

Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binária e real

Adair Santa Catarina^{1*} e Sirlei Lourdes Bach²

¹Curso de Informática, Universidade do Oeste do Paraná, Rua Universitária, 2069. 85814-110, Cascavel, Paraná, Brasil. ²Curso de Ciência da Computação, Unipan, Av. Brasil, 7210, 85802-970, Cascavel, Paraná, Brasil. *Autor para correspondência. e-mail: asc@unioeste.br

RESUMO. Este artigo apresenta um estudo acerca dos efeitos da variação dos parâmetros genéticos sobre a qualidade da solução e sobre o tempo de convergência em algoritmos genéticos (AGs) que se utilizam de codificações binária e real. Para tanto, desenvolveram-se 3 AGs: o primeiro, chamado AG Clássico, utiliza codificação binária e os outros dois, chamados AGM1 e AGM2, utilizam codificação real. Os parâmetros estudados para o AG Clássico foram o tamanho da população, o número de indivíduos que compõe a elite selecionada e a taxa de mutação. Para os AGM1 e AGM2, estudaram-se os parâmetros tamanho da população, número de indivíduos na nuvem aleatória gerada ao redor dos indivíduos, desvio-padrão utilizado na criação dessa nuvem, número de indivíduos na elite selecionada e taxa de mutação. Observou-se que o AG Clássico foi influenciado pelos parâmetros tamanho da população inicial, pelo número de elementos na elite selecionada e pela taxa de mutação. O AGM1 foi influenciado apenas pelo tamanho da população inicial. O AGM2 foi influenciado pelos parâmetros tamanho da população inicial, número de elementos na elite selecionada e taxa de mutação.

Palavras-chave: algoritmos genéticos, parâmetros genéticos, codificação binária, codificação real.

ABSTRACT. Study of the effect of the genetic parameters about the optimized solution and time of convergence in genetic algorithms with binary and real codification. This work presents a study about the effects of variation of genetic parameters over the solution quality and time of convergence in genetic algorithms that uses binary and real codification. Three algorithms was developed: the first called "AG Clássico" uses binary codification, the others two, called "AG M1" and "AG M2", uses real codification. The parameters studied for "AG Clássico" was the population's size, the number of individuals in selected elite and the mutation rate. The parameters studied for "AGM1" and "AG M2" was the population's size, the number of individuals in selected elite, the number of individuals in random cloud generated around the individuals, the standard-deviation used to create the cloud and the mutation rate. The "AG Clássico" was affected by the parameters size of initial population, number of individuals in selected elite and mutation rate. The "AG M1" was affected only by the size of initial population. The "AG M2" was affected by the parameters size of initial population, number of individual in selected elite and mutation rate.

Key words: genetic algorithms, genetic parameters, binary codification, real codification.

Introdução

Os algoritmos genéticos (AGs) são um tipo de algoritmo de busca que se utiliza do paradigma genético/evolucionário (Holland, 1975). Os AGs foram criados com o intuito de imitar alguns dos processos observados na evolução natural das espécies (Goldberg, 1989).

No começo dos anos 70, John Holland propôs a incorporação das características da evolução natural a

algoritmos computacionais para solucionar problemas da mesma forma que a natureza fazia para resolver os seus problemas, ou seja, usando a evolução. Acreditando nisso, Holland deu início a uma pesquisa sobre algoritmos que manipulavam *strings* de 0's (zeros) e 1's (uns), as quais foram denominadas cromossomos. Através da manipulação do conteúdo desses cromossomos, os algoritmos propostos por Holland eram capazes de encontrar

cromossomos melhores. A codificação dos dados em cromossomos compostos por *strings* de 0's e 1's é a forma clássica de codificação dos cromossomos.

A codificação clássica, quando utilizada em problemas que possuem variáveis contínuas e cujas soluções requeridas necessitam de boa precisão numérica, torna os cromossomos longos. Para cada ponto decimal acrescentado na precisão, é necessário adicionar 3,3 bits na *string*. (Galvão e Valença, 1999).

A consequência imediata do aumento da *string*, que representa o cromossomo, é o aumento no tempo necessário para calcular o equivalente decimal desse cromossomo.

Por esse motivo, formas não-clássicas de codificação dos cromossomos foram desenvolvidas, gerando codificações adequadas para problemas específicos (Herrera *et al.*, 1996).

Uma das formas de codificação mais utilizada é a codificação real. Essa forma de codificação consiste em representar, em um gene ou cromossomo, uma variável numérica contínua através de seu próprio valor real. As primeiras aplicações da codificação real foram propostas por Lucasius e Kateman (1989) e Davis (1989). A partir de então, a codificação real tornou-se padrão em problemas de otimização numérica com variáveis contínuas.

O comportamento dos AGs é influenciado pelos operadores genéticos, bem como pelos parâmetros genéticos utilizados. Os principais operadores genéticos são a seleção, o cruzamento e a mutação. Os principais parâmetros genéticos são o tamanho da população inicial, a taxa de cruzamento e a taxa de mutação.

A finalidade da seleção em um AG é escolher os elementos da população que devem se reproduzir, de tal forma que dê maior chance de reprodução àqueles mais adaptados ao meio ambiente, isto é, àqueles que apresentam melhor *fitness*. A mais conhecida e utilizada forma de se fazer a seleção é a roleta, ou algoritmo Monte Carlo (Davis, 1996; Mendes F.^o, 1998).

O elitismo é uma adição aos métodos de seleção que força os AGs a reter um certo número de "melhores" indivíduos em cada geração. Tais indivíduos podem ser perdidos se não forem selecionados para reprodução ou se forem destruídos por cruzamento ou mutação. A utilização do elitismo reduz o tempo de convergência dos AGs, mas pode fazê-los convergir para um ótimo local (Yepes, 2003).

O objetivo final dos operadores cruzamento e mutação é fazer que os cromossomos criados durante o processo de reprodução sejam diferentes dos cromossomos dos pais. O operador de

cruzamento é responsável por combinar os cromossomos dos pais na criação dos cromossomos dos filhos, e o operador de mutação é responsável pela introdução de pequenas mudanças aleatórias nos cromossomos dos filhos.

O tamanho da população determina o número de cromossomos na população, afetando o desempenho global e a eficiência dos AGs. Com uma população pequena, o desempenho pode cair, pois, desse modo, a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior.

A taxa de cruzamento determina a probabilidade em que um cruzamento ocorrerá. Quanto maior for essa taxa, mais rapidamente novas estruturas serão introduzidas na população. Mas se essa for muito alta, a maior parte da população será substituída, e pode ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.

A taxa de mutação determina a probabilidade com que uma mutação ocorrerá. Uma baixa taxa de mutação previne que uma dada solução fique estagnada em um valor, causando uma convergência prematura, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta, a busca se torna essencialmente aleatória.

Os parâmetros genéticos acima citados afetam diretamente o desempenho dos AGs. Os efeitos decorrentes da escolha inadequada desses parâmetros vão desde aumento no tempo de convergência, convergência prematura, estagnação da busca, maior necessidade de recursos computacionais até a não-convergência para uma solução viável.

Cursi e Cortes (1999) estudaram a influência dos métodos aleatórios de perturbação em problemas de otimização, principalmente os métodos baseados em distribuições gaussianas. Uma das conclusões desse trabalho é que o resultado da otimização realizada não é dependente da população inicial. Para vários testes realizados, o resultado ótimo foi encontrado em tempo e precisão similares.

O objetivo deste trabalho é apresentar o resultado de um estudo acerca do efeito da variação dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em AGs que utilizam

codificação binária e codificação real. Para tanto, implementaram-se três algoritmos genéticos. O primeiro dos AGs, nomeado AG Clássico, utiliza codificação binária; o segundo e o terceiro, nomeados AGM1 e AGM2 respectivamente, utilizam codificação real. Os AGs AGM1 e AGM2 diferem quanto aos operadores genéticos utilizados. O AGM1 não utiliza os operadores de cruzamento e mutação.

Material e métodos

O problema a ser otimizado

O problema escolhido foi a localização do ponto máximo da função

$$f(x) = x \cdot \sin(10\pi \cdot x) + 5$$

no intervalo (0; 4). O ponto máximo está localizado nas coordenadas (3,85026; 8,85013). A característica que motivou a escolha dessa função foram os diversos máximos locais encontrados no intervalo definido (Figura 1).

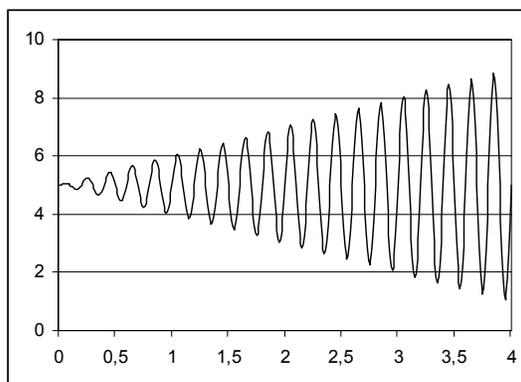


Figura 1. Gráfico da função $f(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 5$

Os AGs implementados

Os AGs foram implementados no ambiente integrado de desenvolvimento Borland Delphi 6, em linguagem Object Pascal. Esse ambiente é compatível com o sistema operacional Microsoft® Windows.

O primeiro AG implementado, o AG Clássico, utiliza codificação binária e os operadores de seleção, cruzamento em um ponto e mutação. O cromossomo foi codificado em um único gene com 18 alelos: 2 para a parte inteira e 16 para a parte fracionária; essa codificação assegurou uma precisão da ordem de $1,526 \cdot 10^{-5}$. Utilizou-se o operador de cruzamento em um ponto; o ponto de corte é fixo, o 9.º alelo do cromossomo. A mutação é trivial; inverte-se o valor do bit escolhido.

O segundo AG implementado, o AGM1, utiliza codificação real e é desprovido dos operadores de cruzamento e mutação. Nesse caso, devido à codificação real, a precisão foi superior a $1,0 \cdot 10^{-8}$.

O terceiro AG implementado, o AGM2, utiliza codificação real e apresenta precisão numérica igual ao AGM1. Nesse protótipo, implementaram-se os operadores de mutação e cruzamento. Pela utilização do operador de cruzamento, necessitou-se implementar o operador de seleção. Utilizou-se também a técnica do Elitismo. O operador de cruzamento utilizado foi o cruzamento aritmético de Michalewicz (Michalewicz, 1994).

A seguir são apresentados os algoritmos correspondentes aos AGs implementados. Através da análise destes algoritmos, é possível identificar as principais diferenças existentes entre os mesmos (Figuras 2 a 4).

1. Gerar a população inicial;
2. Repita até obter 250 gerações sem evolução
 - 2.1 Avaliar a população;
 - 2.2 Selecionar a Elite;
 - 2.3 Repita até gerar nova população
 - 2.3.1 Selecionar dois indivíduos;
 - 2.3.2 Cruzar esses indivíduos gerando dois novos indivíduos;
 - 2.4 Realizar a mutação da população gerada;
 - 2.5 Verificar se houve evolução;

Figura 2. Algoritmo do AG Clássico

1. Gerar a população inicial;
2. Repita até obter 250 gerações sem evolução
 - 2.1 Avaliar a população;
 - 2.2 Para cada indivíduo de população inicial
 - 2.2.1 Gerar uma nuvem de indivíduos com distribuição normal de média igual ao indivíduo atual e desvio-padrão adotado;
 - 2.2.2 Avaliar a nuvem gerada;
 - 2.2.3 Selecionar o melhor indivíduo da nuvem;
 - 2.2.4 Se o melhor indivíduo selecionado na nuvem for melhor que o indivíduo atual, copiá-lo para a nova população, senão manter o indivíduo atual na nova população;
 - 2.3 Verificar se houve evolução;

Figura 3 Algoritmo do AGM1

1. Gerar a população inicial;
2. Repita até obter 250 gerações sem evolução
 - 2.1 Avaliar a população;
 - 2.2 Para cada indivíduo de população inicial
 - 2.2.1 Gerar uma nuvem de indivíduos com distribuição normal de média igual ao indivíduo atual e desvio-padrão adotado;
 - 2.2.2 Avaliar a nuvem gerada;
 - 2.2.3 Selecionar o melhor indivíduo da nuvem;
 - 2.2.4 Se o melhor indivíduo selecionado na nuvem for melhor que o indivíduo atual, substituí-lo na população inicial;
 - 2.3 Selecionar a Elite;
 - 2.4 Repita até gerar nova população
 - 2.4.1 Selecionar dois indivíduos;
 - 2.4.2 Cruzar estes indivíduos gerando um novo indivíduo;
 - 2.5 Realizar a mutação da população gerada;
 - 2.6 Verificar se houve evolução;

Figura 4. Algoritmo do AGM2

Coleta e análise de dados

O AG Clássico foi executado num total de 180 vezes. Nessas 180 execuções, variaram-se, com 15 repetições cada, os parâmetros:

- tamanho da população: 10 ou 20 indivíduos;
- número de indivíduos na elite: 0 ou 1 indivíduo;
- taxa de mutação: 0,5%, 2% ou 5%.

O AGM1 foi executado num total de 120 vezes. Nessas 120 execuções, variaram-se, com 15 repetições cada, os parâmetros:

- tamanho da população: 10 ou 20 indivíduos;
- tamanho da nuvem de indivíduos: 25 ou 50 indivíduos;
- desvio-padrão utilizado na criação da nuvem de indivíduos: 0,1 ou 0,2.

A nuvem de indivíduos é um conjunto de valores aleatórios com distribuição normal de média igual ao valor do indivíduo de referência e um desvio-padrão qualquer. Para cada indivíduo da população, cria-se uma nuvem com a função de explorar o espaço de busca na vizinhança do mesmo.

O AGM2 foi executado 720 vezes. Nessas 720 execuções, variaram-se, com 15 repetições cada, os parâmetros:

- tamanho da população: 10 ou 20 indivíduos;
- tamanho da nuvem de indivíduos: 25 ou 50 indivíduos;
- desvio-padrão utilizado na criação da nuvem de indivíduos: 0,1 ou 0,2;
- número de indivíduos na elite: 0 ou 1 indivíduo;
- taxa de mutação: 0,5%, 2% ou 5%.

Os níveis para o tamanho de população foram definidos de forma a proporcionar uma cobertura razoável (10 indivíduos) e uma boa cobertura (20 indivíduos) do espaço de busca. Yepes (2003) afirma que populações pequenas podem convergir para ótimos locais e que populações maiores evitam esse problema, mas tornam o tempo de convergência maior.

O número de indivíduos na elite foi definido como zero ou um. Zero indica a não utilização do elitismo, enquanto um indica o uso da técnica; neste caso, preserva-se apenas o melhor indivíduo da população. Segundo Mendes F.^o (2000), o elitismo é uma técnica que acelera a convergência de um AG.

Os níveis para a taxa de mutação foram definidos com base em valores recomendados na literatura. A literatura recomenda utilizar taxa de mutação entre 0,5% e 5%. Valores maiores que 5% tornam a busca aleatória. Os níveis para o tamanho da nuvem de indivíduos e o desvio-padrão utilizados no AGM1 e no AGM2 foram definidos empiricamente.

A taxa de cruzamento foi mantida fixa e igual a 80% tanto no AG Clássico como no AGM2. A literatura da área recomenda utilizar taxa de cruzamento entre 60% a 80%.

Resultados e discussão

Após a coleta das informações, realizou-se a análise de variância (Anova) para comparar o efeito da variação dos parâmetros tamanho da população, número de indivíduos na elite, taxa de mutação, tamanho da nuvem de indivíduos e desvio-padrão utilizado na criação da nuvem de indivíduos sobre a solução otimizada e sobre o tempo de convergência, de acordo com o algoritmo implementado. O nível de significância empregado na análise foi de 5%.

AG clássico

Para as 180 execuções do AG Clássico obtiveram-se os resultados apresentados nas Tabelas de 1 a 3. Os índices sobrescritos juntos aos valores nas Tabelas de 1 a 12 têm a seguinte interpretação: letras iguais, resultados estatisticamente iguais ao nível de 5% de significância, segundo a Anova.

Tabela 1. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na população inicial.

N.º de indivíduos na população inicial	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
10	90	8,6781 ^a	10,036 ^b
20	90	8,7008 ^a	9,962 ^b

Tabela 2. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na elite selecionada.

N.º de indivíduos na elite	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0	90	8,6962 ^a	11,268 ^b
1	90	8,6827 ^a	8,73 ^c

Tabela 3. Média da solução otimizada e média do tempo de convergência variando-se a taxa de mutação.

Taxa de mutação (%)	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0,5	60	8,5207 ^a	10,661 ^d
2	60	8,7179 ^b	9,723 ^d
5	60	8,8297 ^c	9,613 ^d

Os testes realizados com o AG Clássico mostraram que os níveis escolhidos para os fatores número de indivíduos na população inicial e número de elementos na elite selecionada não influenciaram a qualidade da solução encontrada. O único fator que exerceu influência sobre a qualidade da solução otimizada foi a taxa de mutação. O menor valor para a taxa de mutação resultou na pior

solução, demonstrando que uma baixa taxa de mutação prejudica a busca por uma solução ótima global.

O tempo de convergência foi afetado somente pelo fator número de indivíduos na elite selecionada. Esse fato comprova a hipótese de que o elitismo contribui para uma convergência mais rápida dos algoritmos genéticos.

AGM1

Para as 120 execuções do AGM1 obtiveram-se os resultados apresentados nas Tabelas de 4 a 6.

Tabela 4. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na população inicial.

N.º de indivíduos na população inicial	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
10	60	8,85013 ^a	9,946 ^b
20	60	8,85013 ^a	12,128 ^c

Tabela 5. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na nuvem gerada.

N.º de indivíduos na nuvem gerada	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
25	60	8,85013 ^a	10,605 ^b
50	60	8,85013 ^a	11,469 ^b

Tabela 6. Média da solução otimizada e média do tempo de convergência variando-se o desvio-padrão utilizado na criação da nuvem de indivíduos.

Desvio-padrão	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0,1	60	8,85013 ^a	10,828 ^b
0,2	60	8,85013 ^a	11,245 ^b

Os testes realizados com o AGM1 mostraram que nenhum dos fatores escolhidos interfere na qualidade da solução otimizada. Quanto ao tempo de convergência, este foi afetado somente pelo fator número de indivíduos na população inicial. Como era de se esperar, uma população inicial reduzida faz que o algoritmo convirja mais rapidamente para a solução.

AGM2

Para as 720 execuções do AGM2, obtiveram-se os resultados apresentados nas Tabelas de 7 a 11.

Tabela 7. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na população inicial.

N.º de indivíduos na população Inicial	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
10	360	8,85013 ^a	10,363 ^b
20	360	8,85013 ^a	12,362 ^c

Tabela 8. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na nuvem gerada.

N.º de indivíduos na nuvem	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
25	360	8,85013 ^a	10,972 ^b
50	360	8,85013 ^a	11,752 ^c

Tabela 9. Média da solução otimizada e média do tempo de convergência variando-se o desvio-padrão utilizado na criação da nuvem de indivíduos.

Desvio-padrão	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0,1	360	8,85013 ^a	11,508 ^b
0,2	360	8,85013 ^a	11,216 ^b

Tabela 10. Média da solução otimizada e média do tempo de convergência variando-se o número de indivíduos na elite selecionada.

N.º de indivíduos na elite	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0	360	8,85013 ^a	11,813 ^b
1	360	8,85013 ^a	10,911 ^c

Tabela 11. Média da solução otimizada e média do tempo de convergência variando-se a taxa de mutação.

Taxa de mutação (%)	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
0,5	240	8,85013 ^a	10,820 ^b
2	240	8,85013 ^a	11,556 ^c
5	240	8,85013 ^a	11,710 ^c

Os testes realizados com o AGM2 mostraram que nenhum dos fatores escolhidos interfere na qualidade da solução otimizada. Quanto ao tempo de convergência, este foi afetado pelos fatores número de indivíduos na população inicial, número de indivíduos na nuvem gerada, desvio-padrão utilizado na criação da nuvem de indivíduos e taxa de mutação. Conforme esperado, os níveis mais altos dos fatores número de indivíduos na população inicial e número de indivíduos na nuvem gerada contribuem para aumentar o tempo necessário para que o AGM2 convirja para uma solução otimizada.

Assim como no AG Clássico, o uso do elitismo proporcionou uma redução no tempo de convergência para uma solução otimizada. Níveis mais altos do fator taxa de mutação resultaram em aumento no tempo de convergência, indicando que, para o AGM2, uma baixa taxa de mutação é recomendada.

Comparação entre os AGs implementados

A Tabela 12 apresenta a comparação realizada com os três algoritmos implementados.

O AG Clássico difere dos AGM1 e AGM2 quanto à solução encontrada. O AG Clássico forneceu, em média, soluções inferiores àsquelas

apresentadas pelos outros dois algoritmos, que usam codificação real. Em média, o AG Clássico convergiu em tempo inferior aos tempos dos algoritmos AGM1 e AGM2.

Tabela 12. Média da solução otimizada e média do tempo de convergência considerando os 3 algoritmos implementados.

Algoritmo	Repetições	Média da solução otimizada	Média do tempo de convergência (s)
AG Clássico	180	8,6894 ^a	9,999 ^c
AGM1	120	8,8501 ^b	11,037 ^d
AGM2	720	8,8501 ^b	11,362 ^d

Considerações finais

No estudo realizado, verificou-se que os AGs implementados são extremamente robustos; mesmo com uma escolha não adequada dos parâmetros genéticos, eles convergem para uma solução otimizada.

Quanto ao AG Clássico, observou-se que:

- a variação do parâmetro genético tamanho da população inicial não influenciou a solução otimizada nem o tempo de convergência;
- a adoção do elitismo também não influenciou a solução otimizada, mas reduziu o tempo de convergência do mesmo;
- a taxa de mutação foi o único fator que afetou a solução otimizada. Baixas taxas de mutação prejudicam a busca por uma solução ótima global, pois não permitem ao algoritmo explorar todo o espaço das soluções viáveis.

Quanto ao AGM1, observou-se que:

- nenhum dos fatores escolhidos interfere na qualidade da solução otimizada;
- o tempo de convergência foi afetado somente pelo fator número de indivíduos na população inicial. Uma população inicial reduzida faz que o algoritmo convirja para uma solução num tempo menor.

Quanto ao AGM2, observou-se que:

- nenhum dos fatores escolhidos interfere na qualidade da solução otimizada;
- o tempo de convergência aumentou quando aumentou o número de indivíduos na população inicial e o número de indivíduos na nuvem gerada;
- o uso do elitismo reduziu o tempo necessário para que o algoritmo convirja;
- o aumento na taxa de mutação resultou em aumento no tempo de convergência, indicando ser recomendada uma baixa taxa de mutação para o algoritmo.

Os testes realizados confirmam os resultados de

Cursi e Cortes (1999). O tamanho da população inicial não interfere na qualidade da solução otimizada quando se utiliza codificação real. Percebeu-se, porém, que interfere no tempo de convergência.

Os três AGs implementado proporcionaram boas soluções. O AGM1 e o AGM2, que trabalham com codificação real, apresentaram soluções melhores que o AG Clássico, que trabalha com codificação binária; isso se justifica pela baixa precisão utilizada neste último; entretanto o aumento da precisão fará que o gene adquira um tamanho maior, aumentando o custo de decodificação e, conseqüentemente, aumentando o tempo de convergência do AG Clássico.

Referências

- CURSI, J. E. S.; CORTES, M. B. S. Approximate gaussian distributions in optimization by random perturbation methods. *Applied Numerical Mathematics*, Amsterdam, v. 30, n.1, p. 23-30, 1999.
- DAVIS, L. Adapting operator probabilities in Genetic Algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3, 1989, San Mateo, *Proceedings...* San Mateo, 1989, p. 61-69.
- DAVIS, L. *Handbook of Genetic Algorithms*. Reissue edition. Stamford: International Thomson Publishing, 1996.
- GALVÃO, C. O.; VALENÇA, M. J. S. *Sistemas inteligentes: aplicações a recursos hídricos e sistemas ambientais*. Porto Alegre: Ed. Universidade/UFRGS/ABRH, 1999.
- GOLDBERG, D. E. *Genetic algorithms in search, optimization & machine learning*. Reading: Addison-Wesley, 1989.
- HERRERA, F. *et al.* Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. *Artif. Intell. Rev.*, Dordrecht, v. 12, n. 4, p. 265-319, 1998.
- HOLLAND, J. H. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1975.
- LUCASIU, C. B.; KATEMAN, G. Applications of genetic algorithms in chemometrics. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3, 1989, San Mateo, *Proceedings...* San Mateo, 1989, p.170-176.
- MENDES F^o, E. F. *Algoritmos Genéticos*. 1998. Disponível em: <<http://www.icmsc.sc.usp.br/~prico/gene1.html>>. Acesso em 20 mar. 2000.
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. 3. ed. Berlin: Springer-Verlag, 1994.
- YEPES, I. *Uma incursão aos algoritmos genéticos*. 2003. Disponível em <<http://www.geocities.com/igoryepes/>>. Acesso em 20 set. 2003.

Received on July 29, 2003.

Accepted on November 13, 2003.