

# ESTUDO DO EFEITO DOS PARÂMETROS GENÉTICOS DE UM ALGORITMO GENÉTICO NA SOLUÇÃO OTIMIZADA E NO TEMPO DE CONVERGÊNCIA EM UMA FUNÇÃO DE DUAS VARIÁVEIS

Marcelo Henrique dos Santos

Universidade de Uberaba, Engenharia de Computação, Uberaba – MG, marcelo.h.santtos@gmail.com

Daniele Carvalho Oliveira

Universidade Federal de Uberlândia, Faculdade de Ciência da Computação, Uberlândia – MG,  
daniele@cidaeli.com.br

Cintia Carvalho Oliveira

Universidade Federal de Uberlândia, Faculdade de Ciência da Computação, Uberlândia – MG,  
cintia@cidaeli.com.br

Leandro Martins Silva

Instituto Federal do Triângulo Mineiro, Uberaba – MG,  
leandromartins@iftm.edu.br

*Resumo - O objetivo deste artigo é desenvolver um Algoritmo Genético (AG) como solução a uma função matemática conhecida na literatura e observar os efeitos da variação dos parâmetros genéticos na qualidade da solução e no tempo de convergência de acordo com o número de iterações. Assim, desenvolvem-se 3 AGs: AG1, AG2 e AG3 que se diferenciam pelo operador de crossover utilizado. Foram aplicados nesses AGs o método de crossover de um ponto, crossover de dois pontos e crossover uniforme respectivamente. Os parâmetros estudados foram: quantidade de iterações, os métodos de seleção por roleta e torneio como operador de seleção e no operador genético de mutação alterou-se a taxa de mutação. Para comparar e testar o desempenho dos algoritmos genéticos implementados em cada simulação foi calculado a eficácia do valor encontrado como solução pelo AG em relação ao valor exato da função objetivo em questão.*

**Palavras-Chave – Algoritmos Genéticos, Computação Evolucionária, Inteligência Computacional, Otimização.**

## STUDY OF THE EFFECT OF GENETIC PARAMETERS OF A GENETIC ALGORITHM IN OPTIMIZED SOLUTION AND TIME OF CONVERGENCE IN A FUNCTION OF TWO VARIABLES

**Abstract - The objective of this paper is to develop a Genetic Algorithm (GA) as a solution to a mathematical function known in literature and observe the effects of variation of genetic parameters on the quality of solution and the convergence time according to the number of iterations. Thus, they develop three AGs: AG1, AG2 and AG3 distinguished by the crossover operator used. GAs have been applied in such a way crossover point, two crossover points and uniform crossover respectively. The parameters studied were: number of iterations, methods of selection by roulette and tournament selection operator as the operator and genetic mutation altered the**

**mutation rate. To compare and test the performance of genetic algorithms implemented in each simulation was calculated effective value found as a solution by GA compared to the exact value of the objective function in question.**

**Keywords - Genetic Algorithms, Evolutionary Computation, Computational Intelligence, Optimization.**

## I. INTRODUÇÃO

A computação evolucionária é um ramo da Inteligência Artificial que usam modelos computacionais dos processos naturais de evolução descritos por Darwin na Teoria da Evolução [1] como uma ferramenta para resolver problemas.

Os Algoritmos Genéticos (AG) assim como os demais algoritmos evolucionários, funcionam mantendo uma população de estruturas, denominadas indivíduos ou cromossomos, onde o comportamento dessas estruturas é semelhante à evolução das espécies. A essas estruturas são aplicados os chamados operadores genéticos, como *crossover* (recombinação) e mutação. Cada cromossomo recebe uma avaliação pertinente a sua qualidade como solução para o problema proposto. A partir dessa avaliação serão aplicados os operadores genéticos de forma a simular a sobrevivência do mais apto.

O objetivo deste trabalho é desenvolver e testar um Algoritmo Genético (AG) como solução para uma função de duas variáveis. A função avaliada neste artigo está presente na literatura, portanto, os resultados obtidos poderão ser comparados a resultados já obtidos. Assim foram implementados três algoritmos genéticos: AG1, AG2 e AG3 respectivamente.



XI CEEL – ISSN 2178-8308  
25 a 29 de novembro de 2013  
Universidade Federal de Uberlândia – UFU  
Uberlândia – Minas Gerais – Brasil

O objetivo deste artigo é investigar o desempenho do AG testado de acordo com o efeito dos operadores genéticos e de seleção sobre a solução otimizada da função proposta e sobre o tempo de convergência de acordo com o número de iterações.

Este trabalho foi dividido em cinco sessões, incluindo a introdução. Na sessão 2 são apresentados os Algoritmos Genéticos (AG) que constituem uma técnica de busca que se baseia em um processo biológico da evolução natural. Na sessão 3 é abordada a função matemática utilizada e o AG é apresentado como método para resolução da função matemática proposta. Na sessão 4 são apresentados os dados computacionais obtidos durante as execuções dos AGS. As principais conclusões e os trabalhos futuros são apresentados na sessão 5.

## II. ALGORITMOS GENÉTICOS

O algoritmo genético (AG) é um ramo de pesquisa da Computação Evolucionária e pode ser definido como uma técnica de busca baseada em uma metáfora do processo biológico de evolução natural [2].

O cientista inglês Charles Darwin estudando as espécies e suas evoluções, coletou durante anos um volumoso material que demonstrou, principalmente, a existência de inúmeras variações em cada espécie [3]. Juntamente com pesquisas de demais cientistas os estudos de Darwin evidenciaram que as espécies realmente se modificam.

Segundo Darwin na natureza todos os indivíduos dentro de um ecossistema competem entre si por recursos limitados e aqueles indivíduos dentro uma mesma espécie que não obtiverem sucesso, terão menor chance de sobreviverem e a probabilidade de ter seus genes propagados em gerações posteriores será menor [3].

Os algoritmos genéticos (AGs) são algoritmos de busca e otimização baseados nesse processo de seleção natural descrito por Darwin. Em que os melhores indivíduos de uma população estão mais aptos a sobreviverem e reproduzirem mais.

Na década de 1960, John Holland pesquisador da Universidade de Michigan, inventou os Algoritmos Genéticos. A partir de seus estudos sobre a evolução das espécies Holland publicou em 1975 seu livro, no qual observou um problema em forma de uma função matemática, onde os indivíduos melhores avaliados receberam valores maiores da função e cada indivíduo é uma possível solução [1].

Para a reprodução desse conceito em um modelo computacional, o Algoritmo Genético mais Básico (AGB) cria uma população inicial e aplica, ao longo de iterações, operadores genéticos como o de seleção, cruzamento e mutação. O AG procura uma solução muito boa, ou a melhor, por meio da criação genética de populações de indivíduos cada vez mais aptos. O fluxograma de execução de um AGB é mostrado na Figura 1.

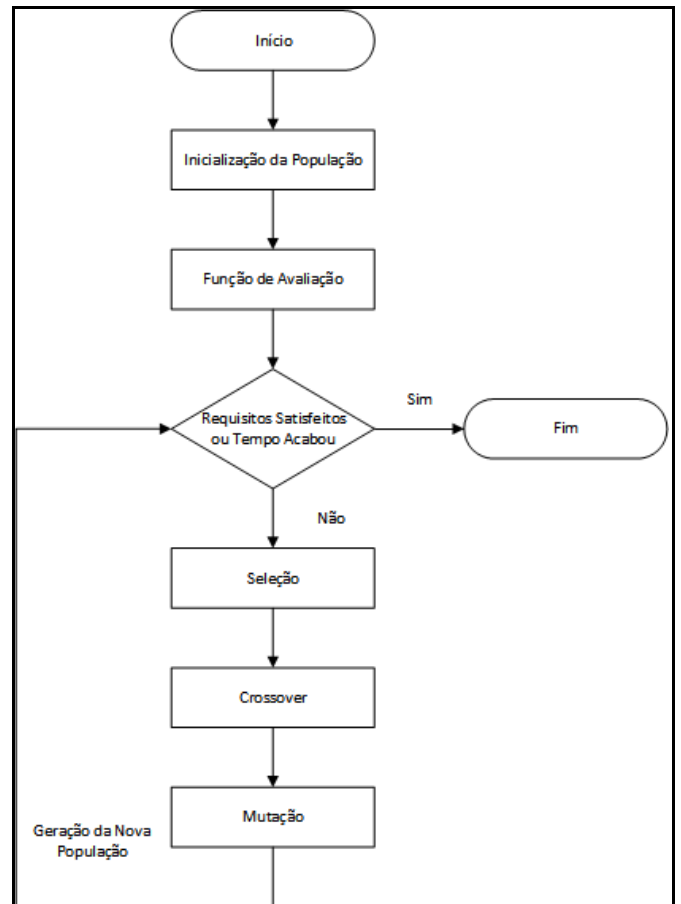


Fig. 1. Fluxograma de um algoritmo genético básico (AGB).

Segundo Goldberg [4], os algoritmos genéticos se diferenciam da maioria dos métodos de busca e otimização por quatro motivos:

- Trabalham em um espaço de soluções codificadas e não diretamente no espaço de busca;
- Trabalham com um conjunto de pontos (população) e não a partir de pontos isolados;
- Não necessitam de derivadas ou outro conhecimento auxiliar, pois utilizam informações de custo ou recompensa (função objetivo); e
- Usam regras de transição probabilísticas.

### A. Representação Cromossomial

Para que as configurações sejam avaliadas pela função objetivo (*fitness*), é necessário que a informação esteja codificada em uma estrutura cromossômica. A forma de implementar a codificação depende do problema a ser estudado. Basicamente, ela consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser tratada pelo computador [5, 2].

Os problemas podem ser trabalhados com variáveis inteiras, binárias, ou reais. Normalmente, a representação mais usada e encontrada na literatura é a representação binária, onde, cada cromossomo é uma sequência de bits e cada bit é um gene. A Figura 2 ilustra um cromossomo binário criado de maneira aleatória com comprimento  $c = 8$ .

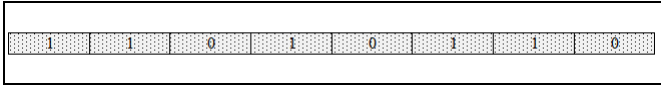


Fig. 2. Representação de um cromossomo de codificação binária.

### B. Função Avaliação (fitness)

Em problemas de otimização existe sempre uma função a ser maximizada ou minimizada. A função de avaliação é a forma com que se pode avaliar a qualidade de um indivíduo do AG como possível solução para o problema em análise. Desta forma é possível diferenciar estruturas cromossômicas que apresentarão boas soluções dos cromossomos que apresentarão soluções não satisfatórias.

A função de avaliação é a única ligação verdadeira do programa com o problema real [2]. É importante destacar que essa função pode ser uma função real, discreta ou até mesmo uma função de inteiros.

### C. Seleção

O objetivo do método de seleção é selecionar os indivíduos que virão a ser utilizados pelos operadores genéticos de mutação e crossover (recombinação), garantindo que seus descendentes farão parte da próxima geração de indivíduos. O operador de seleção tenta simular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes geram mais filhos, mas mesmo os pais menos aptos também podem gerar descendentes [2].

A seleção de indivíduos para serem submetidos aos operadores genéticos deve ser realizada de forma que os indivíduos melhores avaliados pela função de avaliação tenham uma maior chance em relação aos indivíduos que receberem uma pior avaliação, isso viabiliza a presença das melhores características em gerações futuras.

Existem vários métodos de realizar a seleção de indivíduos que serão aplicados aos operadores genéticos, porém, os métodos de seleção por roleta e torneio são os mais comuns.

#### 1) Método da roleta

No método da roleta é criada uma roleta (virtual) onde cada indivíduo de acordo com sua aptidão atribuída pela função de avaliação recebe um pedaço na roleta. Então é girada a roleta e o cromossomo escolhido é aquele em que a roleta para. Indivíduos com maior aptidão são contemplados com uma maior fatia da roleta, justamente por ocuparem um espaço maior do que os demais indivíduos que recebem uma avaliação ruim.

A Equação (1) mostra que a probabilidade de seleção ( $p_{sel}$ ) é a razão entre a aptidão do indivíduo ( $f_i$ ) e o somatório das aptidões dos indivíduos da população ( $N$ ).

$$p_{sel} = \frac{f_i}{\sum_{i=1}^N f_i} \quad (1)$$

#### 2) Método do torneio

Diferente do método da roleta a seleção por torneio é a competição entre  $t$  indivíduos selecionados aleatoriamente da população. O vencedor é aquele dentre os sorteados que obtém uma maior avaliação. O torneio pode possuir qualquer

quantidade de competidores. Normalmente, ele é realizado entre 2 indivíduos  $t = 2$  [2].

### D. Operador de crossover

O operador de *crossover*, cruzamento ou recombinação é o responsável pela troca das características entre dois indivíduos, mecanismo este, que permite que novas e melhores gerações sejam construídas durante a execução do AG. Existem diversos tipos de cruzamentos encontrados na literatura como, por exemplo, o operador de *crossover* uniforme ou a utilização de operadores com porcentagens variáveis, porém o mais utilizado é o *crossover* de um ponto.

#### 1) Crossover de um ponto

No *crossover* de um ponto após a escolha de dois indivíduos (pais) pelo operador de seleção o operador de *crossover* simples ou de um ponto escolhe aleatoriamente uma posição no cromossomo (ponto de corte) e a partir desse ponto realiza a troca de genes entre os pais para formar um novo indivíduo. A metade à esquerda do ponto de corte compõe o material genético de um filho e a metade à direita vai para o outro. A Figura 3 ilustra a operação.

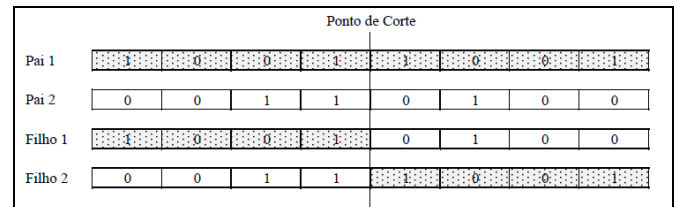


Fig. 3. Crossover de um ponto.

#### 2) Crossover de dois pontos

O crossover de dois pontos é similar ao *crossover* de um ponto onde, são selecionados aleatoriamente dois pontos de corte para a estrutura cromossômica e, depois disso, ocorre à troca de características entre os pais selecionados. A Figura 4 ilustra a operação.

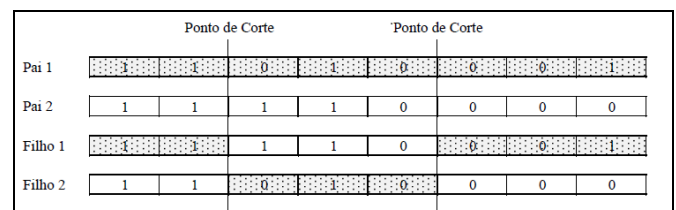


Fig. 4. Crossover de dois pontos.

#### 3) Crossover uniforme

No crossover uniforme para cada gene é sorteado um número 0 ou 1. Caso o valor sorteado é igual a 1, o Filho 1 recebe o gene da posição corrente do Pai 1 e o Filho 2 o gene corrente do Pai 2. Por outro lado, se o valor sorteado é igual a 0, as atribuições são invertidas. O operador de *crossover* de dois pontos é mais rápido do que o *crossover* uniforme, visto que existem menos sorteios por reprodução, porém, graças à sua maior capacidade de combinar esquemas, o *crossover* uniforme tende a obter resultados superiores [2]. A operação de *crossover* uniforme é representada pela Figura 3.

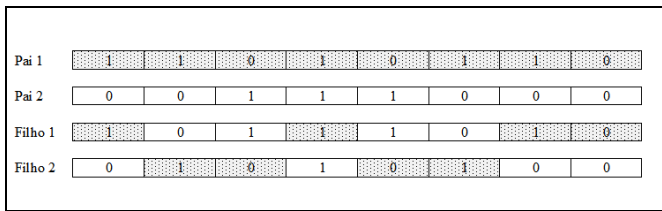


Fig. 3. Crossover uniforme.

### E. Operador de mutação

O operador genético de mutação no processo de execução de um AG possibilita o surgimento de novos indivíduos no decorrer das gerações mantendo assim a diversidade genética da população. A mutação executa um papel secundário em um AG, porém, ela é de grande importância, pois possibilita alterar aleatoriamente um ou mais componentes em uma estrutura cromossômica assim restaurando a diversidade genética que naturalmente se perde durante o processo de execução [4].

A mutação ocorre sobre os bits de um cromossomo. Nas codificações binárias ocorre uma simples troca de um valor pelo outro. A aplicação do operador de mutação em um indivíduo com codificação binária é representada na Figura 5.

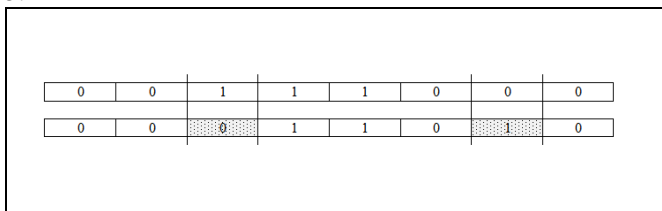


Fig. 5. Operador de mutação.

### F. Elitismo

O conceito de elitismo é manter um número  $n$  de indivíduos melhores avaliados pela função avaliação, ou seja, as características avaliadas como melhor solução são copiadas para a geração seguinte garantindo que esses cromossomos não sejam perdidos ao se descartar a população atual pela nova população ou mesmo pelos operadores genéticos de recombinação ou mutação. Esse processo permite que o melhor indivíduo da próxima geração contenha pelo menos as mesmas ou melhores características do que a geração atual.

### G. Critério de parada

Existem diversos critérios para encerrar a execução de um AG. Alguns dependem da evolução dos resultados produzidos pelo algoritmo como solução para o problema proposto e outros são pré-definidos previamente.

Alguns desses critérios mais utilizados para terminar um algoritmo genético são:

1. Após executar uma quantidade de gerações pré-estabelecidas;
2. Quando as avaliações dos melhores indivíduos de uma população acabam se tornando semelhantes;
3. Quando se conhece a melhor solução do problema a ser otimizado pela função de avaliação; e

4. Quando a população perde a diversidade durante as iterações e acabam não evoluindo.

## III. MATERIAL E MÉTODOS

### A. Função Objetivo

O principal objetivo deste artigo é analisar como se comporta os AGs implementados na otimização de uma função de duas variáveis, observando o espaço de busca por soluções para as funções.

O problema escolhido foi a localização do ponto máximo da função mostrada na Equação (2).

$$f(x, y) = 4 - x^2 - y^2 \quad (2)$$

A Figura 6 ilustra o gráfico da função mostrada na Equação (2). O domínio dessa função está no intervalo de  $[0;16]$ . O valor máximo de  $f(x, y) = 4 - x^2 - y^2$  é  $f(0, 0) = 4$ .

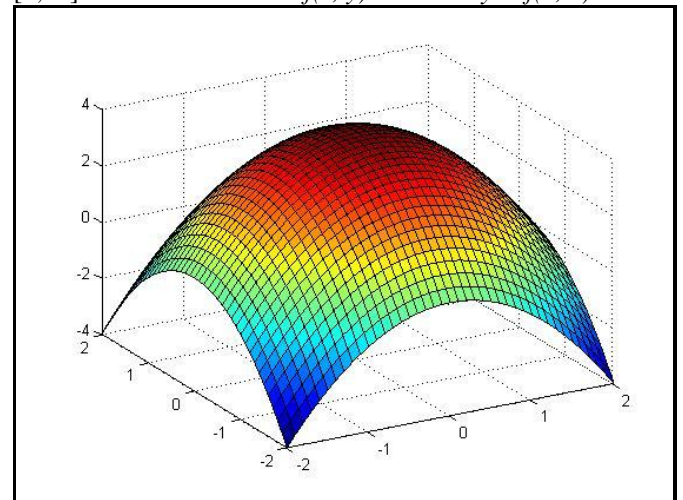


Fig. 6. Gráfico da função  $f(x, y) = 4 - x^2 - y^2$

### B. Algoritmos Genéticos Implementados

Foi desenvolvido um sistema na linguagem de programação Java que utiliza os algoritmos genéticos, com o objetivo de observar diversos tipos de funções matemáticas. Esses experimentos computacionais foram realizados em um notebook DELL Inspiron N4030 com processador Intel(R) Core(TM) i3 CPU M 380 2.53 GHz, 4 GB de memória RAM e sistema operacional Microsoft Windows 8 Pro 64 bits.

Para a implementação do **AG1**, **AG2** e **AG3** a **inicialização da população** foi de forma aleatória, onde, cada cromossomo utilizou a **representação cromossomial** mostrada na Figura 2, onde, para cada variável foi utilizado 22 bits sendo 4 bits para parte inteira e os 18 bits restantes para a parte fracionária. Esses cromossomos foram avaliados segundo a **função de avaliação (fitness)** que é a própria função objetivo. O **tamanho da população** foi de 60 indivíduos.

Foram aplicados como **operador de seleção** os métodos da roleta e do torneio e como **operador de crossover** foi utilizado o método de *crossover* de um ponto para **AG1**, *crossover* de dois pontos para **AG2** e *crossover* uniforme para **AG3**. Foram realizadas simulações para cada um dos três algoritmos genéticos alterando o número de iterações, operador de mutação e operador de seleção com a finalidade de encontrar a melhor solução e observar como a troca desses



parâmetros influi no resultado obtido e no tempo de execução.

O elitismo foi utilizado em todos os AGs implementados, pois o tempo de execução acrescido pelo seu uso quase não altera o tempo de processamento, mas garante que o desempenho do AG sempre cresça com o decorrer das gerações [2].

#### IV. RESULTADOS OBTIDOS

Foi realizada uma simulação com os AGs propostos. Alterou-se o número de iterações, operador de mutação e operador de seleção com a finalidade de encontrar a melhor solução e observar como a troca desses parâmetros influi no resultado obtido e no tempo de execução.

No AG1, AG2 e AG3, foram variados os seguintes parâmetros:

- taxa de mutação: 1 %, 3% e 7%;
- número de iterações: 75, 150 e 300 execuções;
- operador de seleção: roleta e torneio.

A comparação da eficácia de cada simulação do algoritmo para o método exato é calculado pela razão entre o valor exato e a melhor solução encontrada pelo AG, como mostrado na Equação (4):

$$E_f = \frac{f_o}{f_{AG}} \times 100 \quad (4)$$

onde:

$E_f$  - eficácia (%);

$f_o$  - valor do máximo global da função proposta;

$f_{AG}$  - valor da melhor solução obtida pelo AG (R\$);

A Tabela I mostra os valores obtidos pelo AG1 como solução após a média de 10 execuções utilizando como operadores de seleção o método da roleta e o método do torneio com 2 indivíduos.

Tabela I – Média da solução otimizada obtida após 10 execuções do AG1 e seu respectivo tempo computacional.

iterações	Operador de seleção	Taxa de mutação	Solução otimizada	Tempo de convergência (ms)
75	Roleta	1%	3,99997883073	7862,10
150	Roleta	1%	3,99999974039	15929,90
<b>300</b>	<b>Roleta</b>	<b>1%</b>	<b>3,99999999939</b>	<b>31492,10</b>
75	Roleta	3%	3,99998465977	7940,90
150	Roleta	3%	3,99999860568	15722,00
300	Roleta	3%	3,99999998561	32363,80
75	Roleta	7%	3,99990755917	8167,60
150	Roleta	7%	3,99999106294	15906,00
300	Roleta	7%	3,99999914765	32222,40
75	Torneio	1%	3,99758896742	2205,70
150	Torneio	1%	3,99999498122	4306,90
<b>300</b>	<b>Torneio</b>	<b>1%</b>	<b>3,99999943946</b>	<b>8517,50</b>
75	Torneio	3%	3,99366589119	2227,00
150	Torneio	3%	3,99987170158	4311,60
300	Torneio	3%	3,99999099762	8637,20
75	Torneio	7%	3,99724516432	2231,30
150	Torneio	7%	3,99707808405	4433,70
300	Torneio	7%	3,99962841151	8647,80

Os resultados obtidos pelo método da roleta foram ligeiramente melhores comparado ao método do torneio.

O tempo de convergência foi aproximadamente 3,5 vezes maior e apresentou linearidade em relação ao número de iterações. A taxa de mutação ajustada em taxa menores em ambos os métodos de seleção obteve resultados melhores e um número maior de iterações obteve melhores resultados embora o tempo de convergência seja maior.

A Tabela II ilustra os valores obtidos pelo AG2 como solução após a média de 10 execuções utilizando como operadores de seleção o método da roleta e o método do torneio.

Tabela II – Média da solução otimizada obtida após 10 execuções do AG2 e seu respectivo tempo computacional.

iterações	Operador de seleção	Taxa de mutação	Solução otimizada	Tempo de convergência (ms)
75	Roleta	1%	3,99989353353	7967,60
150	Roleta	1%	3,99999975004	15619,20
<b>300</b>	<b>Roleta</b>	<b>1%</b>	<b>3,99999999988</b>	<b>31467,70</b>
75	Roleta	3%	3,99993761506	8033,50
150	Roleta	3%	3,99999872474	15718,20
300	Roleta	3%	3,99999999832	31496,40
75	Roleta	7%	3,99991125587	7945,40
150	Roleta	7%	3,99995756105	21267,90
300	Roleta	7%	3,99999614624	34015,60
75	Torneio	1%	3,99306962070	2281,00
150	Torneio	1%	3,99988280148	4456,10
<b>300</b>	<b>Torneio</b>	<b>1%</b>	<b>3,99999981837</b>	<b>8859,20</b>
75	Torneio	3%	3,99758042127	2489,80
150	Torneio	3%	3,99968033109	4877,70
300	Torneio	3%	3,99999972538	8639,10
75	Torneio	7%	3,99491218854	2234,20
150	Torneio	7%	3,99792431836	4442,30
300	Torneio	7%	3,99998456884	8506,90

O tempo de convergência e a média da solução otimizada obtida pelo AG2 utilizando os métodos de seleção da roleta e do torneio foram semelhantes aos resultados obtidos pelo AG1.

A Tabela III mostra os valores obtidos pelo AG3 como solução após a média de 10 execuções utilizando como operadores de seleção o método da roleta e o método do torneio.

Tabela III – Média da solução otimizada obtida após 10 execuções do AG3 e seu respectivo tempo computacional.

iterações	Operador de seleção	Taxa de mutação	Solução otimizada	Tempo de convergência (ms)
75	Roleta	1%	3,99998766774	8047,20
150	Roleta	1%	3,99999947434	16077,40
<b>300</b>	<b>Roleta</b>	<b>1%</b>	<b>3,99999999697</b>	<b>32169,40</b>
75	Roleta	3%	3,99998329417	8140,50
150	Roleta	3%	3,99999495315	17433,30
300	Roleta	3%	3,99999602815	34147,70
75	Roleta	7%	3,99962636507	7950,20
150	Roleta	7%	3,99994110464	15759,90
300	Roleta	7%	3,99998746427	32002,50
75	Torneio	1%	3,99794644442	2262,40
<b>150</b>	<b>Torneio</b>	<b>1%</b>	<b>3,99995795404</b>	<b>4442,10</b>
300	Torneio	1%	3,99995731709	8888,00
75	Torneio	3%	3,98881672988	2344,00
150	Torneio	3%	3,99957438411	4483,60
300	Torneio	3%	3,99983972226	8703,80
75	Torneio	7%	3,98977565081	2274,30
150	Torneio	7%	3,99622941793	4459,00
300	Torneio	7%	3,99929207067	9509,30

Os resultados obtidos utilizando o *crossover* uniforme como operador de *crossover* foi semelhante aos resultados obtidos com o *crossover* de um ponto e *crossover* de dois pontos. O tempo de convergência novamente apresentou linearidade em relação ao número de iterações.

### A. Eficácia

A comparação da eficácia dessa simulação do algoritmo para o método exato é calculada como mostrado na Equação (4), lembrando que o máximo da função está localizado nas coordenadas (0,0). A eficácia dos melhores resultados dessa simulação para cada método seleção do AG1, AG2 e AG3 é ilustrado na Tabela IV.

**Tabela IV** – Eficácia dos melhores resultados obtidos pelo AG variando o método de seleção após 10 execuções.

iterações	AG	Operador de seleção	Taxa de mutação	Eficácia (%)
300	AG1	Roleta	1%	99,99999998475
300	AG1	Torneio	1%	99,99998598650
300	AG2	Roleta	1%	99,99999999700
300	AG2	Torneio	1%	99,99999545925
300	AG3	Roleta	1%	99,9999992425
150	AG3	Torneio	1%	99,99894885100

### B. Tempo Computacional

O comparativo do tempo computacional de execução dos algoritmos AG1, AG2 e AG3 em função do número de iterações e do operador de seleção é ilustrado na Figura 7.

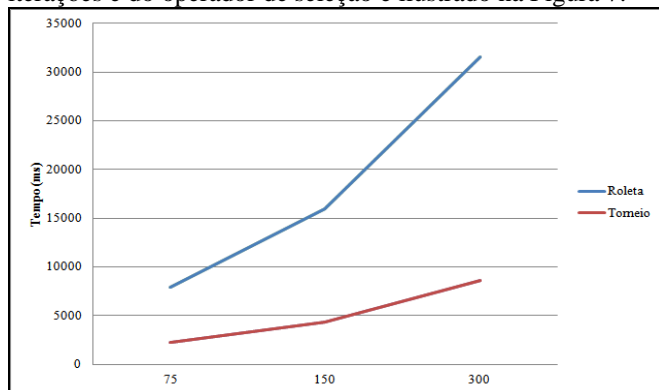


Fig. 7. Tempo de execução dos algoritmos em função do número de iterações e do operador de seleção.

## V. CONCLUSÃO

O sistema desenvolvido utilizando algoritmos genéticos obteve soluções ligeiramente melhores quando se utiliza o método da roleta como operador de seleção comparado ao método do torneio.

O tempo de convergência para a solução nos AGs implementados aumentou significativamente com o número de iterações. O método do torneio se mostrou mais eficiente

que o método da roleta, onde, o tempo de execução foi aproximadamente 3,5 vezes menor.

O aumento da eficácia do AG não está relacionado com o aumento do número de iterações e a variação da taxa de mutação influenciou de forma pequena no resultado.

Os resultados obtidos podem ser considerados satisfatórios com os três operadores de *crossover* utilizados, pois a média da eficácia nos resultados obtidos foram de 99,99% comparados ao valor exato.

Como trabalhos futuros sugere-se otimizar funções matemáticas de maior complexidade conhecidas na literatura e verificar se as soluções encontradas serão satisfatórias.

## REFERÊNCIAS

- [1] J.H. HOLLAND, *Adaptation in natural and artificial Systems*. Massachusetts: MIT Press, 1992.
- [2] R. LINDEN, *Algoritmos genéticos*. 2.ed. Rio de Janeiro: Brasport, 2008.
- [3] C. DARWIN, *A origem das Espécies e a Seleção Natural*. Brasil: Editora Madras, 2004.
- [4] D.E. GOLDBERG, *Genetic algorithms in search, optimization, and machine learning*. Massachusetts: Addison Wesley Publisher Company, 1989.
- [5] L.C.T. ALBUQUERQUE, *Alocação de canais em sistemas de comunicação celular empregando algoritmo genético distribuído*. Dissertação (Mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Ilha Solteira, 2009.