

## ALGORITMO GENÉTICO DE USO MÚLTIPLO

**Diego SOUZA (1); Carlos FONSECA (2)**

(1) CEFET-RN, Av. Sen. Salgado Filho, 1559, Tirol, Natal-RN – CEP: 59015-000, (84)88858027, e-mail: [diegouern@gmail.com](mailto:diegouern@gmail.com)

(2) UERN, e-mail: [carlosandre@uern.br](mailto:carlosandre@uern.br)

### RESUMO

Técnicas de otimização têm sido bastante utilizadas na em diversos setores, como indústria e serviços. Pode-se citar a aplicação na indústria de petróleo, como no problema de gerenciamento dinâmico de sondas de produção terrestre, onde sondas terceirizadas realizam serviços de manutenção em poços. Como a quantidade de poços é muito superior a de sondas, deve-se decidir, a cada instante, para onde enviar as sondas disponíveis. Essa decisão envolve vários parâmetros como: tempo de intervenção, distância a ser percorrida pela sonda e a capacidade de produção do poço que precisa de intervenção. Busca-se então otimizar a utilização das sondas.

Neste trabalho foi implementado um Algoritmo Genético, que é um método de busca dinâmica, baseado nos mecanismos de seleção e evolução naturais, que tem o objetivo de encontrar uma solução ótima para um dado problema de otimização.

A aplicação foi desenvolvida de forma a servir como ferramenta para o ensino desta técnica de otimização e possibilitar a sua utilização em qualquer tarefa de otimização. A forma genérica e modular com que o programa foi implementado, em Java, permite ao usuário visualizar e configurar todos os parâmetros de um Algoritmo Genético, facilitando o aprendizado desta ferramenta de otimização. Consegue-se facilmente modificar a tarefa a ser realizada, alterando-se a Função de Avaliação que é o modelo matemático do problema a ser otimizado. Ao exibir gráficos esta ferramenta auxilia na tomada decisões.

Concluindo, em todas as áreas existem problemas que demandam otimização. Desde que se obtenha um modelo matemático do problema de minimização de custos ou de maximização de lucros, o Algoritmo Genético será capaz de varrer o espaço de busca, delimitado pelo conjunto de valores possíveis para as variáveis envolvidas, em busca de uma solução ótima.

**Palavras-chave:** algoritmos genéticos, inteligência artificial, otimização, programação Java

## **1. INTRODUÇÃO**

A Inteligência Artificial é uma das mais recentes ciências. O seu desenvolvimento se iniciou após a Segunda Guerra Mundial e sua própria denominação foi dada em 1956. A IA abrange vários campos como aprendizado e percepção. Ela sistematiza e automatiza tarefas intelectuais e por isso é importante para qualquer esfera da atividade intelectual humana.

Este trabalho tem o objetivo de mostrar como a IA pode ser útil na resolução de um problema prático. Através do uso de um Algoritmo Genético (AG), um tipo de algoritmo usado na IA Evolucionista, pôde-se encontrar a solução para um problema proposto.

Além de ilustrar de forma objetiva o funcionamento de um AG, o presente trabalho tem o importante papel de exercitar a prática nesta importante área, que é a IA. Uma deficiência evidente no sistema de ensino brasileiro atual é a falta de integração da prática com a teoria. Por diversas vezes os conceitos não são corretamente aprendidos em sala de aula e o objetivo final, que é a formação de um profissional consciente e capaz, não é alcançado. Ao longo do desenvolvimento do AG vários pontos, antes nebulosos, foram sendo esclarecidos.

## **2. CONCEITOS FUNDAMENTAIS**

### **2.1. O que é Inteligência Artificial?**

A Inteligência Artificial (IA) é um ramo da Ciência da Computação dedicado ao estudo das técnicas que possibilitam a representação em máquinas de algum aspecto da cognição humana, como exemplos tem-se: a inteligência, a intuição e a perseverança. Na psicologia a inteligência é caracterizada pela manifestação de uma ou mais das seguintes qualidades: aprendizagem, adaptação e capacidade de resolver problemas. A pesquisa em IA é voltada para a construção de uma máquina que melhore a compreensão sobre a inteligência.

IA é o estudo de como fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor (RICHT e KNIGHT, 1994).

Um dos principais objetivos da IA é encontrar modelos computacionais para os processos humanos inteligentes (KELLER, 1991).

Com base nessas definições, pode-se concluir que IA é o campo do desenvolvimento de programas de computadores que simulam a cognição humana. Os pesquisadores em IA concordam com duas questões distintas a serem consideradas: a questão científica, que lida com aspectos teóricos da cognição, ou seja, estuda os processos de raciocínio humano para compreender o mecanismo da inteligência, e a questão tecnológica, que se preocupa com a representação destes processos através da máquina.

### **2.2. Surgimento da IA**

A IA surgiu no verão de 1956, em Hanover, New Hampshire, USA, quando um grupo formado por Jonh MacCarthy, Marvin Minsky, Claude Shanon e Nathaniel Rochester realizou a primeira conferência sobre IA, no Dartmouth College, patrocinada pela Fundação Rockefeller. McCarthy criou o termo Artificial Intelligence, que é Inteligência Artificial em inglês.

### **2.3. Abordagens da IA**

A IA divide-se em três abordagens diferentes a Simbólica, a Conexcionista e a Evolucionária. A IA Simbólica está relacionada com a psicologia e tem os Sistemas Especialistas como seu principal representante. A IA Conexcionista tem origem na neurofisiologia e tem as Redes Neurais Artificiais como destaque. E a IA Evolucionista é baseada na Teoria da Evolução de Darwin e utiliza-se de Algoritmos Genéticos para resolução de problemas de otimização.

## **2.4. IA Evolucionista**

É baseada na Teoria da Evolução (TE), originada dos trabalhos do naturalista inglês Charles Robert Darwin (1809-1882), que estabeleceram os mecanismos segundo os quais os animais evoluem a partir de formas mais simples, como resultado da necessidade de melhor adaptação ao seu meio ambiente. No decorrer desse processo vai ocorrendo uma seleção natural – a sobrevivência dos mais aptos – de acordo com as Leis de Seleção Natural (LSN) formuladas por Gregor Mendel (1822-1884).

## **3. ALGORITMOS GENÉTICOS (AGs)**

Os AGs são métodos de busca de resultados baseados nos critérios de seleção e evolução da natureza, TE e LSN, objetivando encontrar a melhor solução para um problema proposto. Essa solução é chamada de “indivíduo ótimo”. Esse “indivíduo ótimo” será avaliado através das gerações do algoritmo que se refinam ao longo de sua evolução. As gerações são como as da natureza: as espécies copulam e trocam genes que são passados para os seus descendentes. Os genes estão presentes nos cromossomos, e cada gene é responsável por uma ou mais características do indivíduo. No algoritmo um indivíduo é representado por um cromossomo, que também possui genes, representados no computador por número dentro de uma faixa determinada, por exemplo entre zero e dez. O tamanho do cromossomo, quantidade de genes que o cromossomo possui, é escolhido de acordo com o problema a ser resolvido.

Pode-se perceber claramente o quão flexível é este tipo de algoritmo: ele pode facilmente representar as características de uma determinada população, por exemplo os indivíduos de uma casa ou bairro, e computar a interação gênica ao longo das gerações, ou seja, a reprodução destes indivíduos: seus filhos, netos e bisnetos. Após a execução o AG retornaria a configuração genética da população na mesma localidade alguns anos mais tarde. É um tipo de computação poderosa e versátil, já que qualquer conjunto de características pode ser representado por um cromossomo, como dados de um motor de carro por exemplo. Então o AG, nesse caso do motor, combinaria diversos tipos de ajuste, cada ajuste representado por um cromossomo, nesse motor a fim de que, ao longo das gerações, que seriam as combinações entre os diversos tipos de ajustes, encontre-se o melhor tipo de ajuste, indivíduo ótimo, que seria o que oferecesse o melhor desempenho, durabilidade e economia.

O processo de melhoria, ou refinamento da população se dá através de critérios probabilísticos de seleção e reprodução naturais. A quantidade de gerações varia de acordo com o problema a ser resolvido: no caso de um casal que quer saber as possibilidades gênicas de seus filhos bastaria uma geração. Quanto mais indivíduos e quanto mais gerações forem computadas maior será a probabilidade de encontrar-se um indivíduo que melhor atenda às expectativas. É como uma empresa que procura um funcionário para preencher uma vaga. Quanto mais pessoas forem convidadas para uma entrevista maior a chance de achar alguém bom, um “indivíduo ótimo”, e esta chance aumenta com a quantidade de dias fazendo as entrevistas.

### **3.1. Conceitos em AGs**

Denomina-se geração a ordem cronológica de cada população surgida ao longo da execução computacional do algoritmo. Desta maneira, um mesmo indivíduo ao sobreviver computacionalmente a uma determinada geração, passa a pertencer à geração seguinte, diferentemente da interpretação biológica, em que cada indivíduo pertence à uma única geração. Computacionalmente os AGs possibilitam a coexistência de várias gerações em um mesmo intervalo de tempo. Devido à estanqueidade temporal existente entre as gerações, isto é, quando surge uma nova geração e um indivíduo da geração anterior continua vivendo, é como se ele “morresse” e uma cópia sua “nascesse” na nova geração, ou seja, na prática não é o mesmo indivíduo, e sim uma cópia sua, um clone.

#### **3.1.1. Representação Cromossômica**

As técnicas de Computação Evolucionária trabalham com uma população de pontos ao invés de um único ponto (indivíduo) durante o processo de busca, assim como os AG's. Cada indivíduo representa uma solução

potencial à um problema de otimização. O indivíduo é representado por um cromossomo, em metáfora com a biologia. A representação de um indivíduo por um cromossomo pode ser observada na Figura 1. Como pode-se observar na Figura 1, no primeiro cromossomo, há diversos “zeros” e “uns”, cada qual em um quadrado. Cada quadrado desses é um gene e representa uma característica. O “zero” pode representar a ausência de algum atributo de um indivíduo, como por exemplo a falta de cabelo, e o “um” representa a presença desse atributo. Quanto mais posições um cromossomo possuir mais atributos ele pode representar.

Cada possibilidade dentro de um gene é chamada de Alelo. Em um cromossomo binário os possíveis alelos são 0 e 1, em um decimal vão de 0 a 9 e em um hexadecimal de 0 a F.

Cromossomo de 16 posições com gene binário (0 ou 1).

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Cromossomo de 16 posições com gene decimal (0 a 9).

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 6 | 2 | 1 | 3 | 4 | 0 | 4 | 6 | 7 | 3 | 5 | 9 | 7 | 6 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Cromossomo de 8 posições com gene binário (0 ou 1).

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

Cromossomo de 8 posições com gene hexadecimal (0 a F).

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| F | A | 0 | 0 | 8 | 7 | 1 | B |
|---|---|---|---|---|---|---|---|

**Figura 1 – Representação de Diversos Cromossomos**

A escolha da representação cromossômica é crucial para o bom desempenho e eficiência do AG. Tome-se como exemplo uma representação cromossômica de 16 genes binários. Com essa representação pode-se representar 65535 possibilidades. Observe-se a equação abaixo [Eq. 01], onde “A” é referente ao alelo e “N” é referente à notação utilizada, no caso a binária, ou seja, N deve ser substituído por 2. O “A” abaixo está com o valor 1 para ilustrar o valor máximo que esse cromossomo pode representar.

#### *Valor do Indivíduo*

$$= A * N^{15} + A * N^{14} + A * N^{13} + A * N^{12} + A * N^{11} + A * N^{10} + A * N^9 + A * N^8 + A * N^7 + A * N^6 + A * N^5 + A * N^4 + A * N^3 + A * N^2 + A * N^1 + A * N^0$$

[Eq. 01]

Caso essa precisão não seja suficiente para a resolução do problema, deve-se adotar um cromossomo maior ou uma representação mais detalhada, como a binária ou hexadecimal. O comprimento do cromossomo pode ser obtido através da seguinte equação [Eq. 02], onde o Fim e o Início são os limites superior e inferior da função onde o AG está sendo aplicado e a Precisão é o grau de detalhamento que se deseja do AG.

$$Comprimento = 1 + inteiro \left( \ln \left( 1 + \frac{Fim - Início}{Precisão} \right) \right) \quad [Eq. 02].$$

Para se calcular o valor decimal (VD) de um cromossomo, que é o seu valor corrigido pelos limites inferior e superior da função, utiliza-se a equação abaixo [Eq. 03].

$$Valor\ Decimal = Início - (Fim - Início) * \frac{VDI}{VDS} \quad [Eq. 03].$$

### 3.1.2. Inicialização da População

A primeira população geralmente nasce de maneira aleatória (randômica). Isso significa que se o cromossomo segue a notação decimal, alelos entre 0 e 9 serão gerados para preencher seus genes. Esta geração randômica é bastante facilitada quando se utilizam os geradores randômicos normalmente disponíveis em linguagens de programação científicas. Por este e outros motivos, a linguagem Java foi escolhida para o desenvolvimento do AG a ser apresentado como solução ao problema proposto.

### 3.1.3. Avaliação

Cada indivíduo ou cromossomo de uma determinada população será avaliado de acordo com uma função matemática que traduz o seu comportamento. Tal função é denominada Função de Avaliação (FAV).

### 3.1.4. Adaptabilidade

A adaptabilidade (AD) indica o quanto um determinado indivíduo está adaptado aos aspectos modelados matematicamente pela função de avaliação. Normalmente é feito igual ao valor de FAV quando FAV > 0.

Porém quando FAV < 0 obtêm-se AD somando um valor relativamente pequeno (arbitrário) a todos os valores de FAV para que AD seja sempre > 0.

### 3.1.5. Número de Cópias Esperadas (NCE)

Fornece a quantidade de cópias esperadas de um determinado indivíduo na próxima geração [Eq. 04].

$$NCE = \frac{AD}{Média\ dos\ ADs} \quad [Eq. 04]$$

### 3.1.6. Adaptabilidade Relativa (ADR)

Indica a adaptação de um indivíduo em relação aos demais indivíduos da população [Eq. 05]. A soma das ADR's de uma população deve ser igual a 1.

$$ADR = \frac{AD}{\sum AD} \quad [\text{Eq. 05}]$$

### 3.1.7. Seleção

Selecionar significa escolher quais indivíduos darão origem à nova população ou nova geração. Os indivíduos selecionados são chamados de pais. Existem três tipos de seleção: Determinística, Estocástica e Híbrida.

Na seleção determinística o indivíduo tem 0 ou 100% de chances de ser escolhido. Na seleção estocástica os indivíduos que com maior AD possuem maiores chances de serem escolhidos. A seleção híbrida é uma combinação das anteriores em etapas diferentes do processo de seleção. No AG apresentado utiliza-se a seleção estocástica por ser mais dinâmica.

#### 3.1.7.1. Seleção Estocástica

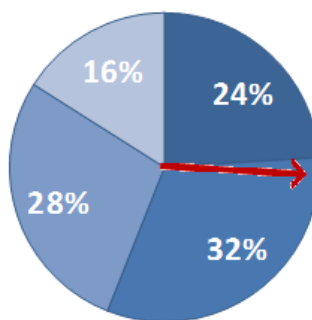
Por ser um processo probabilístico é mais dinâmico que o determinístico. Em cada etapa do processo, os indivíduos que apresentarem maior adaptabilidade terão mais chance de serem escolhidos. É importante não confundir processo estocástico com processo aleatório. No estocástico há probabilidades de o indivíduo ser selecionados diferentes das probabilidades aleatórias.

##### 3.1.7.1.1. Método de Seleção da Roleta

É o método de seleção mais utilizado. É um processo probabilístico, e foi adotado para a fase de seleção do AG apresentado. O método da roleta utiliza as ADR's para escolher os indivíduos. Uma tabela como a Tabela 1 é montada e então a ponteiro é girado, parando em um determinado indivíduo que é então selecionado. Pode-se observar que quanto maior a ADR maior a chance de o ponteiro parar no indivíduo pois sua parte na roleta torna-se maior em função da ADR.

**Tabela 1 – Exemplo de Seleção por Método da Roleta**

| Indivíduos | ADR  | +ADR |
|------------|------|------|
| 1          | 0,24 | 0,24 |
| 2          | 0,32 | 0,56 |
| 3          | 0,28 | 0,84 |
| 4          | 0,16 | 1,00 |



### 3.1.8. Reprodução

A reprodução é a fase do AG em que os cromossomos-filhos são gerados para preencherem a nova geração. Essa nova geração é formada pela combinação de cromossomos-pais ou a partir de um indivíduo-pai único. Os métodos utilizados para a reprodução são a reprodução, a mutação, a inversão e a rotação.

### 3.1.8.1. Recombinação (Crossover)

Consiste na troca de material genético entre dois cromossomos gerando um único filho. São escolhidas duas posições nos cromossomos-pais e entre elas, ou fora delas, ocorrerá a troca de genes.

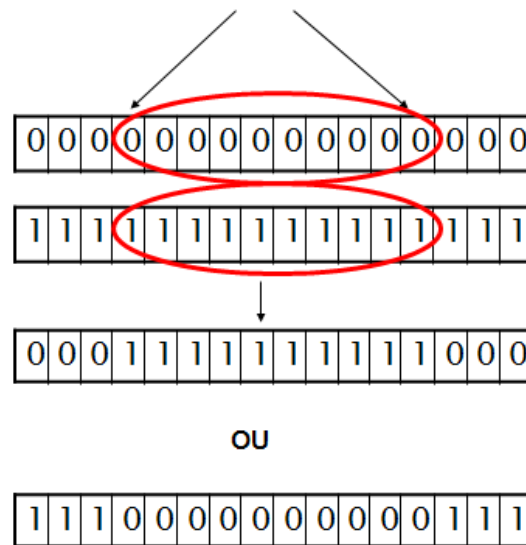


Figura 2133131 – Exemplo de Reprodução por Recombinação

### 3.1.8.2. Mutação

Consiste na mudança de um único gene para outro alelo possível. Neste tipo de reprodução ocorre a participação de um único cromossomo-pai.

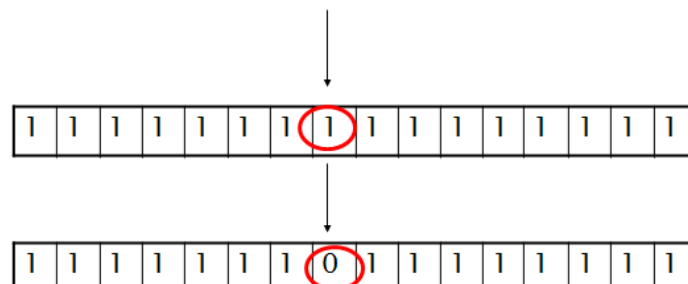


Figura 2133131 Exemplo de Reprodução por Mutação

### 3.1.8.3. Inversão

Os genes entre duas posições do cromossomo-pai são invertidos como se fosse colocados em frente a um espelho.

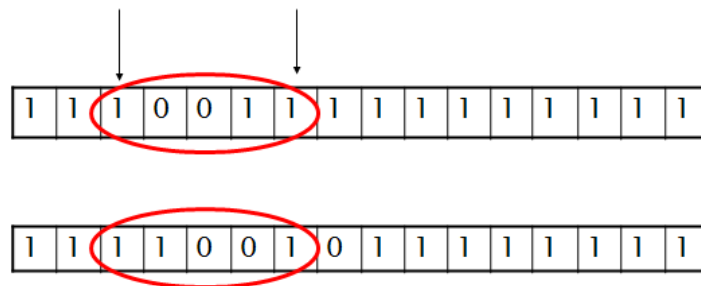


Figura 2133131 – Exemplo de Reprodução por Inversão

#### 3.1.8.4. Rotação

Os genes entre duas posições dos cromossomos-pais são deslocados para a direita. O gene que “passar” da área selecionada é então colocado no início da área selecionada, mais a esquerda.

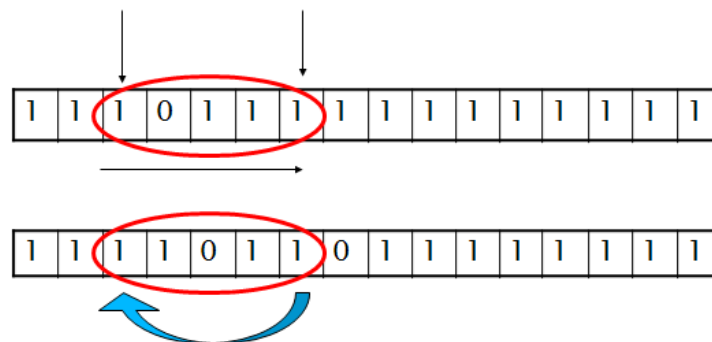


Figura 2133131 – Exemplo de Reprodução por Rotação

#### 3.1.9. Condições de Parada

Por ser um processo iterativo o AG precisa ter suas condições de parada previamente estabelecidas. Podem ser baseadas na quantidade de gerações, quantidade de vezes que o melhor indivíduo se repetiu (estagnação do melhor indivíduo), ou por estagnação da população (quando a carga genética da população muda muito pouco de uma geração para outra).

#### 3.1.10. Funcionamento Computacional de um AG

O funcionamento computacional de um AG é relativamente simples. Inicialmente escolhe-se a representação cromossômica e uma população tem sua configuração genética gerada aleatoriamente. Essa população passa por uma fase de avaliação. Se uma condição de parada for atendida o programa termina, senão ocorre a fase de reprodução. Após a reprodução a população é novamente avaliada, até que uma condição de parada seja atendida. Essa sequência é ilustrada na Figura 2.



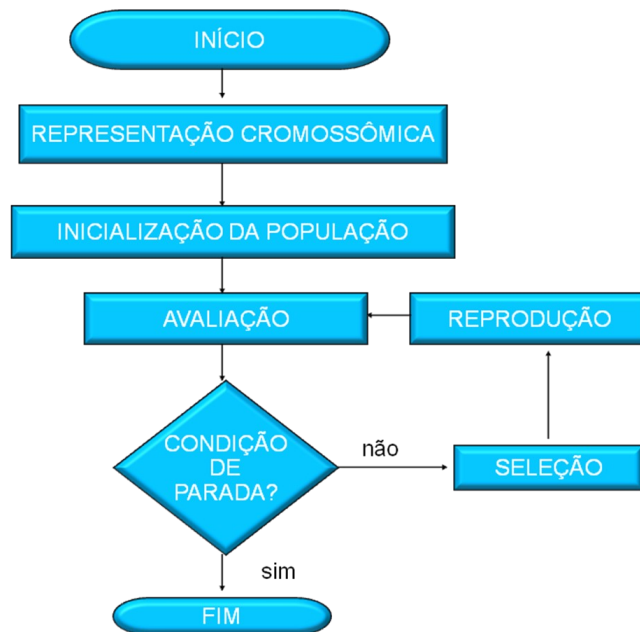


Figura 2 – Fluxograma de Funcionamento de um Algoritmo Genético

#### 4. PROBLEMA PROPOSTO

Desenvolver um Algoritmo Genético que encontre o mínimo da função a seguir [Eq. 01]:

$$f(x) = |(-x^4 + 5x^3 - 17x^2 + x + 4000) * \sin(1,7x - 200) * e^{(x/250)} + 5997| \quad [\text{Eq. 01}]$$

Tabela 2 – Legenda dos Símbolos Utilizados Na Equação

| Símbolo | Significado                         |
|---------|-------------------------------------|
| $x$     | Variável da Função                  |
| $ $     | Módulo                              |
| $e$     | Número de Euler = 2.718281828459045 |
| $\sin$  | Seno                                |

$$f := x \rightarrow \left| (-x^4 + 5x^3 - 17x^2 + x + 4000) \sin(1.7x - 200) e^{\left(\frac{1}{250}x\right)} + 5997 \right|$$

Figura 3 – Representação Gráfica da Equação do Problema Proposto

#### 4.1. Dados do Problema

**Tabela 3 – Configuração do AG Solução**

| <i>Característica</i>        | <i>Configuração</i>  |
|------------------------------|--|
| <i>Sistema de Numeração</i>  | Binário (0 ou 1)   |
| <i>Tamanho do Cromossomo</i> | 16 Posições (Genes)  |
| <i>Limites</i>               | De -12 a 12  |
| <i>Método de Seleção</i>     | Roleta   |
| <i>Crítérios de Parada</i>   | Estagnação do Melhor Indivíduo e Atingir o Número Máximo de Gerações |

O sistema de numeração a ser utilizado é o binário (0 ou 1), com limites de -12 a 12, utilizando cromossomos formados por 16 genes, o método de seleção a ser utilizado é o da roleta e o programa deverá obedecer a dois critérios de parada: estagnação do melhor indivíduo, atingir o número máximo de gerações. Um resumo das configurações que o AG deve possuir pode ser visualizado na Tabela 3, acima.

O algoritmo desenvolvido deve atender à estas especificações e também gerar gráficos comparativos para visualização do avanço do melhor indivíduo através das gerações.

#### 5. PROGRAMA SOLUÇÃO

O programa solução foi desenvolvido utilizando-se a linguagem de programação orientada a objetos Java. Com uma interface amigável e intuitiva este poderoso software pode ser completamente configurado para resolver diversos problemas de otimização. Uma captura da tela inicial de configuração do programa pode ser visualizada na Figura 4.

Todas as entradas do AG podem ser configuradas na área *Configurações de Entrada*, como número de indivíduos de uma população, quantidade de gerações, a quantidade de vezes que o melhor indivíduo se repete, o método de seleção, os critérios de parada, o tamanho do cromossomo, o sistema de numeração do cromossomo, os limites da saída, a precisão desejada na saída do programa e a porcentagem que cada um dos tipos de reprodução ocupará na roleta (recombinação, inversão, rotação e mutação), são configuráveis através da interface gráfica com o usuário.

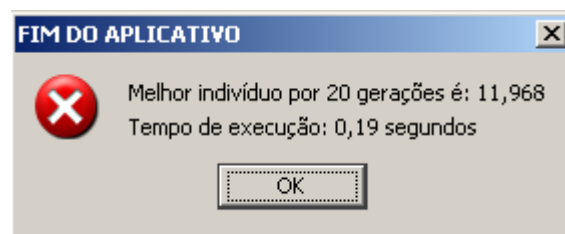
Além das configurações já citadas o programa possui, ainda, um gerador de exemplo de cromossomo, que fornecerá um indivíduo exemplo com as configurações selecionadas na interface. É uma função importante para o usuário verificar se realmente a configuração que ele realizou é a desejada.

Há também, na área *Configurações de Saída*, a opção de configuração dos detalhes na saída. O usuário pode escolher, um a um, os dados que deseja visualizar, como por exemplo o AD, o VD, a geração e o cromossomo. O usuário pode escolher, também, se deseja visualizar todos os indivíduos de uma geração, apenas 10, 5 ou 1. Técnicas de programação foram utilizadas para que a saída fosse dada o mais rápido possível. O usuário pode escolher opções de melhor desempenho para o programa, a fim de obter resultados mais rapidamente.

Após configurado como na Tabela 3, o programa iniciará sua execução seguindo o fluxograma da Figura 2. Sua execução será encerrada apenas quando um dos critérios de parada for atendido: atingir o número máximo de gerações ou a estagnação do melhor indivíduo. A mensagem de resposta do AG é mostrada na Figura 5. A mensagem de resposta varia de acordo com o porquê de o AG ter parado e com o tempo de execução. Dependendo das configurações escolhidas o tempo pode variar bastante.



**Figura 4 –Captura de Tela da Janela Inicial com Indicações de Configuração do AG Desenvolvido**



**Figura 5 –Captura da Tela de Resposta do AG Desenvolvido**

## 6. CONCLUSÃO

Como pode-se observar pela Figura 5 o AG encontrou sua resposta em uma fração de segundos inferior a um quinto de segundo. Foi uma resposta rápida e adequada demonstrando como esse tipo de técnica da IA é poderosa. Este problema específico foi resolvido rapidamente por ser apenas uma demonstração do funcionamento de um AG, porém problemas maiores podem demandar mais de uma execução com configurações mais extremas que podem levar minutos ou horas para terem uma resposta do AG.

## REFERÊNCIAS

- RICHT, E; KNIGHT, K. **Inteligência Artificial**. 1. ed. Makron, 1994.
- KELLER, R. **Tecnologia de Sistemas Especialistas**. 1. ed. Makron, 1991.
- RUSSEL, S; NORVIG, P. **Inteligência Artificial**. 2. ed. Elsevier, 2004.
- FILHO, O.G. **Tópicos Avançados de Inteligência Artificial**. 1. ed. UnP. 1998.
- HAYKIN, S. **Redes Neurais**. 2. ed. São Paulo: Bookman, 2001.
- BRAGA, A.P.; CARVALHO, A.C.; LUDERMIR, T.B. **Redes Neurais Artificiais**. 1. ed. Rio de Janeiro: LTC, 2000.

## AGRADECIMENTOS

Deixamos nosso agradecimento às seguintes instituições e pessoas, sem as quais o presente poderia não ter sido completado:

- Universidade do Estado do Rio Grande do Norte, que disponibilizou estrutura física para o desenvolvimento deste trabalho.
- Centro Federal de Educação Tecnológica do Rio Grande do Norte, que proveu recursos como máquinas modernas para os cálculos e Internet para as pesquisas.
- Ao professor Francisco de Assis, que ajudou bastante no apoio matemático ao desenvolvimento do aplicativo.