores otimizados para os parâmetros

obtidos em duas etapas: a) simulações preliminares foram conduzidas adotando-se valores típicos encontrados na literatura; b) simulações adicionais foram feitas visando a otimização dos parâmetros, de modo não exaustivo, porém com resultados de desempenho superiores aos obtidos na etapa a). Observe-se que para todos os 9 algoritmos analisados neste trabalho nem sempre é possível encontrar na literatura valores otimizados de parâmetros aplicados à detecção multiusuário.

Para os algoritmos evolucionários na condição de baixo carregamento do sistema, $L = 37,5\%$, adotou-se $p = 50$; $p_m = 15\%$ para o algoritmo EP; $i_{\%} = 10\%$ para o EP-C; $p_c = 85\%$ e $p_m = 5\%$ para o GA; $n = 10$, $v = 0,25$ e $\rho = 5$ para o algoritmo Clonalg. Em alto carregamento, $L = 75\%$, apenas os valores de p , n e v foram modificados, adotando-se $p = 160$, $n = 20$ e $v = 0,2$.

Para os algoritmos de busca local adotou-se $k = 3$ para o algoritmo k -opt LS; para temperatura inicial foi adotado o valor de 30% do número de usuários ($T(0) = 0,3K$), $\gamma = 0,9$ e $\mathcal{L} = 2$ para o algoritmo SA-LS; para o algoritmo STTS adotou-se $P = 3$ e para o algoritmo RTS adotou-se $P(0) = 1$, $\psi = 0,3$ e $stp = 2$.

As figuras 5 à 12 apresentam o desempenho para os 9 algoritmos heurísticos em função da velocidade de convergência para diferentes condições de sistema. Nas curvas, cada ponto indica o desempenho na g -ésima geração ou na m -ésima iteração.

5.1 DESEMPENHO DOS ALGORITMOS EVOLUCIONÁRIOS

A figura 5 indica que os algoritmos evolucionários convergem para o desempenho ML após um certo número de gerações. Percebe-se o efeito dos critérios de intensificação e de diversificação em cada algoritmo, sendo que o algoritmo EP-C alcançou o desempenho ML mais rapidamente. O algoritmo EP-C apresenta ganho de convergência em relação ao EP tradicional, pois utiliza duas estratégias adicionais, a clonagem e a matriz de desvio padrão adaptativa. Embora a estratégia de clonagem do algoritmo Clonalg apresente boas características, devido à ponderação no processo de clonagem, a taxa de convergência mostrou-se ser lenta, causada pela sua ineficiente estratégia de mutação. O algoritmo GA mostrou ter uma boa velocidade de convergência, pois combina duas estratégias de diversificação, o *crossover* e a mutação.

A figura 6 mostra o desempenho alcançado pelos algoritmos em um cenário com controle perfeito de potência e com alto carregamento, $L = 75\%$. Note-se que os algoritmos EP e Clonalg não atingiram o desempenho ML após $G = 60$ gerações, mostrando que há perda de convergência à medida que o carregamento cresce. Esta perda de convergência pode ser explicada pela ineficiente estratégia de mutação do algoritmo Clonalg e pela falta de uma estratégia de intensificação do algoritmo EP. Novamente, o algoritmo EP-C alcançou o desempenho ML com o menor número de gerações, mostrando que suas estratégias de intensificação, clonagem e matriz de desvio padrão adaptativa, são eficientes para o problema da detecção multiusuário síncrono em canal Rayleigh

³O que garante um erro na BER menor que 10% com um intervalo de confiança de 95% [30].

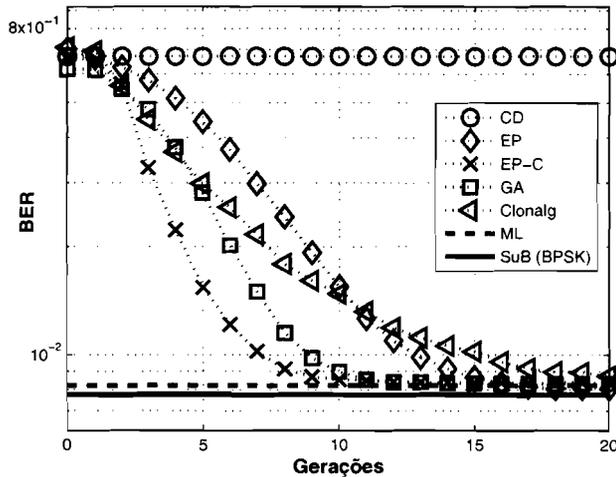


Figura 5. Desempenho em função da velocidade de convergência dos algoritmos evolucionários na condição de baixo carregamento e controle perfeito de potência.

plano, tanto em baixo quanto em alto carregamento. Embora o algoritmo GA tenha boas estratégias de diversificação, este não possui estratégia de intensificação, acarretando perda de convergência com o aumento do carregamento.

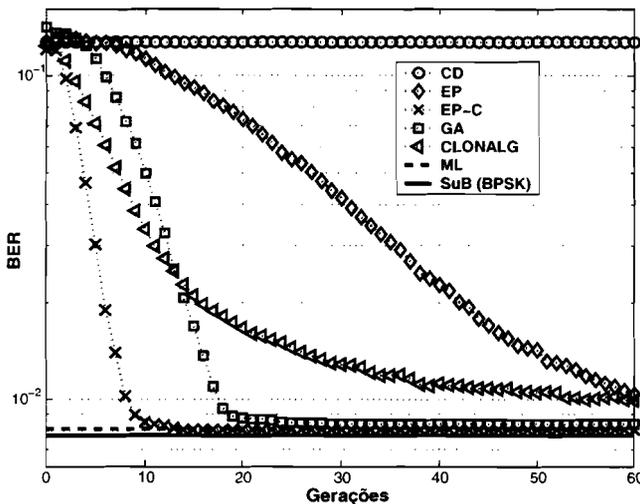


Figura 6. Desempenho em relação à velocidade de convergência para os algoritmos evolucionários na condição de alto carregamento e com controle perfeito de potência.

Finalmente a figura 7 mostra o desempenho alcançado pelos algoritmos em um cenário com usuários divididos em níveis distintos de potência recebida e alto carregamento: $L = 75\%$, 8 usuários com $NFR = 0\text{dB}$, 8 usuários com $NFR = 15\text{dB}$ e 8 usuários com $NFR = 30\text{dB}$, sendo obtido o desempenho médio em relação aos usuários com menor nível de potência recebida. Percebe-se que o algoritmo EP não converge após $G = 60$, evidenciando ainda mais sua ineficiente estratégia de diversificação e ausência de uma estratégia de intensificação.

O algoritmo Clonalg convergiu após $g \approx 52$ gerações, o

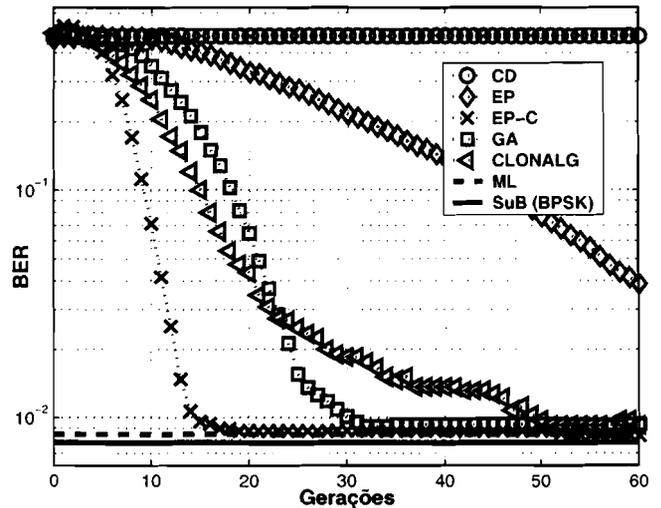


Figura 7. Desempenho dos algoritmos evolucionários em função da velocidade de convergência para alto carregamento e fortes disparidades de potência.

que mostra que apesar da ineficiente estratégia de mutação, equação (17), esta consegue identificar os anticorpos por afinidade devido à existência da razão NFR , fazendo com que a porcentagem de mutação dos anticorpos com menor afinidade fosse maior, melhorando um pouco sua velocidade de convergência em relação ao caso de $NFR = 0$.

Novamente, o algoritmo EP-C mostrou possuir a melhor convergência dentre os evolucionários, pois atingiu o desempenho ML com o menor número de gerações. Estes resultados mostram a robustez do algoritmo EP-C simultaneamente aos efeitos do aumento de carregamento e *near-far*, indicando que as estratégias conjuntas de intensificação, clonagem e matriz de desvio padrão adaptativa, são altamente eficientes para o problema de otimização considerado.

Percebe-se que a convergência do GA é afetada tanto pelo aumento do carregamento quanto pela disparidade de potência entre os usuários (NFR).

5.2 DESEMPENHO DOS ALGORITMOS DE BUSCA LOCAL

Através da figura 8, nota-se que os algoritmos 1-opt LS, SA-LS, STTS e RTS convergem praticamente para o mesmo desempenho utilizando o mesmo número de iterações, mostrando que o universo de busca baseada na equação (8) não apresenta regiões de máximo local críticas, pois até as estratégias de escape menos elaboradas conseguem atingir, na mesma iteração, desempenho similar a outras estratégias mais sofisticadas.

O algoritmo 3-opt LS convergiu com um menor número de iterações, pois este procura uma solução em um espaço de busca maior por iteração. No entanto sua complexidade computacional por iteração é muito maior que a do algoritmo 1-opt LS, como será verificado na seção 6.

A figura 9 mostra o desempenho alcançado pelos algoritmos em um cenário com controle perfeito de potência e com alto carregamento, $L = 75\%$. Nota-se que os algoritmos

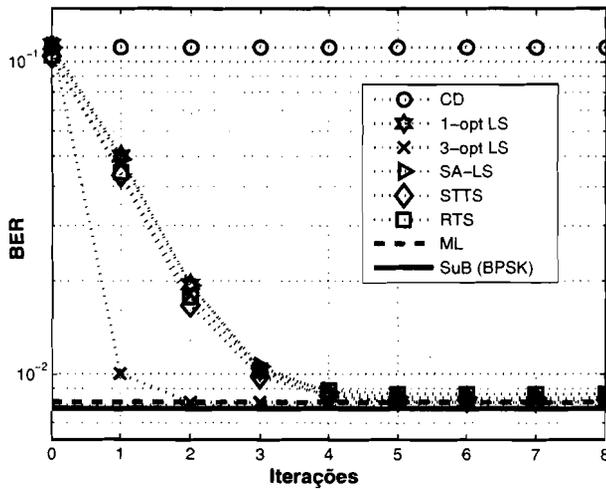


Figura 8. Desempenho dos algoritmos de busca local para baixo carregamento e com controle perfeito de potência.

convergem para o desempenho ML utilizando um mesmo número de iterações, o que corrobora a idéia do emprego de algoritmos com estratégias mais simples para escapar de soluções locais em problemas que não apresentem máximos locais críticos. Os algoritmos de busca local mostraram-se robustos ao aumento do carregamento quando há controle perfeito de potência, pois todos os algoritmos convergiram para a solução ML, sem perda na velocidade de convergência.

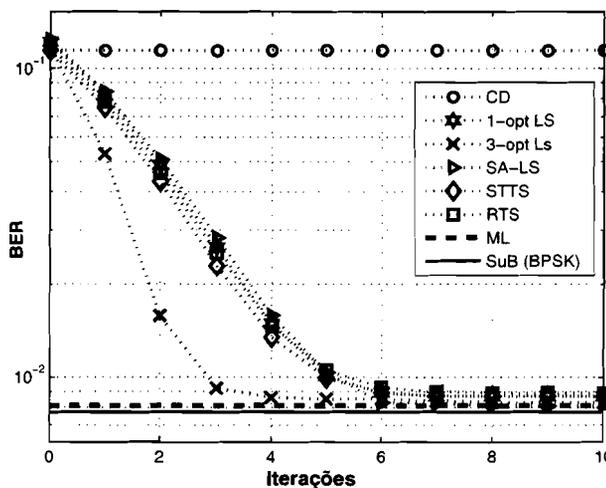


Figura 9. Desempenho dos algoritmos de busca local para alto carregamento e com controle perfeito de potência.

Finalmente, a figura 10 mostra o desempenho alcançado pelos algoritmos em um cenário com usuários divididos em níveis distintos de potência e alto carregamento: $L = 75\%$, 8 usuários com $NFR = 0\text{dB}$, 8 usuários com $NFR = 15\text{dB}$ e 8 usuários com $NFR = 30\text{dB}$, sendo avaliado o desempenho médio dos usuários com menor nível de potência. Note-se que os algoritmos necessitaram de um maior número de iterações para convergirem nesta condição do que na

condição da figura 9, evidenciando uma redução na velocidade de convergência quando há disparidade de potência recebidas entre os usuários ($NFR \neq 0$). Isto indica que

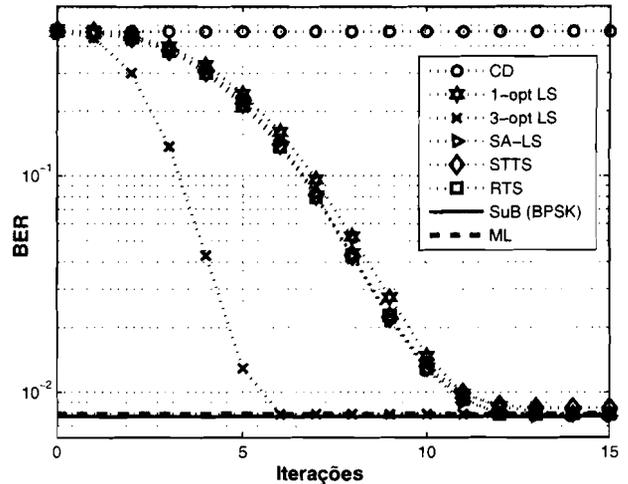


Figura 10. Desempenho dos algoritmos de busca local para alto carregamento e fortes disparidades de potência.

a velocidade de convergência dos algoritmos heurísticos de busca local, aplicados ao problema da detecção multiusuário para sistemas DS/CDMA em canais com desvanecimento Rayleigh Plano, é sensível ao efeito *near-far*, implicando em processamento adicional na determinação e cômputo das iterações necessárias para se alcançar o desempenho ML nas diferentes condições de NFR.

Note-se que para as condições de carregamento de sistema simulado ($L = 37, 5\%$ e 75%), os desempenhos dos detectores ML e dos algoritmos heurísticos evolucionários e de busca local resultaram muito próximos ao SuB. No entanto, para sistemas sobrecarregados ocorre perda progressiva de desempenho; a figura 11 ilustra a redução de desempenho do algoritmo 1-opt LS com o aumento do carregamento, $L = 100\%$, 150% e 200% . Utilizou-se apenas o algoritmo 1-opt LS na obtenção da degradação de desempenho considerando sistemas sobrecarregados, pois todos os algoritmos heurísticos analisados procuram maximizar a mesma função custo, equação (8), e um aumento da MAI acarreta uma perda idêntica de desempenho para todos os algoritmos. Caso fosse utilizado outro algoritmo, o limite (patamar) de desempenho seria o mesmo obtido com o algoritmo 1-opt LS, mudando apenas a velocidade de convergência.

Observe-se que uma degradação considerável no desempenho do detector ML, nas condições de canal simuladas, ocorre apenas para carregamentos maiores que 150% . Adicionalmente, o algoritmo 1-opt LS atinge o mesmo desempenho do detector ML em todas as condições de carregamento, mostrando que a utilização de algoritmos heurísticos traz um ganho não só em termos de BER, como também de capacidade do sistema em termos do número máximo de usuários ativos, quando comparados com outras técnicas de detecção multiusuário sub-ótimas [2, 25].

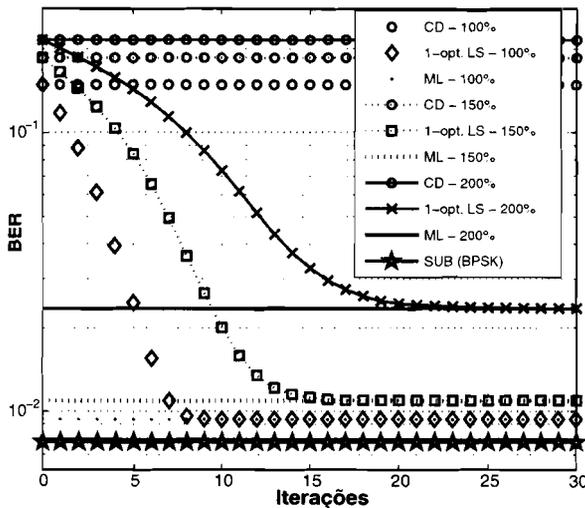


Figura 11. Desempenho do algoritmo 1-opt LS para sistema sobrecarregado.

5.3 DESEMPENHO COM ERROS NAS ESTIMATIVAS DE CANAL

Novamente, utilizou-se apenas o algoritmo 1-opt LS para a obtenção da degradação de desempenho considerando erros nas estimativas de canal, pois todos os demais algoritmos heurísticos analisados procuram maximizar a mesma função custo (equação (8)) e erros nas estimativas de canal acarretam perdas idênticas de desempenho para todos os algoritmos. Esta degradação está sintetizada na figura 12. O erro foi introduzido na estimativa do módulo (β) e/ou da fase (ϕ) dos coeficientes de canal separadamente ou em conjunto, para cada usuário e atualizados a cada símbolo transmitido. Estes erros foram modelados através de distribuições uniformes:

$$\hat{\beta}_k = \mathcal{U}[1 \pm \Delta_\beta] \times \beta_k \quad ; \quad \hat{\phi}_k = \mathcal{U}[1 \pm \Delta_\phi] \times \phi_k$$

com Δ_β e $\Delta_\phi \in [0; 0, 10; 0, 20; 0, 25; 0, 50]$.

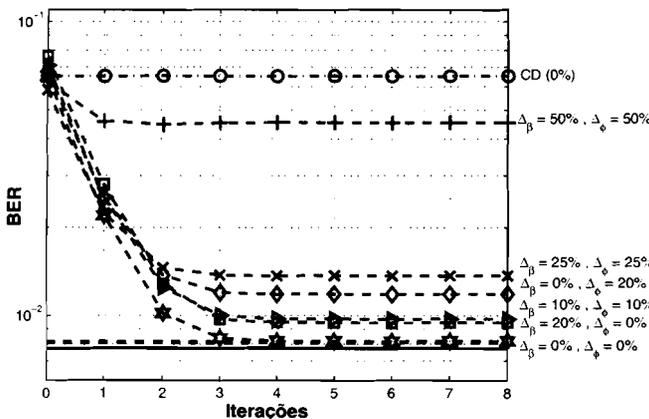


Figura 12. Degradação do desempenho para o algoritmo 1-opt LS quando há erros nas estimativas do canal, controle perfeito de potência e $K = 12$ usuários.

Mesmo com erros nas estimativas do módulo e fase dos coeficientes de canal, da ordem de $\pm 50\%$, uniformemente dis-

tribuídos no intervalo $[1 \pm 0, 5] \times \beta_k$ e $[1 \pm 0, 5] \times \phi_k$, os algoritmos heurísticos atingem melhores desempenhos que o obtido com o detector convencional com estimativas perfeitas para os coeficientes de canal. Observe-se ainda que a tolerância a erros na estimativa da fase do canal é menor que a tolerância a erros na estimativa do módulo.

6. COMPLEXIDADE COMPUTACIONAL

No intuito de expressar a complexidade dos algoritmos analisados, encontrou-se o número de operações envolvidas em cada cálculo da função custo. No cálculo da função custo, conforme a equação (8), as operações $f_1 = 2y^T c^H A$ e $f_2 = c A R A c^H$ podem ser obtidas antes do laço de busca da solução global em cada algoritmo. Para cada teste de solução candidata, computam-se $f_1 \cdot b$ e $b^T \cdot f_2 \cdot b$, que em termos de operações é equivalente a $K^2 + 2K$ multiplicações e 1 transposição de ordem K .

Para o detector ML, o número de operações cresce exponencialmente com o número de usuários, i.e., $O(2^K K^2)$, onde o operador $O(\cdot)$ indica proporcionalidade ao argumento. São necessárias 2^K gerações de bits de ordem K e 2^K cálculos da função custo para a detecção simultânea de 1 bit dos K usuários.

6.1 COMPLEXIDADE PARA OS ALGORITMOS EVOLUCIONÁRIOS

O número de operações no algoritmo EP cresce dependendo da relação $O(pgK^2)$, sendo necessárias $pg + p - 1$ gerações de bits de ordem K , pg seleções de ordem K , $pg + p$ cálculos da função custo e $2pg$ ordenações de ordem K .

Da mesma forma, o algoritmo EP-C possui uma complexidade que também cresce dependendo da relação $O(pgK^2)$, sendo necessárias $pg + p - 1$ gerações de bits de ordem K , $pg + p$ cálculos da função custo e $2pg$ ordenações de ordem K . Em contraste com o EP, o algoritmo EP-C realiza pg/I_C seleções de ordem K e gI_C clonagens de ordem K .

A complexidade computacional para o algoritmo GA também cresce dependendo da relação $O(pgK^2)$, podendo ser obtida adicionando a complexidade do operador recombinação genética, *crossover*, à complexidade do algoritmo EP. Esta etapa realiza $pg \cdot p_c$ operações de ordem K .

Para o algoritmo Clonalg, o número de operações cresce dependendo da relação $O(gK^2(p + N_C))$, sendo necessárias $g(p - n) + p - 1$ gerações de bits de ordem K , $2gn$ seleções de ordem K , $g(p + N_C)$ cálculos da função custo, gN_C ordenações de ordem K , gN_C clonagens de ordem K e gN_C cálculos de ζ de ordem 1.

6.2 COMPLEXIDADE PARA OS ALGORITMOS DE BUSCA LOCAL

O número de operações do algoritmo 1-opt LS cresce dependendo da relação $O(mK^3)$, sendo necessárias mK trocas de bits de ordem K , $mK + 1$ cálculos da função custo, m seleções de bits de ordem K e m comparações de ordem 1.

O algoritmo k -opt LS possui uma complexidade que cresce dependendo da relação $O(mQK^2)$, sendo necessárias mQ

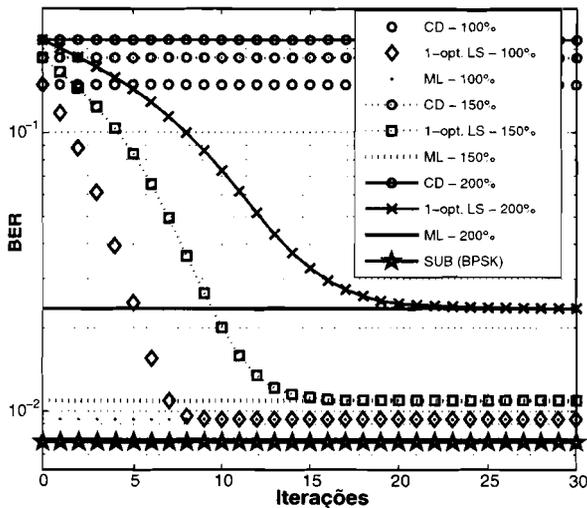


Figura 11. Desempenho do algoritmo 1-opt LS para sistema sobrecarregado.

5.3 DESEMPENHO COM ERROS NAS ESTIMATIVAS DE CANAL

Novamente, utilizou-se apenas o algoritmo 1-opt LS para a obtenção da degradação de desempenho considerando erros nas estimativas de canal, pois todos os demais algoritmos heurísticos analisados procuram maximizar a mesma função custo (equação (8)) e erros nas estimativas de canal acarretam perdas idênticas de desempenho para todos os algoritmos. Esta degradação está sintetizada na figura 12. O erro foi introduzido na estimativa do módulo (β) e/ou da fase (ϕ) dos coeficientes de canal separadamente ou em conjunto, para cada usuário e atualizados a cada símbolo transmitido. Estes erros foram modelados através de distribuições uniformes:

$$\hat{\beta}_k = \mathcal{U}[1 \pm \Delta_\beta] \times \beta_k \quad ; \quad \hat{\phi}_k = \mathcal{U}[1 \pm \Delta_\phi] \times \phi_k$$

com Δ_β e $\Delta_\phi \in [0; 0, 10; 0, 20; 0, 25; 0, 50]$.

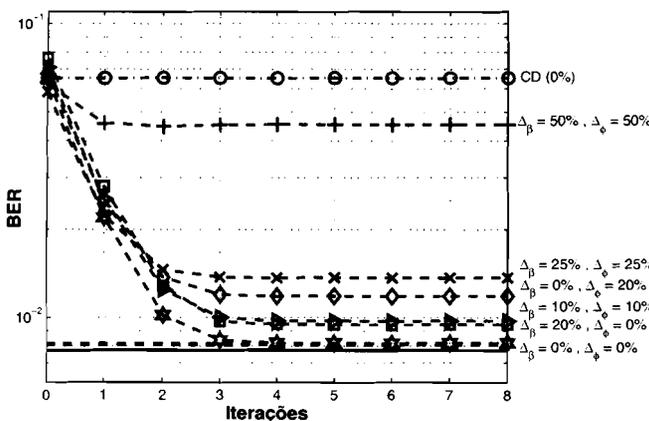


Figura 12. Degradação do desempenho para o algoritmo 1-opt LS quando há erros nas estimativas do canal, controle perfeito de potência e $K = 12$ usuários.

Mesmo com erros nas estimativas do módulo e fase dos coeficientes de canal, da ordem de $\pm 50\%$, uniformemente dis-

tribuídos no intervalo $[1 \pm 0, 5] \times \beta_k$ e $[1 \pm 0, 5] \times \phi_k$, os algoritmos heurísticos atingem melhores desempenhos que o obtido com o detector convencional com estimativas perfeitas para os coeficientes de canal. Observe-se ainda que a tolerância a erros na estimativa da fase do canal é menor que a tolerância a erros na estimativa do módulo.

6. COMPLEXIDADE COMPUTACIONAL

No intuito de expressar a complexidade dos algoritmos analisados, encontrou-se o número de operações envolvidas em cada cálculo da função custo. No cálculo da função custo, conforme a equação (8), as operações $f_1 = 2y^T c^H A$ e $f_2 = c A R A c^H$ podem ser obtidas antes do laço de busca da solução global em cada algoritmo. Para cada teste de solução candidata, computam-se $f_1 \cdot b$ e $b^T \cdot f_2 \cdot b$, que em termos de operações é equivalente a $K^2 + 2K$ multiplicações e 1 transposição de ordem K .

Para o detector ML, o número de operações cresce exponencialmente com o número de usuários, i.e., $O(2^K K^2)$, onde o operador $O(\cdot)$ indica proporcionalidade ao argumento. São necessárias 2^K gerações de bits de ordem K e 2^K cálculos da função custo para a detecção simultânea de 1 bit dos K usuários.

6.1 COMPLEXIDADE PARA OS ALGORITMOS EVOLUCIONÁRIOS

O número de operações no algoritmo EP cresce dependendo da relação $O(pgK^2)$, sendo necessárias $pg + p - 1$ gerações de bits de ordem K , pg seleções de ordem K , $pg + p$ cálculos da função custo e $2pg$ ordenações de ordem K .

Da mesma forma, o algoritmo EP-C possui uma complexidade que também cresce dependendo da relação $O(pgK^2)$, sendo necessárias $pg + p - 1$ gerações de bits de ordem K , $pg + p$ cálculos da função custo e $2pg$ ordenações de ordem K . Em contraste com o EP, o algoritmo EP-C realiza pg/I_C seleções de ordem K e gI_C clonagens de ordem K .

A complexidade computacional para o algoritmo GA também cresce dependendo da relação $O(pgK^2)$, podendo ser obtida adicionando a complexidade do operador recombinação genética, *crossover*, à complexidade do algoritmo EP. Esta etapa realiza $pg \cdot p_c$ operações de ordem K .

Para o algoritmo Clonalg, o número de operações cresce dependendo da relação $O(gK^2(p + N_C))$, sendo necessárias $g(p - n) + p - 1$ gerações de bits de ordem K , $2gn$ seleções de ordem K , $g(p + N_C)$ cálculos da função custo, gN_C ordenações de ordem K , gN_C clonagens de ordem K e gN_C cálculos de ζ de ordem 1.

6.2 COMPLEXIDADE PARA OS ALGORITMOS DE BUSCA LOCAL

O número de operações do algoritmo 1-opt LS cresce dependendo da relação $O(mK^3)$, sendo necessárias mK trocas de bits de ordem K , $mK + 1$ cálculos da função custo, m seleções de bits de ordem K e m comparações de ordem 1.

O algoritmo k -opt LS possui uma complexidade que cresce dependendo da relação $O(mQK^2)$, sendo necessárias mQ

trocas de bits de ordem K , $mQ + 1$ cálculos da função custo, m seleções de bits de ordem K e m comparações de ordem 1.

O número de operações do algoritmo SA também cresce na proporção $O(mK^3)$, onde são realizadas mK trocas de bits de ordem K , $mK + 1$ cálculos da função custo, $4m$ comparações de ordem K , $m(K + 1)$ gerações de números aleatórios, m cálculos de $z(m)$ e m cálculos de $T(m)$, todos de ordem 1.

O algoritmo STTS possui uma complexidade que cresce dependendo também da relação $O(mK^3)$, sendo necessárias mK trocas de bits de ordem K , $mK + 1$ cálculos da função custo, mK gerações de bits de ordem K , $4m$ comparações de ordem 1 e $\frac{1}{2}m(P^2 + P)$ comparações de ordem K .

Para o algoritmo RTS o número de operações também cresce na proporção $O(mK^3)$, sendo realizadas mK trocas de bits de ordem K , $mK + 1$ cálculos da função custo, mK gerações de bits de ordem K , $4m$ comparações de ordem 1 e $\frac{1}{2}m(P_m^2 + P_m)$ comparações de ordem K , onde $\langle P_m \rangle = \frac{1}{m} \sum_{i=1}^m P(i)$.

6.3 NÚMERO DE OPERAÇÕES

Admitindo que os tempos computacionais das operações sejam idênticos, pode-se expressar a complexidade computacional dos receptores, em termos de operações, substituindo o número de operações de cada função custo e somando todas as outras operações multiplicadas por suas respectivas ordens, como indicado na tabela 1.

Utilizando os valores numéricos obtidos nas simulações, figuras 5 a 10, para as variáveis g (geração em que houve convergência), K , p , I_C , N_C , n , m (número da iteração referente à convergência), Q , P e $\langle P_m \rangle$, é possível expressar a complexidade de cada algoritmo, em termos do número de operações, para se atingir o desempenho ML. A tabela 2 sintetiza estes resultados. Nota-se que para o problema da detecção multiusuário em canal Rayleigh plano síncrono, a estratégia singela de otimização 1-opt LS resulta em menor complexidade em termos do número de operações computacionais, garantindo convergência para o desempenho ML para todas as condições de operação de sistema analisadas.

Detector	Número de Operações
ML	$2^K K (K + 4)$
EP	$pg(K^2 + 7K) + K(Kp + 4p - 1)$
EP-C	$pg(K^2 + 6K + K/I_C) + K(Kp + 4p - 1 + gI_C)$
GA	$pg(K^2 + K(7 + p_C)) + K(Kp + 4p - 1)$
Clonalg	$g(p + N_C)(K^2 + 5K) + K(g(n - p + \frac{N_C}{K}) + p - 1)$
1-opt LS	$(m(K^2 + 1) + K)(K + 4) - 3m - K$
k-opt LS	$(m(QK + 1) + K)(K + 4) - 3m - K$
SA-LS	$(m(K^2 + 5) + K)(K + 4) - 17m - K$
STTS	$(mK^2 + K)(K + 5) + 0,5K(m(P^2 + P) - 4)$
RTS	$(mK^2 + K)(K + 5) + 0,5K(m(\langle P_m \rangle^2 + \langle P_m \rangle) - 4)$

Tabela 1. Complexidade dos detectores multiusuário em termos de operações.

Detector	Fig. 5 e 8	Fig. 6 e 9	Fig. 7 e 10
ML	786	$> 11 \times 10^6$	$> 11 \times 10^6$
EP	203	7845*	9630*
EP-C	140	1730	2300
GA	151	3040	4020
Clonalg	437	12234*	9820
1-opt LS	9	114	211
3-opt LS	86	6250	9370
SA-LS	10	115	212
STTS	10	119	220
RTS	10	118	219

* Complexidade estimada para uma aproximação da geração de convergência.

Tabela 2. Complexidade (núm. de operações $\times 10^3$).

7. AGRADECIMENTOS

Aos revisores pelas sugestões que muito contribuíram para o aperfeiçoamento desse artigo.

8. CONCLUSÕES

Uma grande variedade de algoritmos heurísticos aplicados ao problema MuD em canal Rayleigh Plano síncrono foram avaliados e resultaram em desempenhos muito próximos ao obtido com o detector ótimo.

Dentre os algoritmos heurísticos analisados, a escolha mais apropriada consiste em utilizar o algoritmo sub-ótimo 1-opt LS, pois este resultou em menor complexidade, atingindo o mesmo desempenho do ML para carregamentos elevados e relação sinal-ruído média. No entanto, todos os algoritmos heurísticos, com exceção do EP-C, apresentaram perda na velocidade de convergência em presença do efeito *near-far* e aumento de carregamento.

O algoritmo EP-C, apesar de resultar em uma complexidade computacional maior que o algoritmo 1-opt LS, não apresentou perda na velocidade de convergência em presença de desbalanços nas potências recebidas, devido à característica adaptativa da matriz de desvio padrão, simplificando consideravelmente o processo de determinação do número de gerações necessárias para se alcançar o desempenho do detector ML.

A análise da degradação de desempenho em função do aumento de carregamento mostrou que para até 150% os nove algoritmos heurísticos não sofrem perda de desempenho considerável quando comparados ao SuB, indicando um potencial aumento na capacidade do sistema. Já a análise da degradação de desempenho dos detectores, devido à introdução de erros nas estimativas dos coeficientes de canal, indicou de forma geral ser possível tolerar simultaneamente erros da ordem de até $\pm 25\%$ no módulo e fase, uniformemente distribuídos, sem que o desempenho seja degradado consideravelmente. Adicionalmente, o sistema DS/CDMA com detectores multiusuário baseados em algoritmos heurísticos em canal Rayleigh Plano síncrono mostrou-se mais robusto aos erros nas estimativas do módulo do que aos erros na fase dos coeficientes de canal.

REFERÊNCIAS

- [1] S. Moshavi, "Multi-User detection for DS-CDMA communications". *IEEE Commun. Mag.*, v. 34, p.132-136, Oct. 1996.
- [2] S. Verdú, *Multuser Detection*, New York: Cambridge University Press, 1998.
- [3] P. H. Tan, "Multiuser detection in CDMA-Combinatorial optimization Methods". 2001. 93 f. Monography, Chalmers University of Technology. Göteborg. 2001.
- [4] M. G. Shayesteh, M. Menhaj, B. G. Nobary, "A New Modified Genetic Algorithm for Multiuser Detection in DS/CDMA Systems", *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol.1, pp. B-11-14, Sept.2001.
- [5] X. Wu, T. C. Chuah, B. S. Sharif, O. R. Hinton. "Adaptive Robust Detection for CDMA using Genetic Algorithm", *IEE Proceeding Communications*, vol.150, no.6, pp.437-444, Dec. 2003.
- [6] M. M. S. Al-Sawafi and J. A. Jervase. "A Micro-Genetic Algorithm-based CDMA Multi-user Detector", *IEEE Proc. of the Second Annual Conference on Communication Networks and Services Research (CNSR'04)*, 2004
- [7] L. Dong, X. Youyun, S. Wentao, L. Hanwen, L. Xingzhao, "Genetic Algorithm Based Multiuser Detection for CDMA systems", *Proceedings of the IEEE 6th Circuits and Systems*, Vol. 1, pp.321-324, May 2004
- [8] K. Yen; L. Hanzo. "Genetic Algorithm Assisted Joint Multiuser Symbol Detection and Fading Channel Estimation for Synchronous CDMA Systems". *IEEE Journal on Selected Areas in Communications*. vol.19. p.985-997. 2001.
- [9] S. Verdú. "Computational Complexity of Optimum Multiuser Detection", *Algorithmica*, no. 4, pp:303-312, 1989.
- [10] M. C. Goldbarg; H. P. L. Luna, *Otimização Combinatória e Programação Linear*. Rio de Janeiro: Campus. 2000. 649 p.
- [11] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Nova York: AddisonWesley, 1989.
- [12] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.
- [13] D. B. Fogel. "An Introduction to Simulated Evolutionary Optimization". *IEEE Transactions on Neural Networks*. vol.5. p.03-13. 1994.
- [14] L. N. de Castro, F. J. Von Zuben, "Learning and Optimization Using the Clonal Selection Principle", *IEEE Transactions on Evolutionary Computation*, vol. 6, pp.239-251, Jun. 2002.
- [15] C. Reeves, *Modern Heuristic techniques for combinatorial problems*. Oxford: Blackwell Scientific, 1993.
- [16] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, pp. 671-680, 1983.
- [17] V. Cerny, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm", *Journal of Opt. theory and App.*, vol. 45, pp.41-51, 1985.
- [18] K. Katayama; H. Narihisa, "Performance of Simulated Annealing-based Heuristic for Unconstrained Binary Quadratic Programming Problem", *Eur. Journal of Operational Research*, vol.134, pp.103-119, 2001.
- [19] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers. Boston MA. 1997.
- [20] R. Battiti and G. Tecchiolli, "The Reactive Tabu Search", *ORSA Journal on Computing*, no.2, pp.126-140, 1994.
- [21] C. R. Darwin, *On The Origin of Species by Means of Natural Selection*. London: J. Murray, 1859. ISBN 0-517-12320-7.
- [22] H. S. Lim, M. V. C. Rao, A. W. C. Tan, and H. T. Chuah, "Multiuser Detection for DS-CDMA Systems Using Evolutionary Programming", *IEEE Communications Letters*, vol.7, n. 3, pp. 101-03, March 2003.
- [23] T. Abrão; F. Ciriaco; P. J. Jeszensky. "Evolutionary Programming with Cloning and Adaptive Cost Function Applied To Multi-User DS-CDMA Systems". *IEEE International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, pp. 160-163, Sep. 2004.
- [24] F. Ciriaco, T. Abrão, P. J. E. Jeszensky, "Algoritmos Heurísticos Evolucionários Aplicados à Detecção Multiusuário DS-CDMA" *XXI Simpósio Brasileiro de Telecomunicações, SBT2004*, Sept. 2004, Belém, PA.
- [25] T. Abrão, "Canceladores de Interferência Multiusuário Aplicados a Sistemas DS/CDMA de Múltipla Taxa", PhD Thesis, *Depto de Engenharia de Telecomunicações e Controle, Escola Politécnica da Universidade de São Paulo*, março, 2001.
- [26] C. Ergün; K. Hacioglu. "Multiuser Detection Using a Genetic Algorithm in CDMA Communications Systems". *IEEE Transactions on Communications*. vol.48. p.1374-1382. 2000.
- [27] F. Glover, "Heuristic for Integer Programming Using Surrogate Constraints", *Decision Sciences*, vol. 8, pp. 156-166, 1977.
- [28] F. Glover, "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research*, no. 5, pp. 533-549, 1986.
- [29] P. Dent, G E. Bottomley and T. Croft, "Jakes fading model revisited", *Electron. Lett.*, vol. 29, no. 3, pp. 1162-1163, June 1993.
- [30] M. C. Jeruchim and P. Balaban and K. S. Shanmugan. *Simulation of Communication Systems*, Pleum Press, New York, 1992.
- [31] J. Proakis. *Digital Communications*. McGraw-Hill. 1989.

Fernando Ciriaco é graduado em Engenharia Elétrica pela Universidade Estadual de Londrina em 2004. Desenvolveu projetos de Iniciação Científica com bolsa PIBIC/CNPq na área de Telecomunicações. Atualmente é aluno de mestrado em Eng. Elétrica na mesma instituição. Comunicações sem fio, sistemas DS/CDMA e detectores multiusuários constituem as principais áreas de pesquisa de seu interesse.

Taufik Abrão é graduado, mestre e doutor em Engenharia Elétrica em 1992, 1996 e 2001, respectivamente, pela Escola Politécnica da Universidade de São Paulo. Atualmente é professor adjunto AD-C do Departamento de Engenharia Elétrica da Universidade Estadual de Londrina, PR. Seus interesses de pesquisa atuais incluem sistemas CDMA, detecção multiusuário, métodos de otimização aplicados aos problemas de comunicação sem fio e projeto de seqüências de espalhamento.

Paul Jean Etienne Jeszensky obteve os títulos de Engenheiro Eletricista, Mestre em Engenharia Elétrica, Doutor em Engenharia Elétrica-Sistemas Eletrônicos e Livre Docente na área de Telecomunicações, pela Escola Politécnica da USP em 1972, 1981, 1989 e 1992, respectivamente. Autor do livro *Sistemas Telefônicos* (Editora Manole, 2004) e de cerca de 70 trabalhos técnicos/científicos, apresentados em Simpósios/Congressos/revistas especializadas, nacionais e internacionais. Professor da EPUSP desde 1978, concursado em 1984, em dedicação exclusiva a partir de 1990, na área de Telecomunicações. General Co-Chairmen do ITS'98-SBT/IEEE International Telecommunications Symposium, realizado em SP de 9 a 13/8/1998. Foi professor visitante da UPC-Universitat Politècnica de Catalunya, Barcelona-Espanha em 1995 e da TUB-Technical University of Budapest, Hungria em 2001. Suas áreas de interesse para pesquisa incluem: comunicações sem fio em geral, seqüências de códigos para uso em sistemas DS/CDMA, sistemas DS/CDMA quase síncronos e detecção multiusuário em DS/CDMA.