

# Algoritmos Heurísticos Evolucionários Aplicados à Detecção Multiusuário DS-CDMA

Fernando Ciriaco Dias Neto, Taufik Abrão e Paul Jean E. Jeszensky

**Resumo**—Neste trabalho são apresentados os principais algoritmos heurísticos baseados na teoria da evolução genética aplicados ao problema da detecção multiusuário (MuD) para sistemas DS/CDMA em canais com desvanecimento Rayleigh Plano síncrono. Uma nova forma de utilização do algoritmo de Programação Evolucionária (EP) é proposta, adaptando a matriz de variância em função dos parâmetros  $E_b/N_0$ , efeito near-far (NFR) e função custo. Adicionalmente, introduziu-se a estratégia de clonagem ao EP baseada em [1]. E por fim, é feita uma comparação entre os algoritmos através da relação desempenho versus complexidade computacional. A complexidade computacional é computada em termos do número de operações para se alcançar o desempenho ML.

**Palavras-Chave**—algoritmos heurísticos, programação evolucionária, detecção multiusuário.

**Abstract**—This work shows the main heuristic algorithms, based on genetic evolution theory, applied to the DS-CDMA multi-user detection (MuD) problem in synchronous flat Rayleigh channels. A new use form for the evolutionary programming (EP) is proposed. In this proposal the variance matrix is adapted as a function of the parameters  $E_b/N_0$ , Near-Far Ratio (NFR) and cost function. Additionally the EP cloning strategy proposed in [1] are inserted to the algorithm. Finally, a comparison between algorithms, through the performance versus computational complexity relation, is carried out. The computational complexity is shown in terms of the number of operations to reach the ML performance.

**Keywords**—heuristic algorithms, evolutionary programming, multi-user detection.

## I. INTRODUÇÃO

O sinal recebido em um sistema DS-CDMA (*Direct Sequence/Code-Division Multiple Access*) em canal com desvanecimento Rayleigh Plano não pode ser demodulado de forma ótima utilizando um detector convencional (CD), pois este é afetado pela interferência de múltiplo acesso (MAI - *Multiple Access Interference*) e pelo efeito near-far (*NFRs*), resultando num sistema que não aproveita a capacidade total do canal. Em [2] mostrou-se que o detector ótimo pode ser obtido pelo emprego de um detector de máxima verossimilhança (ML). Este detector considera que as saídas dos filtros casados apresentam estatísticas suficientes para a detecção de todos usuários.

Assim, pode-se obter o detector multiusuário ótimo, aplicando um detector de Viterbi às saídas dos filtros casados. Porém, este detector apresenta uma complexidade exponencial com o número de usuários, tornando-o inviável para implementação.

Fernando Ciriaco Dias Neto 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 Telecomunicações e Controle, São Paulo, Brasil, E-mail: pj@lcs.poli.usp.br.

Portanto, é relevante uma busca de detectores multiusuário sub-ótimos que atendam aos critérios de alta eficiência e baixa complexidade. Este trabalho faz uma síntese comparativa dos principais algoritmos heurísticos, baseados na teoria da evolução genética, aplicáveis ao problema da detecção multiusuário. Esta análise visa comparar a complexidade computacional dos algoritmos através do número de operações computacionais que cada receptor necessita para a demodulação.

## II. DESCRIÇÃO DO PROBLEMA

Em uma transmissão do tipo *binary phase-shift-keying* (BPSK) através de um canal com desvanecimento Rayleigh Plano com  $K$  usuários síncronos em um sistema de múltiplo-acesso DS-CDMA, o sinal à saída do banco de filtros casados é um vetor de comprimento  $K$  que pode ser escrito na forma matricial:

$$y = R\alpha Ab + \eta \quad (1)$$

onde:  $A$  é a matriz diagonal das amplitudes recebidas dos  $K$  usuários,  $\alpha$  é a matriz diagonal dos coeficientes do canal para os  $K$  usuários,  $b \in \{-1, +1\}^K$  é o vetor bit de informação transmitidos;  $R$  é a matriz de correlação cruzada normalizada e  $\eta$  representa o ruído AWGN à saída do banco de filtros casados.

Em [2] mostra-se que uma solução ótima para recuperar os bits de informação de (1) consiste em utilizar a saída de um detector de máxima verossimilhança (ML).

$$\hat{b} = \arg \left\{ \max_{b \in \{+1, -1\}^K} 2y^T \alpha^H A b - b^T \alpha A R A \alpha^H b \right\} \quad (2)$$

onde o operador  $(\cdot)^H$  indica transposta conjugada.

O detector multiusuário ótimo consiste na busca do melhor vetor de bits de dados em um conjunto com todas as possibilidades, ou seja, é um problema com combinação NP-completa, no qual os algoritmos tradicionais são ineficientes.

Então, o uso de algoritmos heurísticos aplicados a esse problema se torna atraente, pois são métodos que encontram soluções ótimas ou próximas na otimização de problemas de combinação em um curto tempo e espaço de busca.

Sob a restrição de um espaço de busca, todos os algoritmos heurísticos buscam melhores soluções seguindo uma função objetivo, capaz de quantificar a tendência de melhoria em relação à solução ótima. Esta função é chamada de função custo (*fitness value*), e no contexto da detecção MuD, dada pela função de verossimilhança.

$$f(s) = 2y^T \alpha^H A s - s^T \alpha A R A \alpha^H s \quad (3)$$

Cada algoritmo heurístico aplicado ao problema da detecção MuD busca maximizar (3), encontrando assim um vetor de bits

candidatos,  $s$ , cuja BER média resulta a mais próxima possível daquela obtida com um detector ML.

### III. 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],[4].

Sintetiza-se a seguir, as características dos principais algoritmos heurísticos de evolução genética aplicados ao problema da detecção MuD.

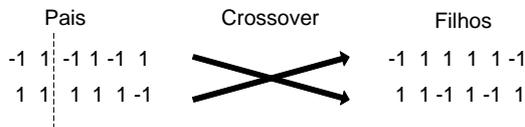
Os algoritmos genéticos constituem métodos de busca baseados em mecanismos de seleção e evolução natural seguindo a teoria da evolução das espécies de Darwin, sendo programação evolucionária (EP) e algoritmos genéticos (GA) os principais. Em [1] foi introduzido um outro algoritmo baseado na evolução genética, denominado CLONALG, o qual visa representar a resposta adaptativa da imunização de um organismo na presença de um antígeno.

Nestes algoritmos, o número de usuários é denotado por  $K$ ,  $g$  indica a iteração atual,  $G$  o número de iterações total e  $p$  o tamanho da população.

As estratégias utilizadas para a diversificação do universo de busca são baseadas nos processos que ocorrem na reprodução das células, sendo chamadas de mutação e *crossover*. O critério de mutação consiste na troca de um gene escolhido aleatoriamente por um outro que possui uma característica diferente. Ex:



O critério de *crossover* utiliza dois indivíduos, com o objetivo de formar novos indivíduos a partir da troca de genes entre os indivíduos pais. O ponto de *crossover* também é escolhido aleatoriamente. Ex:



A estratégia para a intensificação proposta neste trabalho consiste em clonar (replicar) os indivíduos que possuem melhores genes, no intuito de produzir uma população altamente evoluída na próxima geração.

Outra estratégia proposta consiste na tentativa de quantificar o número necessário de mutações que um indivíduo deve sofrer para que ele resulte em um melhor conjunto de genes. Esta estratégia visa adaptar a matriz de desvio padrão do algoritmo EP através dos parâmetros  $Eb/No$ , efeito near-far (NFR) e função custo.

O mesmo problema de detecção multiusuário pode ser atacado utilizando algoritmos heurísticos de busca local.

#### A. Programação Evolucionária (EP)

O algoritmo heurístico de programação evolucionária (EP) é o algoritmo evolucionário mais simples, pois utiliza apenas o critério de mutação, com probabilidade  $p_m$  de ocorrência, como estratégia de diversificação e nenhum critério para a estratégia de intensificação [6], [7]. Com isto, este algoritmo possui a menor complexidade computacional por geração, mas a convergência é lenta, sendo necessárias gerações extras para atingir o desempenho desejado. Descreve-se a seguir o algoritmo EP.

*Algoritmo EP:*

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

$$s_1 = \hat{b}_{CD} = \text{sign}(y) \quad (4)$$

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

$$s_i = U \{-1, +1\}^K \quad (5)$$

onde  $U \{-1, +1\}^K$  é um vetor binário uniformemente distribuído. A matriz de genes iniciais possui  $K$  linhas e  $p$  colunas (pais) e pode ser expressa por:

$$S = [s_1, \dots, s_p] \quad (6)$$

- c) Para cada  $s_i, i = 1, \dots, p$ , calcula-se o valor da função custo dada por (3):
- d) Para  $g = 1, 2, \dots, G$ :
  - (i) Para cada  $s_i, i = 1, \dots, p$ , é introduzida a mutação, no intuito de se criar filhos que possuam melhores genes.

$$S_{k,j+p} = \text{sign} \{ S_{k,j} + N(0, \sigma_{k,j}^2) \} \quad (7)$$

onde  $k = 1, \dots, K$ ;  $j = 1, \dots, p$ ;  $S_{k,j}$  representa o  $k$ -ésimo gene e  $j$ -ésimo indivíduo e  $N(0, \sigma_{k,j}^2)$  representa um processo aleatório Gaussiano com média 0 e variância  $\sigma_{k,j}^2$ . Esta variância é proporcional à porcentagem de troca dos genes, mutação, com probabilidade  $p_m$  de ocorrência.

- (ii) Para cada  $s_{i+p}, i = 1, \dots, 2p$ , é calculado o valor da função custo como em (3):
  - (iii) A população é ordenada de forma decrescente em relação ao valor obtido pela função custo. Os melhores  $p$  indivíduos, ou seja, que possuem maiores valores para a função custo, são selecionados para compor a base da nova geração.
  - (iv) Retorna-se para a etapa d) até que o número de gerações  $g$  atinja um valor  $G$  pré-estabelecido.
- e) Toma-se o indivíduo  $S_1$  como sendo o vetor de saída do algoritmo.

#### B. Programação Evolucionária com Clonagem (EP-C)

Este algoritmo também é baseado na teoria da programação evolucionária, porém aqui são aplicadas as duas estratégias de intensificação descritas anteriormente, a clonagem e a utilização da matriz de desvio padrão adaptativa.

Na obtenção da matriz de desvio padrão otimizada, analisou-se a percentagem de troca dos genes do indivíduo candidato na  $g$ -ésima geração do algoritmo EP em função do desvio padrão total,  $\sigma_{EP}$ , a qual indicou um comportamento acentuadamente crescente na faixa de  $\sigma_{EP} \approx [0, 5; 2]$ , figura 1. Otimizações feitas a seguir levaram em consideração este intervalo. Considerou-se esta faixa para  $\sigma_{EP}$ , uma vez que, valores muito abaixo ou muito acima desta faixa representam ou uma chance remota de troca de genes ou uma chance elevada de troca, respectivamente. No último caso, é mais razoável (menos complexo) gerar um novo indivíduo candidato. No primeiro caso, não há evolução (geração sem evolução).

Assim, adotando-se  $\sigma_{EP} = [0, 5; 2]$ , verifica-se que a faixa de troca de pico estará confinada ao intervalo  $\approx 5\%$  a  $60\%$ , respectivamente. Efetivamente está-se restringindo a probabilidade média de troca do gene na  $g$ -ésima geração à faixa de  $2,5\%$  a  $30\%$ , garantindo um compromisso entre variabilidade (entropia) e evolução.

Em função dos parâmetros  $SNR$ ,  $NFR$  e função custo, obteve-se uma expressão, via técnica de "curve fitting", para o desvio padrão (ou de forma equivalente, da variância) do processo EP referente à probabilidade de mudança de genes do indivíduo candidato na  $g$ -ésima geração:

$$\sigma_{EP(i,j)} = \sqrt{\frac{4}{Eb/No}} - \sqrt{\frac{NFR(i)}{32}} + 1 - \frac{\varphi(j)}{\max[\varphi(j)]} \quad (8)$$

Onde os índices  $i$  e  $j$  referem-se às linhas e colunas da matriz do desvio padrão, que refletem os genes ( $i$ -ésima linha) e os indivíduos candidatos ( $j$ -ésima coluna).

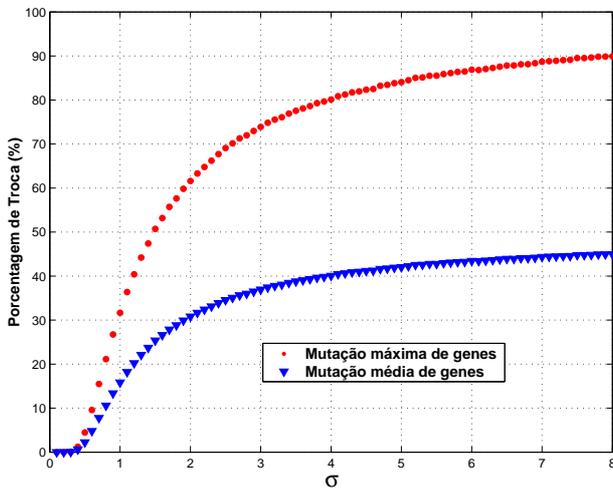


Fig. 1. Percentagem de troca dos genes em relação ao desvio padrão.

O primeiro termo em (8) é devido ao efeito da relação sinal-ruído, sendo neste caso um escalar, idêntico para todos os elementos da matriz desvio padrão ( $\sigma_{EP}$ ). O segundo termo é devido ao efeito Near-Far. Quanto maior o efeito Near-Far, menor será o desvio padrão. O efeito Near-Far é incluído em cada gene da  $i$ -ésima linha, isto é, resultará em valor idêntico para todas as gerações do EP. O último termo é devido ao efeito da função custo. Este efeito terá um impacto sobre cada indivíduo ( $j$ -ésima coluna), implicando em um 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.

A matriz de desvio padrão é calculada a cada geração (apenas o último termo), pois os valores da função custo são diferentes a cada iteração.

Para que  $\sigma_{EP} \in [0, 5; 2]$ , algumas condições são impostas.

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

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

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

A estratégia de clonagem substitui a etapa d(iii) do algoritmo EP pela seguinte etapa:

- (iii) A população é ordenada de forma decrescente em relação ao valor obtido pela função custo. Deve-se escolher apenas uma percentagem de indivíduos para que sejam usados na próxima geração. Para  $i = 1, \dots, K$  e  $j = 1, \dots, p/I_C$ :

$$S = \Omega \{S_{i,j}\} \quad (9)$$

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 usar um índice de clonagem cuja divisão  $p/I_C$  seja um número inteiro. O índice de clonagem é dado por:

$$I_C = p \cdot i\% \quad (10)$$

onde  $i\%$  é o índice de seleção dos indivíduos que possuem maior função custo e dado em porcentagem.

Estas modificações visam melhorar a taxa de convergência, diminuindo o número de gerações necessárias para se alcançar o desempenho ML.

### C. Algoritmo Genético (GA)

O algoritmo GA é parecido com o algoritmo EP, pois estes se baseiam na teoria de evolução genética, mas este utiliza os critérios de mutação e *crossover* para a estratégia de diversificação e nenhum critério para a intensificação [8], [9].

O processo de *crossover* visa criar novos indivíduos que contém características mescladas em relação à geração que os precedeu, sendo que este processo ocorre com uma alta probabilidade,  $p_c$ , de ocorrência. No algoritmo GA a porcentagem de mutação,  $p_m$ , é bem menor que a do EP.

O algoritmo GA é idêntico ao algoritmo EP incluindo o efeito do critério de *crossover*. Este critério é incluído antes da etapa d(i) e é descrita por:

- (i) É introduzido o efeito da recombinação genética, o que cria dois filhos combinando-se sub-partes de dois pais (com probabilidade de *crossover* =  $p_c$ ). A matriz de pais é dada por:

$$S = [s_1, \dots, s_p] \quad (11)$$

Então a matriz de filhos deve ser criada trocando-se bits dos vetores pais a partir de pontos de *crossover*. Estes pontos são gerados aleatoriamente.

#### D. Algoritmo de Clonagem Seletiva (CLONALG)

Este algoritmo foi proposto por [1] e é baseado na resposta imunológica de um sistema na presença de um antígeno. Quando um antígeno é introduzido o organismo produz diversos tipos de anticorpos no intuito de combater o antígeno introduzido. Apenas os anticorpos que tiverem maior afinidade com o antígeno são reproduzidos. Quando este antígeno novamente é introduzido no organismo, a resposta a ele se torna mais rápida, pois este reconhece o antígeno e já sabe quais são os melhores tipos de anticorpos. Quanto mais vezes o organismo for colocado em contato com o antígeno, mais rápido se torna à resposta a ele.

Esta análise é equivalente à estratégia do algoritmo EP, onde apenas os melhores indivíduos, que possuem melhores genes, são utilizados na próxima geração.

Este algoritmo utiliza a estratégia de mutação como critério de diversificação e o critério de clonagem como critério de intensificação.

Algoritmo CLONALG:

- a) Utiliza-se a saída do detector convencional como o vetor de genes do indivíduo inicial, como em (4).
- b) Os outros vetores de genes dos anticorpos que formam a população da 1ª geração são determinados por um processo aleatório. Para  $i = 2, \dots, p$ :

$$s_i = U \{-1, +1\}^K \quad (12)$$

onde  $U \{-1, +1\}^K$  é um vetor binário uniformemente distribuído. A matriz de genes iniciais possui  $K$  linhas e  $p$  colunas (anticorpos) e pode ser expressa por:

$$S = [s_1, \dots, s_p] \quad (13)$$

- c) Para  $g = 1, 2, \dots, G$ :
  - (i) Para cada  $s_i, i = 1, \dots, p$ , calcula-se o valor da função custo dada por (3):  
Apenas os  $n$  melhores anticorpos são selecionados.
  - (ii) Utiliza-se o critério de clonagem, criando  $N_C(i)$  cópias de cada anticorpo, onde  $N_C = \sum_{i=1}^n \text{round}\left(\frac{\beta p}{i}\right)$ , sendo  $\beta$  um fator multiplicativo que estabelece um compromisso entre complexidade e convergência [1].  
Com isso, os anticorpos são clonados com valores diferentes, dependendo da classificação em relação ao valor da função custo.
  - (iii) Para cada  $s_i, i = 1, \dots, N_C$ , é utilizado o critério da mutação, visando diversificar a busca. O critério de mutação é introduzido segundo uma função baseada na afinidade do anticorpo com o antígeno dada por:

$$\gamma = 100 \cdot \exp(-\rho f_n) \quad (14)$$

onde  $\gamma$  é a porcentagem de mutação,  $\rho$  controla o decréscimo da função e  $f_n = \frac{f_{best}}{f_i}$  é a afinidade do anticorpo ao antígeno, sendo  $f_{best}$  o valor da função custo para o melhor anticorpo e  $f_i$  o valor da função custo para o anticorpo a ser mutado.

- (iv) Para cada  $s_i, i = 1, \dots, N_C$ , calcula-se o valor da função custo dada por (3):
- (v) A população é ordenada de forma decrescente em relação ao valor obtido pela função custo. Os melhores  $n$  anticorpos, ou seja, que possuem maiores valores para a função custo, são selecionados para compor a base da nova geração.
- (vi) São gerados  $p-n$  anticorpos determinados por um processo aleatório. Para  $i = 1, \dots, p-n$ :

$$s_i = U \{-1, +1\}^K \quad (15)$$

onde  $U \{-1, +1\}^K$  é um vetor binário uniformemente distribuído. A população dos anticorpos é então formada pelos  $n$  anticorpos selecionados na etapa c(v) e pelos  $p-n$  anticorpos gerados em (15). Portanto a matriz de anticorpos possui  $K$  linhas e  $p$  colunas (anticorpos) e pode ser expressa como em (6).

- (vii) Retorna-se para a etapa c) até que o número de gerações  $g$  chegue ao valor  $G$  pré-estabelecido.
- d) Toma-se o anticorpo  $S_1$  como sendo o vetor de saída do algoritmo.

#### IV. RESULTADOS NUMÉRICOS

Nesta seção são apresentados os resultados de desempenho alcançados pelos algoritmos heurísticos evolucionários.

Os seguintes parâmetros foram considerados: seqüências aleatórias de comprimento  $N = 32$ , número de usuários  $K = 12$  e 24 usuários, resultando em um carregamento  $L = K/N = 0,375$  e 0,75 respectivamente; região de média relação sinal-ruído ( $E_b/N_0 = 15dB$ ) e 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 [10], com frequência da portadora  $f_c = 2GHz$ , número de osciladores  $N_d = 36$  e velocidades de deslocamento dos móveis uniformemente distribuídas entre 0 e 120 km/h.

Em todas as simulações Monte Carlo adotou-se um número mínimo de erros/ponto = 100. Para efeito de comparação foram incluídos os desempenhos dos detectores convencional (CD), de máxima verossimilhança (ML) e o limite SuB (*Single user Bound*) BPSK para canal Rayleigh [11].

Os valores dos parâmetros heurísticos  $p$ ,  $p_m$ ,  $i\%$ ,  $p_c$ ,  $p_m$ ,  $n$ ,  $\beta$  e  $\rho$ , adotados nas simulações, foram obtidos através de valores típicos encontrados na literatura seguido de otimização não exaustiva.

A figura 2 mostra o desempenho alcançado pelos algoritmos em um cenário com controle perfeito de potência e com baixo carregamento,  $L = 0,375$ . Adotou-se  $p = 50$  para todos os algoritmos;  $p_m = 15\%$  para o algoritmo EP;  $i\% = 10\%$  para o EP-C;  $p_c = 85\%$  e  $p_m = 5\%$  para o GA;  $n = 10$ ,  $\beta = 0,25$  e  $\rho = 5$  para o algoritmo CLONALG. Note-se que os algoritmos evolucionários convergem para o desempenho ML após um certo número de gerações. Pode-se perceber o efeito dos critérios de intensificação e de diversificação para cada algoritmo, sendo que o algoritmo EP-C alcançou o

desempenho ML mais rapidamente. Como melhoria adicional, há um ganho de convergência com a utilização da clonagem e da matriz adaptativa de desvio padrão para o algoritmo EP-C em relação ao algoritmo tradicional EP. Embora a estratégia de clonagem do algoritmo CLONALG apresente boas características, a taxa de convergência mostrou-se ser lenta, causada pela ineficiente estratégia de mutação baseada na equação (14). O algoritmo GA mostrou ter uma boa convergência, pois utiliza a estratégia de crossover e a estratégia de mutação como princípios de diversificação.

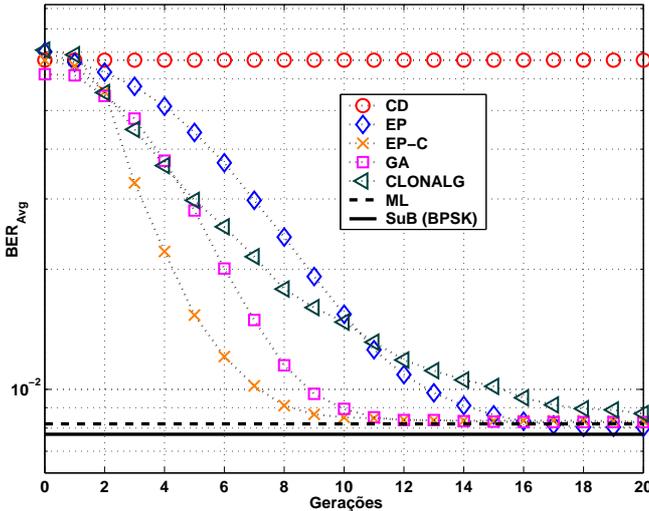


Fig. 2. Desempenho dos algoritmos heurísticos evolucionários para baixo carregamento e com controle perfeito de potência.

A figura 3 mostra o desempenho alcançado pelos algoritmos em um cenário com controle perfeito de potência e com alto carregamento,  $L = 0,75$ . Adotou-se  $p = 160$  para todos os algoritmos;  $p_m = 15\%$  para o algoritmo EP;  $i\% = 10\%$  para o EP-C;  $p_c = 85\%$  e  $p_m = 5\%$  para o GA;  $n = 20$ ,  $\beta = 0,25$  e  $\rho = 5$  para o algoritmo CLONALG.

Note-se que os algoritmos EP e CLONALG atingiram o desempenho ML para  $G = 60$  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 a detecção MuD síncrona. Embora o algoritmo GA tenha boas estratégias de diversificação, este não possui estratégia de intensificação, acarretando perda de convergência com o aumento do carregamento.

Finalmente a figura 4 mostra o desempenho alcançado pelos algoritmos em um cenário com usuários divididos em níveis distintos de potência e alto carregamento:  $L = 0,75$ , 8 usuários com NFR = 0dB, 8 usuários com NFR = 15dB e 8 usuários com NFR = 30dB, sendo o desempenho medido em relação aos usuários com menor nível de potência. Adotou-se  $p = 160$  para todos os algoritmos;  $p_m = 15\%$  para o algoritmo EP;  $i\% = 10\%$  para o EP-C;  $p_c = 85\%$  e  $p_m = 5\%$  para o

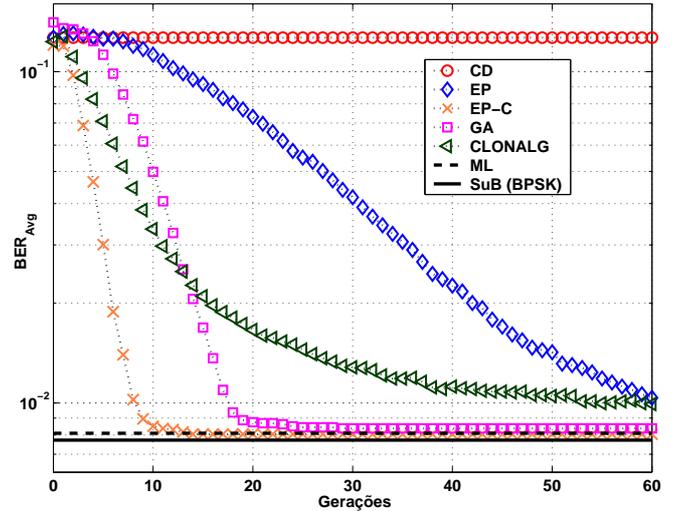


Fig. 3. Desempenho dos algoritmos heurísticos evolucionários para alto carregamento e com controle perfeito de potência.

GA;  $n = 20$ ,  $\beta = 0,25$  e  $\rho = 5$  para o algoritmo CLONALG.

Percebe-se que o algoritmo EP não converge para  $G = 60$ , evidenciando sua ineficiente estratégia de diversificação e ausência de uma estratégia de intensificação. O algoritmo CLONALG convergiu com  $g \approx 52$ , o que mostra que apesar de sua ineficiente estratégia de mutação, esta conseguiu identificar os anticorpos por afinidade devido a existência do efeito NFR, fazendo com que a porcentagem de mutação dos anticorpos com menor afinidade fosse maior.

Novamente o algoritmo EP-C mostrou possuir a melhor convergência, pois atingiu o desempenho ML com o menor número de gerações. Estes resultados provam que as estratégias de intensificação propostas neste trabalho, clonagem e matriz de desvio padrão adaptativa, são altamente eficientes para o problema da detecção multiusuário especificado.

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

## V. COMPLEXIDADE COMPUTACIONAL

No intuito de expressar a complexidade dos algoritmos analisados, deve-se encontrar o número de operações envolvidas em cada cálculo da função custo.

Cada cálculo da função custo é obtida como em (3), onde as operações  $f_1 = 2y^T \alpha^H A$  e  $f_2 = \alpha A R A \alpha^H$  podem ser obtidas antes do laço de cálculo da função custo. Para cada cálculo computa-se as operações  $f_1 \cdot b$  e  $b^T \cdot f_2 \cdot b$ , que em termos de operações é equivalente à  $K^2 + 2K$  multiplicações e 1 transposição de ordem  $K$ .

Para o detector ML, o número de operações cresce exponencialmente com o número de usuários, i.e.,  $O(2^K)$ , 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.

Para o algoritmo EP o número de operações cresce dependendo da relação  $O(pg)$ , sendo necessárias  $K, pg + p - 1$

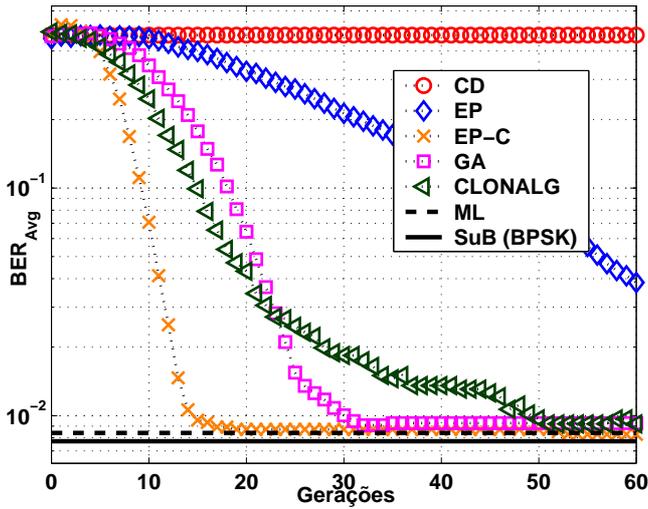


Fig. 4. Desempenho dos algoritmos heurísticos evolucionários para alto carregamento ( $K = 24$  us.) e fortes disparidades de potência.

gerações de bits de ordem  $K$ ,  $pg$  seleções de ordem  $K$ ,  $pg + p$  cálculos da função custo e  $2pg$  ordenações de ordem 1.

Da mesma forma, o algoritmo EP-C possui uma complexidade que também cresce dependendo da relação  $O(2^K)$ , sendo necessárias  $pg + p - 1$  gerações de bits de ordem  $K$ ,  $pg + p$  cálculos da função custo e  $2pg$  ordenações de ordem 1. 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  $O(2^K)$ , podendo ser obtida adicionando a complexidade da etapa de recombinação genética, crossover, à complexidade do algoritmo EP. Esta etapa realiza  $pg.p_c$  operações de ordem  $K$ .

Para o algoritmo CLONALG, o número de operações cresce dependendo da relação  $O(g(p + N_C))$ , sendo necessárias  $(p - n)g + p - 1$  gerações de bits de ordem  $K$ ,  $2ng$  seleções de ordem  $K$ ,  $(p + N_C)g$  cálculos da função custo,  $N_Cg$  ordenações de ordem 1,  $N_Cg$  clonagens de ordem  $K$  e  $N_Cg$  cálculos de  $\gamma$ .

Assumindo que o tempo computacional das operações sejam idênticos, podemos expressar a complexidade computacional dos receptores MuD, em termos de operações, substituindo o número de operações de cada função custo e somando todas as outras operações multiplicadas por suas respectivas ordens. Com isso, temos a seguinte tabela de complexidade:

Usando valores numéricos, obtidos nas simulações das Figs. 2, 3 e 4, para as variáveis  $g$  (geração em que houve convergência),  $K$ ,  $p$ ,  $I_C$ ,  $N_C$  e  $n$ , é possível expressar a complexidade de cada algoritmo, em termos do número de operações, para se atingir o desempenho ML. A tabela V sintetiza estes resultados.

## VI. CONCLUSÕES

Algoritmos heurísticos em geral, e em particular os evolucionários, aplicados ao problema da detecção multiusuário resultam em complexidade computacional muito menor que a

TABELA I

COMPLEXIDADE DOS DETECTORES MUD

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	$g (p + N_C) (K^2 + 5K) + K (g (n - p + N_C/K) + p - 1)$

TABELA II

NÚMERO DE OPERAÇÕES  $\times 10^6$

Receptor	Fig. 2	Fig. 3	Fig. 4
ML	0,786	11274,29	11274,29
EP	0,203	>7,25*	>7,85**
EP-C	0,14	1,73	2,3
GA	0,151	3,04	4,02
CLONALG	0,437	>11,33*	9,82

\* Complexidade subestimada:  $g = 60$ .

\*\* idem:  $g = 65$ .

do detector ótimo com pequena ou nenhuma degradação de desempenho.

Resultados de simulação Monte-Carlo indicaram que entre os algoritmos heurísticos analisados o EP-C, contendo duas estratégias de intensificação específicas propostas neste trabalho, clonagem e matriz de desvio padrão adaptativa, mostrou ser o mais eficiente na solução do problema da detecção MuD em sistemas DS/CDMA síncronos e canal Rayleigh Plano.

Em todas as condições de operação do sistema avaliadas (NFR = 0, e forte disparidade de potência, baixo e alto carregamento), o EP-C atingiu o desempenho ML com menor complexidade computacional que os demais algoritmos analisados.

## REFERÊNCIAS

- [1] 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.
- [2] S. Verdú, *Multuser Detection*, New York: Cambridge University Press, 1998.
- [3] M. C. Goldberg, H. P. L. Luna, *Otimização Combinatória e Programação Linear*, Rio de Janeiro: Campus, 2000.
- [4] P. H. Tan, "Multiuser detection in CDMA-Combinatorial optimization Methods", Thesis for Chalmers University of Technology, Göteborg, 2001.
- [5] F. C. Dias Neto, T. Abrão, P. J. E. Jeszensky, "Algoritmos Heurísticos de Busca Local Aplicados à Detecção Multi-usuário DS-CDMA", em processo de submissão ao XXI Simpósio Brasileiro de Telecomunicações-SBT'04.
- [6] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization", IEEE Transactions on Neural Networks, vol.5, pp.03-13, 1994.
- [7] H. S. Lim, M. V. C. Rao, A.W.C.Tan, H. T. Chuah, "Multiuser Detection for DS-CDMA Systems Using Evolutionary Programming", IEEE Communications Letters, vol.7, pp.101-103, 2003.
- [8] C. Ergün, K. Hacıoglu, "Multiuser Detection Using a Genetic Algorithm in CDMA Communications Systems", IEEE Transactions on Communications, vol.48, pp.1374-1382, 2000.
- [9] 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, pp.985-997, 2001.
- [10] P. Dent, G. E. Bottomley and T. Croft, "Jakes fading model revisited", Electron. Lett., vol. 29, no. 3, pp. 1162-1163, June 1993.
- [11] J., Proakis. *Digital Communications*. McGraw-Hill, 2nd ed., 1989.