

# Algoritmos Swarm, Genético e Programação Evolucionária Aplicados à Detecção Multiusuário

## Swarm, Genetic and Evolutionary Programming Algorithms Applied to Multiuser Detection

Fernando Ciriaco<sup>1</sup>; Leonardo D. de Oliveira<sup>2</sup>; Taufik Abrão<sup>3</sup>; Paul Jean E. Jeszensky<sup>4</sup>

### Resumo

---

Neste trabalho, é analisada, avaliada e comparada a técnica heurística evolucionária de otimização baseada no movimento buliçoso de partículas (*Swarm*), aplicada ao problema da detecção multiusuário (MuD – *Multiuser Detection*) em sistemas DS/CDMA (*Direct Sequence/Code Division Multiple Access*). Compara-se a eficiência do algoritmo de otimização *Swarm* aplicado à detecção multiusuário DS-CDMA (SWARM-MUD) por meio do compromisso desempenho *versus* complexidade computacional. A complexidade é expressa em termos do número de operações necessárias para se alcançar o desempenho obtido pelo detector ótimo ou de máxima verossimilhança, ML (*Maximum Likelihood*). A comparação é realizada entre os algoritmos genético, programação evolucionária com clonagem e o *Swarm* sob uma mesma base de simulação. Adicionalmente, é proposta uma análise de complexidade para os algoritmos MuD-heurísticos, cujo parâmetro é o número de operações computacionais. Finalmente, é feita uma análise dos parâmetros de entrada do algoritmo de otimização *Swarm*, na tentativa de encontrar parâmetros otimizados (ou quase-ótimos) para o algoritmo aplicado ao problema MuD.

**Palavras-chave:** Detecção Multiusuário. Algoritmo Genético. Programação Evolucionária. Otimização baseado no Movimento Buliçoso de Partículas. Complexidade Computacional.

### Abstract

---

In this paper, the particles swarm optimization technique, recently published in the literature, and applied to Direct Sequence/Code Division Multiple Access systems (DS/CDMA) with multiuser detection (MuD) is analyzed, evaluated and compared. The Swarm algorithm efficiency when applied to the DS-CDMA multiuser detection (Swarm-MuD) is compared through the tradeoff performance versus computational complexity, being the complexity expressed in terms of the number of necessary operations in order to reach the performance obtained through the optimum detector or the Maximum Likelihood detector (ML). The comparison is accomplished among the genetic algorithm, evolutionary programming with cloning and Swarm algorithm under the same simulation basis. Additionally, it is proposed an heuristics-MuD complexity analysis through the number of computational operations. Finally, an analysis is carried out for the input parameters of the Swarm algorithm in the attempt to find the optimum parameters (or almost-optimum) for the algorithm applied to the MuD problem.

**Key words:** Multiuser Detection. Genetic Algorithm. Evolutionary Programming. Particle Swarm Optimization. Computational Complexity.

---

<sup>1,2</sup> Alunos de mestrado, Departamento de Engenharia Elétrica, Universidade Estadual de Londrina (DEEL-UEL)

<sup>3</sup> Professor Adjunto do DEEL-UEL; taufik@uel.br

<sup>4</sup> Professor Titular da Escola Politécnica da USP, PTC; pj@lcs.poli.usp.br

## Introdução

O sinal recebido por um detector convencional (CD-*Convencional Detector*), constituído por um filtro casado (MF-*Matched Filter*) seguido de um decisor abrupto em um sistema DS-CDMA não pode ser recuperado de forma ótima, pois este é afetado pela interferência de múltiplo acesso (MAI - *Multiple Access Interference*) e pelo efeito *near-far* (NFR - *near-far ratio*). 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 potências pequenas no receptor sejam “sufocados” pelos usuários que chegam com maiores energias (VERDÚ, 1998), (ABRÃO, 2001). Disso resulta um sistema cuja capacidade está bem abaixo da capacidade do canal (MOSHAVI, 1996), (VERDÚ, 1998).

Para combater essa dificuldade e obter um aumento real de capacidade e desempenho dos sistemas de comunicação baseados em múltiplo acesso, foi proposta, nas últimas duas décadas, uma grande variedade de detectores multiusuário. O melhor desempenho é alcançado com o detector de máxima verossimilhança. O detector ML procura estimar o vetor transmitido de forma a maximizar a probabilidade de este ter sido enviado conhecendo-se o sinal foi recebido. Nesta situação, o sinal processado estende-se por toda a mensagem, com a hipótese de que todas as mensagens transmitidas sejam equiprováveis. Porém, o detector ML possui uma complexidade computacional que cresce exponencialmente com o número de usuários, inviabilizando sua implementação (VERDÚ, 1998). Dessa forma, faz-se necessário investigar soluções sub-ótimas na área de detecção multiusuário que atendam aos critérios de alto desempenho e baixa complexi-

dade.

Este trabalho desenvolve uma sistemática comparação entre o algoritmo genético, programação evolucionária com clonagem (CIRIACO; ABRÃO; JESZENSKY, 2005), (CIRIACO; ABRÃO; JESZENSKY, 2004) e o algoritmo de otimização baseada no movimento buliçoso de partículas (PSO - *Particle Swarm Optimization*) (KENNEDY; EBERHART, 1995), (KENNEDY; EBERHART, 1997), aplicáveis ao problema da detecção multiusuário em canal síncrono AWGN (ZHAO; ZENG, 2004), (LU; YAN, 2004).

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

Além desta introdução, este trabalho é composta por mais 6 seções. A segunda seção trata da descrição do modelo matemático de um sistema DS/CDMA em canal AWGN síncrono. Na terceira seção, são apresentadas as características e a descrição do algoritmo genético e do algoritmo evolucionário com clonagem. Na quarta seção, são descritos e analisados os parâmetros do algoritmo de otimização baseado no movimento buliçoso de partículas. Já na quinta seção, são apresentados resultados numéricos para diferentes condições de operação do sistema via simulação Monte Carlo. Na sexta seção, 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 sétima seção são apresentadas as principais conclusões deste estudo.

## Modelo do Sistema

Em um sistema de múltiplo acesso do tipo DS-CDMA com modulação binária por deslocamento de fase (BPSK) constituído de  $K$  usuários ativos, o sinal transmitido pelo  $k$ -ésimo é dado por (VERDÚ, 1998):

$$x_k(t) = \sqrt{2P_k} \sum_{i=1}^I b_k^{(i)} s_k(t - iT_b) \cos(\omega_c t) \quad (1)$$

onde  $P_k = A_k^2/2$  representa a potência de transmissão do  $k$ -ésimo usuário e  $A_k$  a amplitude;  $I$  é o número de bits transmitidos para cada usuário;  $b_k^{(i)}$  é o  $i$ -ésimo símbolo BPSK transmitido com período  $T_b$ ;  $\omega_c$  é a frequência da portadora;  $s_k(t)$  corresponde a seqüência de espalhamento definida no intervalo  $[0, T_b)$  e zero fora:

$$s_k(t) = \sum_{n=0}^{N-1} p(t - nT_c) \underline{s}_{k,n} \quad (2)$$

onde  $\underline{s}_{k,n} \in \{-1, 1\}$  é o  $n$ -ésimo chip da seqüência de comprimento  $N$  utilizada pelo  $k$ -ésimo usuário;  $T_c$  é o período de chip e o ganho de processamento,  $\frac{T_b}{T_c}$ , é igual a  $N$ ; a formatação de pulso,  $p(t)$ , é assumida retangular com amplitude unitária no intervalo  $[0; T_c)$  e zero fora.

Admitindo filtro passa-baixa ideal, o sinal em banda base recebido, propagado através de um canal AWGN, é:

$$r(t) = \sum_{i=1}^I \sum_{k=1}^K A_k b_k^{(i)} s_k(t - \tau_k) + \eta(t) \quad (3)$$

onde  $t \in [0, T_b]$ , a amplitude  $A_k$  é assumida constante ao longo dos  $I$  bits transmitidos,  $b_k \in \{\pm 1\}$  é o bit de informação transmitido,  $s_k$  é uma réplica da seqüência de assinatura atribuída ao  $k$ -ésimo usuário, com  $\tau_k$  o atraso aleatório correspondente para o usuário  $k$ ;  $\eta(t)$  representa o

ruído AWGN com densidade de potência bilateral igual a  $N_0/2$ . Por simplicidade de análise, considerou-se canal síncrono, ou seja,  $\tau_k = 0$ . Resultados considerando canais assíncronos (CIRIACO; ABRÃO; JESZENSKY, 2006) mostram que o desempenho é equivalente ao obtido em canais síncronos quando há estimativas confiáveis dos atrasos dos sinais na estação rádio-base.

Utilizando notação vetorial, a equação (3) pode ser re-escrita:

$$r(t) = \sum_{i=1}^I \mathbf{s}^T(t - iT_b) \mathbf{a} \mathbf{b}^{(i)} + \eta(t) \quad (4)$$

onde:  $\mathbf{s}(t) = [s_1(t); s_2(t); \dots; s_K(t)]$  é a matriz de seqüência de espalhamento dos usuários,  $\mathbf{a} = \text{diag}[\sqrt{P_1}; \sqrt{P_2}; \dots; \sqrt{P_K}]$  é a matriz diagonal das amplitudes recebidas dos usuários e o vetor de bits é dado por  $\mathbf{b}^{(i)} = [b_1^{(i)}; b_2^{(i)}; \dots; b_K^{(i)}]$ .

Para canais síncronos AWGN, o receptor convencional consiste de um banco de  $K$  filtros casados as seqüências de espalhamento dos usuários. A saída do filtro casado para o  $k$ -ésimo usuário com detecção coerente amostrado ao final do período do  $i$ -ésimo intervalo de bit pode ser expresso como:

$$\begin{aligned} y_k^{(i)} &= \int_{-\infty}^{+\infty} r(t) s_k(t - iT_b) dt \\ &= \sqrt{P_k} T_b b_k^{(i)} + I_k^{(i)} + n_k^{(i)} \end{aligned} \quad (5)$$

onde o primeiro termo corresponde ao sinal desejado, o segundo, a MAI, e o último o AWGN filtrado.

O termo da MAI,  $I_k^{(i)}$ , é dado por:

$$I_k^{(i)} = \sum_{u, u \neq k}^K \sqrt{P_u} b_u^{(i)} \mathcal{R}_{u,k} \quad (6)$$

onde  $\mathcal{R}_{u,k}$  é a medida de correlação cruzada nor-

malizada entre as seqüências de espalhamento dos usuários  $u$  e  $k$ , sendo dada por:

$$\mathcal{R}_{u,k} = \frac{1}{N} \int_0^{T_b} s_u(t)s_k(t)dt; \quad (7)$$

Utilizando notação vetorial, a saída do banco de filtros casados para o  $i$ -ésimo intervalo de bit é dado por:

$$\begin{aligned} \mathbf{y}^{(i)} &= [y_1^{(i)}; y_2^{(i)}; \dots; y_K^{(i)}] \\ &= \mathbf{Rab}^{(i)} + \mathbf{n}^{(i)} \end{aligned} \quad (8)$$

onde a matriz  $\mathbf{R}$  de dimensão  $K \times K$  é composta pelos elementos  $\mathcal{R}_{u,k}$  definidos pela equação (7):

Para estimar o  $b_k^{(i)}$  bit, o detector convencional considera apenas a parte real da saída do banco de filtros casados,  $y_k^{(i)}$ , seguido de um circuito de decisão abrupta:

$$\hat{b}_k^{(i)} = \text{sign} \left( \text{Re} \left\{ y_k^{(i)} \right\} \right) \quad (9)$$

Neste contexto, para  $K$  usuários ativos e  $I$  bits, o vetor estimado  $\mathcal{B}^{(i)}$  de dimensão  $K \times 1$  pode ser escrito:

$$\hat{\mathbf{b}}^{(i)} = [\hat{b}_1^{(i)}; \hat{b}_2^{(i)}; \dots; \hat{b}_K^{(i)}] \quad (10)$$

O desempenho obtido com o receptor convencional será consideravelmente deteriorado quando o número de usuários compartilhando o mesmo canal crescer, fato que acarreta um aumento da MAI, equação (6), e/ou quando a potência dos usuários interferentes aumentar (efeito *near-far*).

Uma solução para este problema consiste em obter decisões conjuntas de bit, mediante o uso de estratégias de detecção multiusuário. O melhor desempenho, dentre os detectores MuD, é

alcançado com o detector de máxima verossimilhança (VERDÚ, 1998). O objetivo é encontrar o vetor  $\mathcal{B}^{(i)}$  que maximiza:

$$g \left\{ y(t) | \mathcal{B}^{(i)} \right\} = \exp \left( \int_{(i-1)T_b}^{iT_b} [y(t) - S(\mathcal{B}^{(i)})]^2 dt \right) \quad (11)$$

onde  $t \in [(i-1)T_b, iT_b]$  e:

$$S(\mathcal{B}^{(i)}) = \frac{1}{\sqrt{2}} \mathbf{s}^T(t) \mathbf{a} \mathcal{B}^{(i)} \quad (12)$$

Baseado na observação do MFB, vetor  $\mathbf{y}^{(i)}$  em (8), a maximização de (11) é equivalente a selecionar o vetor de bits  $\mathcal{B}^{(i)}$  que maximiza a chamada função de verossimilhança (LLF) (VERDÚ, 1998):

$$\Omega(\mathcal{B}^{(i)}) = 2\text{Re} \left\{ \mathcal{B}^{(i)T} \mathbf{a} \mathbf{y}^{(i)} \right\} - \mathcal{B}^{(i)T} \mathbf{a} \mathbf{R} \mathbf{a} \mathcal{B}^{(i)} \quad (13)$$

Portanto, o detector de máxima verossimilhança consiste em maximizar a equação (13), estimando o vetor de bits transmitidos por todos usuários:

$$\hat{\mathbf{b}}_{ML}^{(i)} = \arg \left\{ \max_{\mathcal{B}^{(i)} \in \{\pm 1\}^K} \left[ \Omega(\mathcal{B}^{(i)}) \right] \right\} \quad (14)$$

onde  $i = 1, \dots, I$ .

Neste trabalho, adotou-se  $I = 1$ . 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. Assim, o conjunto com todas as possibilidades cresce exponencialmente com o número de usuários. Essa busca é conhecida na literatura como um problema de combinação NP-completo. Sabe-se que os algoritmos tradicionais de otimização combinatória são ineficientes no tratamento desta tipo de problema.

Métodos heurísticos aplicados a esse tipo de problema têm-se tornado atraentes, pois permitem obter soluções ótimas ou quase-ótimas, para problemas de combinação em um curto tempo e espaço de busca. Tais métodos proporcionam uma redução na 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 custo (*fitness value*), capaz de quantificar a tendência de melhoria em relação à solução ótima. No contexto da detecção MuD em canais síncronos AWGN, a função custo é expressa por (13). Dessa forma, cada algoritmo heurístico buscará maximizar a LLF testando possíveis vetores de bits transmitidos a cada nova geração. Com essas tentativas, busca-se maximizar o desempenho médio do sistema DS/CDMA e aproximá-lo daquele obtido pelo detector ML à medida que o número de gerações cresce.

## Algoritmo Genético (GA) e Programação Evolucionária com Clonagem (EP-C)

O GA e o EP-C são métodos de busca não determinísticos baseados em mecanismos de seleção e evolução natural seguindo a teoria da evolução das espécies de Darwin (DARWIN, 1859). Esta teoria descreve o comportamento complexo que ocorre na interação entre os diversos níveis biológicos: a célula, o organismo, o indivíduo e a população.

Esses algoritmos apresentam, no processo de implementação, várias etapas equivalentes, apresentando diferenças apenas na etapa de diversificação genética, etapa . Visando a uma comparação visual da diferença entre os algoritmos,

optou-se por apresentar uma descrição genérica para o código do GA e do EP-C, seguido da explicação de cada etapa. Genericamente, os algoritmos GA e EP-C são implementado baseados nos pseudo-códigos a seguir.

### Algoritmo: GA para MuD

---

**Entradas:**  $p, \mathcal{B}_1, M, G_T$       **Saídas:**  $\mathcal{B}_1$   
início  
1. Inicializa primeira população  $\mathcal{B}$ ;  $t = 0$ ;  
2. Calcula a LLF( $\mathcal{B}$ );  
3. Enquanto  $t < G_T$  faça;  
4.      $\mathcal{B}_{selected} = \text{Seleção}(\mathcal{B}, M)$ ;  
5.      $\mathcal{B}_{cross} = \text{Crossover}(\mathcal{B}_{selected})$ ;  
6.      $\mathcal{B}_{new} = \text{Muta\c{c}o\~{a}}(\mathcal{B}_{cross})$ ;  
7.     Calcula a LLF( $\mathcal{B}_{new}$ );  
8.      $\mathcal{B} = \text{Reposi\c{c}\~{a}}o(\mathcal{B} \cup \mathcal{B}_{new})$ ;  
9.      $t = t + 1$ ;  
10. fim  
fim

---

### Algoritmo: EP-C para MuD

---

**Entradas:**  $p, \mathcal{B}_1, M, G_T$       **Saídas:**  $\mathcal{B}_1$   
início  
1. Inicializa primeira população  $\mathcal{B}$ ;  $t = 0$ ;  
2. Calcula a LLF( $\mathcal{B}$ );  
3. Enquanto  $t < G_T$  faça;  
4.      $\mathcal{B}_{selected} = \text{Sele\c{c}\~{a}}o(\mathcal{B}, M)$ ;  
5.      $\mathcal{B}_{clone} = \text{Clonagem}(\mathcal{B}_{selected})$ ;  
6.      $\mathcal{B}_{new} = \text{Muta\c{c}\~{a}}o(\mathcal{B}_{clone})$ ;  
7.     Calcula a LLF( $\mathcal{B}_{new}$ );  
8.      $\mathcal{B} = \text{Reposi\c{c}\~{a}}o(\mathcal{B} \cup \mathcal{B}_{new})$ ;  
9.      $t = t + 1$ ;  
10. fim  
fim

---

## Tamanho da População ( $p$ )

A escolha do tamanho da população ( $p$ ) é um fator importante para o custo computacional e a qualidade das soluções obtidas. Este trabalho utiliza uma equação adaptada para encontrar o tamanho da população adequada ao problema da detecção multiusuário baseada em (AHN; RAMA-

KRISHNA, 2002):

$$p = 10 \cdot \left\lfloor 0.3454 \left( \sqrt{\pi(K-1)} + 2 \right) \right\rfloor \quad (15)$$

onde o operador  $\lfloor \cdot \rfloor$  retorna o menor inteiro. Esta equação é calculada no estágio de inicialização dos algoritmos e mantida constante para todas as gerações.

## Inicialização da População

Neste trabalho, adotaram-se as estimativas obtidas à saída do detector convencional como o indivíduo inicial da população, sendo que os outros membros da primeira população foram obtidos aleatoriamente:

O indivíduo inicial é obtido como:

$$\mathcal{B}_1^{(i)} = \hat{\mathbf{b}}^{(i)} \quad (16)$$

Já os outros membros são obtidos aleatoriamente:

$$\mathcal{B}_j^{(i)} = \text{rand}(K, 1) \quad (17)$$

onde  $j = 2, 3, \dots, p$  e  $\text{rand}(K, 1)$  representa um processo aleatório de dimensão  $K \times 1$  cujos termos só podem assumir os valores  $-1$  e  $+1$ .

A adoção da saída do detector convencional como indivíduo inicial visa a diminuir o número de gerações/iterações para se alcançar a convergência. Testes realizados sem a utilização da saída do detector convencional mostraram que os algoritmos continuam convergindo para a BER esperada. No entanto, o número de iterações necessárias à convergência aumenta.

## Cálculo da LLF

A medida de aptidão é uma característica intrínseca ao indivíduo, indicando, no nível

biológico, qual é a habilidade que um indivíduo possui para sobreviver a predadores, pestes e a outros obstáculos para a subsequente reprodução. Transportando para o algoritmo matemático, representa sua habilidade em produzir a melhor solução.

No contexto da detecção MuD, a medida de aptidão é obtida através da função LLF, equação (13), sendo diretamente responsável pela morte ou vida dos indivíduos (VERDÚ, 1998).

## Seleção

A proposta da etapa de seleção é privilegiar os melhores indivíduos na esperança de que seus descendentes possuam melhores aptidões. Para isso, escolhe-se o tamanho do poço de cruzamento ( $M$ ), sendo um valor que controla a pressão no processo de competição entre os indivíduos.

Ao se escolher um  $M$  pequeno, certamente os melhores indivíduos serão selecionados, no entanto, haverá uma diminuição na diversificação do universo de busca, podendo-se encontrar uma solução local. Utilizando um  $M$  grande, indivíduos com uma aptidão pequena serão selecionados e suas características ruins serão mantidas nas próximas gerações, trazendo lentidão na convergência (MITCHELL, 1998), (YEN; HANZO, 2004).

Deve-se utilizar um  $M$  que garanta velocidade na convergência e qualidade na solução encontrada. O valor de  $M$  deve estar na faixa  $2 \leq M \leq p$ .

Para o GA, adotou-se  $M = 0,1 \cdot p$ . Para o EP-C, adotou-se  $M = p \cdot i\%$ , onde  $i\%$  é o índice de clonagem, descrito na etapa .

Com isso, apenas os  $M$  melhores indivíduos

da população  $p$  são colocados no poço de cruzamento (FOGEL, 1994). Os outros  $p - M$  indivíduos com menores valores de aptidão são removidos da população.

### Diversificação Genética

Para simular o cruzamento entre os  $M$  indivíduos que foram colocados no poço de cruzamento, utiliza-se a idéia de operadores genéticos. Estes operadores são necessários para que a população se diversifique e mantenha características de adaptação adquiridas nas gerações anteriores.

O algoritmo 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 na adaptação dos algoritmos genéticos.

Para o algoritmo EP-C os operadores genéticos utilizados são a clonagem e a mutação, não existindo *crossover*.

### Operador *Crossover*

O operador *crossover* é um operador que combina o material genético de dois genitores escolhidos aleatoriamente a fim de produzir descendentes que apresentem material genético de ambos.

Neste trabalho, utilizou-se o operador denominado de *crossover* uniforme (MITCHELL, 1998). O operador *crossover* uniforme considera que cada gene (bit) é um ponto em potencial para que ocorra o crossover, sendo controlado pela chamada máscara de *crossover*. A máscara

de *crossover* é constituída de um vetor binário aleatório (“1” e “0”) de dimensão  $K \times 1$ .

O *crossover* ocorre quando na posição, que corresponde cada bit dos indivíduos, a máscara de *crossover* possuir valor “1” para aquela posição. Se a máscara possuir valor “0” na posição, não há troca de bits.

Durante a permuta de material genético entre os indivíduos, haverá uma tendência da transmissão das características dominantes para as gerações futuras. Após algumas gerações pode-se observar que uma elevada percentagem dos indivíduos possuem a presença de variáveis ‘dominantes’. Este procedimento é repetido até que  $p$  descendentes tenham sido criados.

### Clonagem

O operador clonagem consiste em clonar (replacar) os indivíduos que possuem maiores valores para a medida de aptidão (melhores genes), no intuito de produzir uma população altamente evoluída na geração posterior.

Esta percentagem de indivíduos ( $i\%$ ) com maiores valores de aptidão são clonados, e cada indivíduo é replicado  $I_c$  vezes, sendo  $I_c = M = p \cdot i\%$ . Com isso, a população volta a ter  $p$  indivíduos (clones). Deve-se usar um índice de clonagem,  $i\%$ , para que a divisão  $p/I_c$  seja um número inteiro.

### Operador Mutação

O operador de mutação consiste em alterar características dos indivíduos. Estes são necessários para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, fornecendo assim, meios para

a introdução de novos elementos na população. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero.

Uma maneira de se implementar a mutação é a geração de uma perturbação (ruído), que será adicionada a cada gene. No caso de alfabeto binário bipolarizado,  $-1$  e  $+1$ , quando a perturbação for pequena, não haverá mudança (mutação) do gene (CIRIACO; ABRÃO; JESZENSKY, 2004), (LIM et al., 2003), (ABRÃO; CIRIACO; JESZENSKY, 2004). No entanto, quando a perturbação for grande o suficiente para trocar o sinal do gene, ocorrerá mutação. Este ruído pode ser escolhido seguindo uma distribuição estatística específica. Neste trabalho adotou-se uma distribuição Gaussiana:

$$\text{descendente} = \text{sign}(\text{genitor} + \mathcal{N}(0, \sigma^2)) \quad (18)$$

onde  $\mathcal{N}(0, \sigma^2)$  representa uma distribuição Gaussiana com média zero e desvio padrão  $\sigma$ .

Esta variância está intimamente ligada à taxa média de mutação (ABRÃO; CIRIACO; JESZENSKY, 2004), ficando evidente que a mutação pode ocorrer em um ou mais pontos, ou ainda não ocorrer em nenhum, devido à natureza estocástica do processo.

O algoritmo EP-C considera o operador mutação como seu principal operador, tentando quantificar o número necessário de mutações que devem ocorrer para que um descendente apresente um melhor conjunto de genes que seu genitor (CIRIACO; ABRÃO; JESZENSKY, 2004). Esta estratégia visa criar uma matriz de desvio padrão,  $\sigma$ , considerando os parâmetros do sistema, como  $E_b/N_o$  (razão entre a energia de bit do  $k$ -ésimo usuário,  $E_{b_k} = A_k^2/T_b$ , e a densidade espectral de potência do ruído,  $N_0$ ), razão *near-*

*far* e o próprio valor para a aptidão.

Na obtenção da matriz de desvio padrão, analisou-se a percentagem de troca dos genes na etapa de reprodução da  $g$ -ésima geração do algoritmo EP-C em função de valores para o desvio padrão, a qual indicou um comportamento acen-tuadamente crescente na faixa de  $\sigma \in [0, 5; 2]$ , figura 1. Otimizações feitas a seguir levaram em consideração este intervalo. Considerou-se esta faixa para  $\sigma$ , 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 computacionalmente menos complexo gerar aleatoriamente um novo indivíduo candidato. No primeiro caso, não há evolução (geração sem evolução).

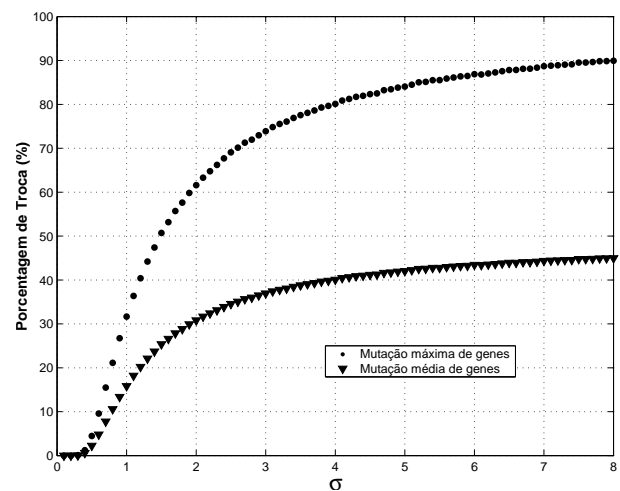


Figura 1: Percentagem de troca dos genes em rela o ao desvio padr o

Assim, adotando-se  $\sigma \in [0, 5; 2]$ , verifica-se que a faixa de troca de pico estar  confinada ao intervalo  $\approx [5\%; 60\%]$ . Efetivamente se est  restringindo a probabilidade m dia de troca do gene na  $g$ - sima gera o   faixa de  $\approx [2, 5\%; 30\%]$ , garantindo um compromisso entre variabilidade (entropia) e evolu o.



Em função dos valores para  $E_b/N_0$ , NFR e aptidão, equação (13), 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 (CIRIACO; ABRÃO; JESZENSKY, 2004):

$$\sigma(i, j) = \sqrt{\frac{4}{E_b/N_0} - \frac{NFR(i)}{32}} + 1 - \frac{\varphi(j)}{\max[\varphi]} \quad (19)$$

onde  $i = 1, \dots, K$  e  $j = 1, \dots, p$  referem-se às linhas e colunas da matriz de desvio padrão, que refletem os genes ( $i$ -ésima linha) e os indivíduos candidatos ( $j$ -ésima coluna),  $\varphi(j) = \Omega(\mathcal{B}_j^{(i)}) - \min[\Omega]$ ,  $\min[\cdot]$  e  $\max[\cdot]$  representam o operador valor mínimo e máximo, respectivamente.

O primeiro termo em (19) 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 ( $\sigma$ ). O segundo termo é devido a razão *near-far*. Quanto maior a razão *near-far*, menor será o desvio padrão. A NFR 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 o 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 calculada a cada nova geração, pois os valores da função custo são diferentes a cada geração.

Para que  $\sigma \in [0, 5; 2]$ , algumas condições foram

impostas:

- Se  $\sigma(i, j) < 0, 5$ , adota-se  $\sigma(i, j) = 0, 5$ .
- Se  $\sigma(i, j) > 2$ , adota-se  $\sigma(i, j) = 2$ .

Para o GA, a variância deve assumir valores pequenos, já que o operador mutação é secundário. Neste trabalho, adotou-se uma taxa média de mutação  $p_m = 100/K$ , que pode ser facilmente convertido em um valor de  $\sigma$  através da figura 1.

## Reposição

A estratégia de reposição procura estabelecer uma regra para a determinação do número máximo de indivíduos, vetores candidatos, a serem mantidos na próxima geração.

O presente trabalho utiliza a estratégia de reposição denominada de elitismo global (MITCHELL, 1998), (GOLDBERG, 1989), sendo mantidos para a próxima geração apenas os melhores  $p$  indivíduos oriundos da população conjunta dos genitores e descendentes.

## Algoritmo de Otimização de Baseado no Movimento Buliçoso de Partículas (PSO)

O PSO é um algoritmo desenvolvido por Kennedy e Eberhart (KENNEDY; EBERHART, 1995), (KENNEDY; EBERHART, 1997), em 1995, sendo um novo tipo de modelo de inteligência social baseado na interação de indivíduos de uma sociedade. Nele, é avaliado um aglomerado de partículas que estão distribuídas em um espaço  $n$ -dimensional, e em um dado instante de tempo, é analisado para cada partícula um deslocamento em busca de uma melhor posição.

Para o caso da detecção multiusuário com o número de bits  $I = 1$ , a posição da  $j$ -ésima partícula em sua  $t$ -ésima iteração é definida por  $\mathcal{B}_j[t] = [b_{j1}[t] \ b_{j2}[t] \ \dots \ b_{jK}[t]]^T$ , e cada dimensão pode assumir os valores 0 ou 1.

Cada partícula possui um desempenho, que é avaliado por meio da função custo. O algoritmo mantém as melhores posições individuais das partículas ( $\mathbf{p}_j$ ) e a melhor posição global encontrada até aquele instante ( $\mathbf{p}_g$ ). Através dessas duas posições e da velocidade anterior, é calculado a cada instante de tempo (geração) uma nova velocidade para cada dimensão  $d$  de cada partícula, que irá atualizar a posição de cada partícula (ZHAO; ZENG, 2004).

$$b_{jd} = b_{jd} + v_{jd} \quad (20)$$

Através desse deslocamento, busca-se a otimização da posição da partícula.

### Parâmetros de entrada do PSO

O PSO possui 4 parâmetros que influenciam em seu comportamento. O primeiro,  $\omega$ , é peso inercial, responsável por inserir a velocidade prévia no cálculo da velocidade presente. O segundo parâmetro,  $\phi_1$ , funciona intensificando a busca sobre o valor das melhores posições individuais das partículas,  $\mathbf{p}_j$ . O parâmetro  $\phi_2$  é responsável por inserir a interação entre as partículas, através da utilização da melhor posição global  $\mathbf{p}_g$ . O último parâmetro,  $V_{max}$ , limita o intervalo de busca de uma geração, criando um valor máximo e um valor mínimo para a velocidade da partícula.

As características do sistema utilizados para otimização dos parâmetros de entrada do algoritmo PSO foram: canal AWGN síncrono,  $E_b/N_o = 6dB$ , seqüência de espalhamento

aleatórias com comprimento  $N = 16$ , todos os usuários com mesma potência de sinal transmitido e número de partículas em um instante de tempo  $p = 20$ .

O método de otimização dos parâmetros consistiu de otimizações progressivas nos mesmos, sendo otimizados um a um até atingir valores que apresentassem o desempenho esperado.

Acerca do parâmetro  $\omega$ , o algoritmo mostrou-se mais eficiente para  $\omega = 1$  e constante durante todo o processo de busca. Valores altos de  $\omega$  implicaram em uma convergência rápida, porém para valores não ótimos, enquanto valores baixos resultaram em uma convergência muito lenta.

O parâmetro  $\phi_1$  insere a diversificação no algoritmo. O seu aumento implicou uma melhoria de desempenho, porém a convergência se torna mais lenta. O aumento de  $\phi_2$  produz o efeito contrário, resultando em uma convergência mais rápida, porém para um valor não ótimo. Admitindo o compromisso complexidade *versus* desempenho, foram adotado os valores  $\phi_1 = 2$  e  $\phi_2 = 2$ .

O aumento do valor de  $V_{max}$  implicou uma perda de diversificação no final da busca, enquanto a sua diminuição resultou em uma convergência mais lenta. O valor adotado foi  $V_{max} = 4$ . Para esse valor há uma chance mínima de 1,8% que o bit irá mudar de estado.

### Inicialização/Atualização de $\mathbf{p}_j$ e $\mathbf{p}_g$

Após a avaliação da população inicial por meio da função custo, a melhor posição entre todas as partículas é armazenada em  $\mathbf{p}_j$ , e a posição inicial de cada partícula é assumida como sendo sua melhor posição individual. A atualização da melhor posição individual da  $j$ -ésima partícula é realizada através da comparação da sua posição

atual com a sua melhor posição individual até aquela iteração,  $\mathbf{p}_j$ . A melhor posição global é atualizada através da comparação da melhor posição encontrada naquela iteração com o valor de  $\mathbf{p}_g$ .

### Cálculo da Velocidade

Com o uso das duas posições já calculadas previamente ( $\mathbf{p}_j$  e  $\mathbf{p}_g$ ), é calculado a cada instante de tempo (geração) uma nova velocidade para a  $j$ -ésima partícula, equacionada a seguir:

$$\mathbf{v}_j[t+1] = \omega \mathbf{v}_j[t] + \phi_1 \mathbf{U}_{j_1}[t] \cdot (\mathbf{p}_j[t] - \mathcal{B}_j[t]) + \phi_2 \mathbf{U}_{j_2}[t] \cdot (\mathbf{p}_g[t] - \mathcal{B}_j[t]) \quad (21)$$

onde  $\mathbf{v}_j$  é um vetor de mesma dimensão da posição  $\mathcal{B}_j$ ,  $\mathbf{U}_{j_1}$  e  $\mathbf{U}_{j_2}$  são matrizes diagonais com dimensão  $K$ , e seus elementos são variáveis aleatórias modeladas através de uma distribuição uniforme  $\mathcal{U}(0, 1)$  gerados para a partícula  $j$ . As velocidades iniciais das partículas são consideradas nulas.

### Atualização da Posição

A mudança de posição é feita através de  $\mathbf{v}_j$  de forma probabilística, o que implica na necessidade dos elementos  $\mathbf{v}_j$  pertencerem ao intervalo  $[0,1]$ . Para isso, à velocidade é aplicada uma transformação logística  $S(\mathbf{v}_j)$ , sendo  $S(\cdot)$  a função sigmóide. A escolha da mudança ou não da posição é analisada do seguinte modo:

1. Se  $\rho_{jd}[t] < S(v_{jd}[t])$  faça:
2.  $b_{jd}[t+1] = 1$ ;
3. caso contrário:
4.  $b_{jd}[t+1] = 0$ ;
5. fim

onde  $\rho_{jd}[t]$  é uma variável aleatória modelada por  $\mathcal{U}(0, 1)$ .

O algoritmo é implementado do seguinte modo:

Algoritmo: PSO para MuD

**Entrada:**  $\mathcal{B}_1, p, G_T, V_{max}, \phi_1, \phi_2, \omega$

**Saída:**  $\mathbf{p}_g$

início

1. Inicializa a primeira população  $\mathcal{B}$ ;  $t = 0$ ;
  2. Calcula a  $LLF(\mathcal{B})$ ;
  3. Inicializa os valores de  $\mathbf{p}_j$  e  $\mathbf{p}_g$ ;
  4. Enquanto  $t < G_T$  faça;
  5.     Calcula a velocidade  $V(\mathcal{B})$ ;
  6.     Atualiza Posição:  $\mathcal{B}_{new} = \mathcal{B} + V(\mathcal{B})$ ;
  7.     Calcula a  $LLF(\mathcal{B}_{new})$ ;
  8.     Atualiza os valores de  $\mathbf{p}_j$  e  $\mathbf{p}_g$ ;
  9.      $\mathcal{B} = \mathcal{B}_{new}$ ;
  10.     $t = t + 1$ ;
  11. fim
- fim

## Resultados Numéricos

Nesta seção são apresentados resultados de desempenho para o sistema DS/CDMA, utilizando os algoritmos PSO, GA e EP-C no processo de detecção multiusuário. Para as figuras 2 a 5, os parâmetros dos sistemas adotados estão apresentados na tabela 1.

Tabela 1: Principais Parâmetros do Sistema

Figura	$K$	$N$	$E_b/N_0$ [dB]	$NFR$ [dB]
2	10	32	8	0
3	16	32	6 e 8	0
4	16	32	8	8 (8us.)
5	32	32	8	0

Na figura 4 o desempenho apresentado considera apenas o grupo de usuários com menor potência ( $NFR = 0$ dB).

Em todas as simulações Monte Carlo adotou-se um número mínimo de erros/ponto = 30. Para

efeito de comparação foram incluídos os desempenhos do detector convencional (CD), o limite quando há apenas um único usuário ativo no sistema (SuB - *Single user Bound*) com modulação BPSK e canal AWGN (PROAKIS, 1989) e, na figura 2, o detector de máxima verossimilhança (ML). O desempenho do ML não foi inserido nas figuras 3 a 5 devido ao alto tempo necessário, uma vez que o tempo de simulação para o ML com  $K > 16$  torna-se impraticável.

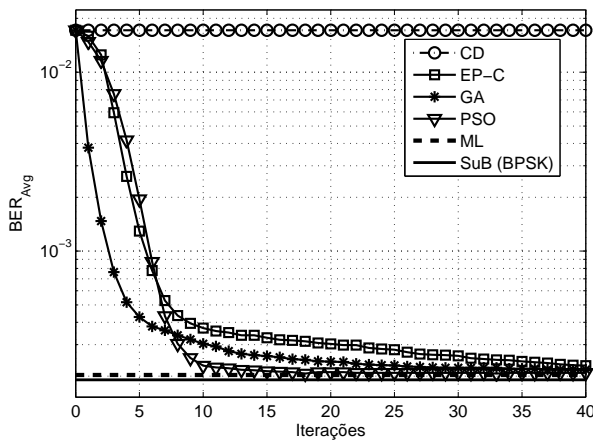


Figura 2: Convergência para os algoritmos na condição de baixo carregamento ( $L = 0,3125$ ) e com controle perfeito de potência, comparado com o ML.

Foram considerados valores otimizados para os parâmetros dos três algoritmos, com a população  $p$  segundo a equação (15); para os parâmetros de entrada, foram adotados  $i\% = 10\%$  para o algoritmo EP-C;  $p_c = 50\%$  e  $p_m = 100/K\%$  para o GA;  $V_{max} = 4$ ,  $\phi_1 = 2$ ,  $\phi_2 = 2$  e  $\omega = 1$  para o PSO.

As figuras 2 a 5 apresentam o desempenho para os três algoritmos heurísticos em função da velocidade de convergência para diferentes condições do sistema. Cada ponto nas curvas indica o desempenho na  $t$ -ésima geração/iteração.

A figura 2 mostra que, para um número de usuários  $K = 10$ , os três algoritmos avalia-

dos convergiram para o ML, demonstrando um enorme ganho de desempenho em relação ao receptor convencional

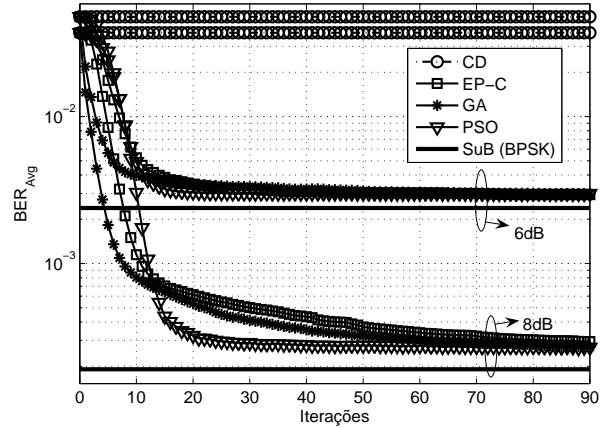


Figura 3: Convergência para os algoritmos na condição de médio carregamento, 16 usuários, com controle perfeito de potência e  $E_b/N_0 = 6$  e  $8dB$ .

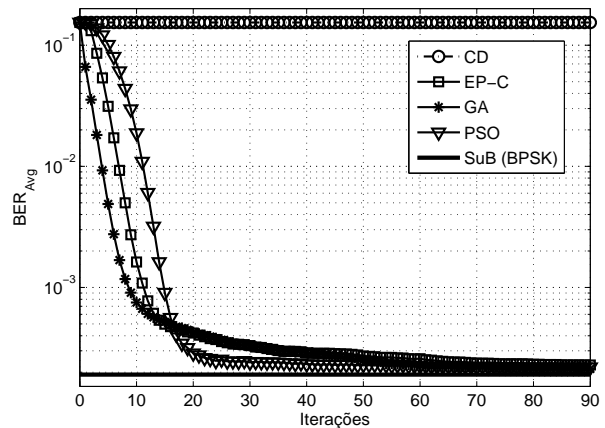


Figura 4: Convergência dos algoritmos para médio carregamento,  $E_b/N_0 = 8dB$  e efeito *near-far*.

A figura 3 mostra que os três algoritmos convergem para o provável desempenho ML, próximo ao SuB, mantendo o mesmo comportamento de desempenho também para a região de baixo  $E_b/N_0$  (6dB). As estratégias de diversificação do GA e do EP-C mostraram-se deficientes, enquanto o PSO apresentou uma boa diversificação para o problema. Comparando com a figura 2, o PSO manteve o bom desempenho, en-

quanto o GA e o EP-C apresentaram uma perda de desempenho devido ao aumento do carregamento do sistema.

A figura 4 mostra o desempenho alcançado pelos algoritmos em um cenário com usuários divididos em níveis distintos de potência e médio carregamento:  $L = 0, 5, 8$  usuários com  $NFR = 0dB$  e 8 usuários com  $NFR = 8dB$ . Notou-se que os três algoritmos continuam atingindo valores próximos ao SuB, ocorrendo ainda uma convergência mais eficiente por parte do GA e do EP-C, porém não atingindo ainda a eficiência do PSO.

Finalmente a figura 5 mostra o desempenho alcançado pelos algoritmos em um cenário com alto carregamento  $L = 1$ , mantendo-se fixo  $E_b/N_0 = 8$ . Comparando-se os gráficos 2, 3 e 5, notou-se que o algoritmo PSO apresenta um atraso inicial maior na convergência com o aumento do carregamento, porém tende a ampliar o ganho de desempenho em relação aos algoritmos GA e EP-C.

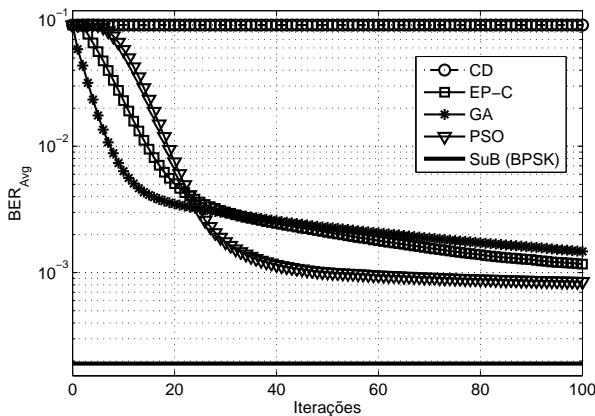


Figura 5: Desempenho dos algoritmos para carregamento  $L = 1$ , e controle perfeito de potência.

## Complexidade Computacional

Para o cálculo da complexidade, utiliza-se  $g$  como a iteração em que o algoritmo atingiu a

convergência. 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 (13), as operações  $f_1 = 2\mathbf{a}\mathbf{y}^{(i)}$  e  $f_2 = \mathbf{a}\mathbf{R}\mathbf{a}$  podem ser obtidas antes do laço de busca da solução global em cada algoritmo. Para cada teste da solução candidata, computam-se  $\mathcal{B}^{(i)T} \cdot f_1$  e  $\mathcal{B}^{(i)T} \cdot f_2 \cdot \mathcal{B}^{(i)}$ , 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.,  $O(2^K K^2)$ , 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.

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

A complexidade computacional para o algoritmo GA também cresce dependendo da relação  $O(pgK^2)$ , sendo necessárias  $pg + p - 1$  gerações de bits de ordem  $K$ ,  $pg$  seleções de ordem  $K$ ,  $pg + p$  cálculos da função custo,  $2pg$  ordenações de ordem  $K$  e  $pg$  comparações de ordem  $K$ .

A complexidade computacional para o algoritmo PSO também cresce, dependendo da relação  $O(pgK^2)$ , sendo necessárias  $(5p + 2)g + 2p - 1$  gerações de bits de ordem  $K$ ,  $g$  seleções de ordem  $K$ ,  $pg + p$  cálculos da função custo e  $(2p + 1)g$  comparações de ordem  $K$ .

Admitindo que os tempos computacionais

das operações sejam idênticos, pode-se expressar a complexidade computacional dos receptores, em termos de operações do seguinte modo: substituiu-se o número de operações de cada função custo e soma-se todas as outras operações multiplicadas por suas respectivas ordens (GOLDBARG; LUNA, 2000), como indicado na tabela 2.

Tabela 2: Complexidade dos detectores multiusuário em termos de operações

MuD	Número de Operações
ML	$2^K (K^2 + 3K)$
EP-C	$pg (K^2 + 6K + K/I_c) + K (p(K + 4) + gI_c - 1)$
GA	$pg (K^2 + 7K) + K (p(K + 3) - 1)$
PSO	$pg (K^2 + 9K) + K (p(K + 4) + 4g - 1)$

Utilizando os valores numéricos obtidos nas simulações, figuras 3 a 6, para as variáveis  $g$  (geração/iteração em que houve convergência),  $K$ ,  $p$ ,  $I_c$ , é possível expressar a complexidade de cada algoritmo, em termos do número de operações, para se atingir o desempenho ML (ou próximo). A tabela 3 sintetiza esses resultados. Nota-se que para o problema da detecção multiusuário em canal AWGN síncrono, as estratégias dos algoritmos EP-C e GA possuem uma menor complexidade, porém a sua convergência para alto carregamento não é tão eficiente quanto à do PSO, o que implica em um maior número de operações para se atingir a convergência.

Tabela 3: Número de operações

Detector	Fig. 2	Fig. 3	Fig. 4	Fig. 5
ML	$133.10^3$	$19.9.10^6$	$19.9.10^6$	$4.8.10^{12}$
EP-C	$135.10^3$	$978.10^3$	$763.10^3$	$6918.10^3$
GA	$114.10^3$	$837.10^3$	$726.10^3$	$10028.10^3$
PSO	$56.10^3$	$371.10^3$	$323.10^3$	$3202.10^3$

## Conclusões

Três algoritmos heurísticos (GA, EP-C e PSO) aplicados ao problema MuD em canal AWGN

síncrono foram avaliados e resultaram em um grande ganho de desempenho em relação ao detector convencional. O algoritmo PSO convergiu para o ML, mesmo com o aumento do número de usuários, enquanto o GA e o EP-C apresentaram uma perda de convergência com o aumento do número de usuários (sistemas com elevado carregamento). Isso mostra que a estratégia de diversificação do algoritmo PSO é mais eficiente que a do GA e EP-C. O atraso inicial do PSO se deve à velocidade inicial nula.

A análise de complexidade computacional indicou que o algoritmo PSO exige um maior número de operações por iteração, porém a sua convergência mais rápida em relação aos outros dois algoritmos analisados implica em uma menor complexidade total para o PSO.

Este trabalho mostrou que a estratégia heurística Swarm-MuD é eficiente e obteve resultados melhores em relação ao compromisso complexidade *versus* desempenho que as técnicas GA e EP-C em canal AWGN síncrono.

Análises futuras incluem extensão do problema MuD para canais mais realistas, como canal com desvanecimento multipercurso.

## Referências

ABRÃO, T. *Canceladores de Interferência Multiusuário Aplicados a Sistemas DS/CDMA de Múltipla Taxa*. Tese (Doutorado) — Departamento de Engenharia de Telecomunicações e Controle, Escola Politécnica da Universidade de São Paulo, março 2001.

ABRÃO, T.; CIRIACO, F.; JESZENSKY, P. J. E. Evolutionary programming with cloning and adaptive cost function applied to multi-user ds-cdma systems. In: *IEEE International Symposium on Spread Spectrum Techniques and Applications, ISSSTA 04*. Sydney, Australia: IEEE, 2004. p. 160–164.

AHN, C. W.; RAMAKRISHNA, R. S. A genetic algorithm for shortest path routing problem and the

- sizing of populations. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 6, p. 566–578, Dec 2002.
- CIRIACO, F.; ABRÃO, T.; JESZENSKY, P. Ds/cdma multiuser detection with evolutionary algorithms. *Journal of Universal Computer Science*, v. 12, n. 4, p. 450–480, May 2006.
- CIRIACO, F.; ABRÃO, T.; JESZENSKY, P. J. E. Algoritmos heurísticos evolucionários aplicados à detecção multiusuário ds-cdma. In: *XXI Simpósio Brasileiro de Telecomunicações, SBrT2004*. Belém, PA: Sociedade Brasileira de Telecomunicações, 2004.
- CIRIACO, F.; ABRÃO, T.; JESZENSKY, P. J. E. Detecção multiusuário utilizando algoritmos heurísticos evolucionários e de busca local. *Semina - Ciências Exatas e Tecnológicas*, v. 25, n. 2, p. 145–162, 2005.
- DARWIN, C. R. *On The Origin of Species by Means of Natural Selection*. London, U.K.: J. Murray, 1859. ISBN 0-517-12320-7.
- FOGEL, D. B. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, v. 5, p. 03–13, 1994.
- GOLDBARG, M. C.; LUNA, H. P. L. *Otimização Combinatória e Programação Linear*. Rio de Janeiro: Campus, 2000. 649 p.
- GOLDBERG, D. E. *Genetic Algorithms in Search Optimization and Machine Learning*. Nova York: AddisonWesley, 1989.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. Perth, WA.: INSPEC number 5263228, 1995. p. 1942–1948.
- KENNEDY, J.; EBERHART, R. A discrete binary version of the particle swarm algorithm. In: *IEEE International Conference on Systems, Man, and Cybernetics*. Orlando, USA: IEEE, 1997. p. 4104–4108.
- LIM, H. S.; RAO, M. V. C.; TAN, A. W. C.; CHUAH, H. T. Multiuser detection for ds-cdma systems using evolutionary programming. *IEEE Communications Letters*, v. 7, n. 3, p. 101–03, March 2003.
- LU, Z.; YAN, S. Multiuser detector based on particle swarm algorithm. In: *IEEE 6th Symp. on Emerging Technologies: Mobile and Wireless Comm*. Shanghai, China: IEEE, 2004. p. 123–127.
- MITCHELL, M. *An Introduction to Genetic Algorithms*. USA: MIT Press, 1998.
- MOSHAVI, S. Multi-user detection for ds-cdma communications. *IEEE Commun. Mag.*, v. 34, p. 132–136, Oct 1996.
- PROAKIS, J. *Digital Communications*. New York, NY, 10020: McGraw-Hill, 1989.
- VERDÚ, S. *Multiuser Detection*. Cambridge, U.K: Cambridge Univ. Press, 1998.
- YEN, K.; HANZO, L. Genetic-algorithm-assisted multiuser detection in asynchronous cdma communications. *IEEE Trans. on Vehicular Technology*, v. 53, n. 5, p. 1413–1422, Sept 2004.
- ZHAO, Y.; ZENG, J. Particle swarm optimization algorithm in signal detection and blind extraction. In: *7<sup>th</sup> International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04)*. Beijing, China: IEEE, 2004. p. 37–41.