

# ALGORITMO GENÉTICO COMO SOLUÇÃO PARA A MOVIMENTAÇÃO DE UM BRAÇO ROBÓTICO SIMULADO

Gustavo Henrique Neves Siloto - [gustavosilot@gmail.com](mailto:gustavosilot@gmail.com)

Felipe Alexandre Cardoso Pazinato - [felipe\\_pazinatto@yahoo.com.br](mailto:felipe_pazinatto@yahoo.com.br)

**RESUMO:** Com a evolução tecnológica novos desafios naturalmente surgem, e soluções para tais desafios, muitas vezes, se tornam difíceis ou impraticáveis para o ser humano. A inteligência artificial e suas técnicas como computação evolutiva e algoritmo genético, vem como opção para solucionar tais desafios de maneira prática e mais eficaz. Este trabalho propõe-se a encontrar a solução para o movimento um braço robótico com três graus de liberdade, utilizando algoritmos genéticos com abordagem elitista. O braço robótico simulado foi construído em OpenGL. Quanto ao algoritmo genético, para os parâmetros seed(10), população(100), número de gerações(0), foram encontrados os melhores resultados, fazendo com que o movimento do braço robótico atinja seu objetivo em apenas uma geração.

**PALAVRAS-CHAVE:** Inteligência Artificial, Computação Evolucionária, Algoritmo Genético.

**ABSTRACT:** With the technological evolution new challenges naturally arises, solutions for such challenges becomes difficult or impractical for the human being. Artificial Intelligence and its techniques such as evolutionary computing and genetic algorithm, comes to solve such challenges with efficiency and practicality. This paper propose to find a method for a robotic arm movement with three degrees of freedom, using genetic algorithms with an elitist approach. The simulated robotic arm was developed using OpenGL. As for the genetic algorithm, for the parameters in seed(10), population(100), number of generations(0), has been found the best results, making the movement of the robotic arm reach the objective with only an one generation.

**KEYWORDS:** Artificial Intelligence, Evolutionary Computing, Genetic Algorithm.

## 1. Introdução

Charles Darwin demonstra em seu livro "On the Origin of Species by Means of Natural Selection"(1859), características interessantes sobre a evolução e a adaptação encontrada em seres orgânicos na natureza. Características de animais obtidas ao longo do tempo, são reflexo direto da maneira no qual conseguiram se adaptar e evoluir, sendo os principais catalisadores destas características evolutivas o ambiente no qual vivem e seus competidores e predadores.

Outro fato sobre tais características: um único organismo diploide, dificilmente irá evoluir sem que ocorra uma troca de genomas, e possíveis mutações.

Com a evolução tecnológica, a ideia de desenvolver sistemas capazes de automatismos não faz mais parte da criatividade de autores de contos épicos, romances ou interesse de cientistas, mas com a era atual tornou-se uma necessidade para o ser humano engendrar sistemas autômatos capazes de gerenciar e executar vastas tarefas repetitivas, longas e complexas. Esta área foi nomeada de *Inteligência Artificial (IA)*.

O campo de IA, sendo uma vasta área de pesquisa, gerou diversos braços, muitos obtendo influência da matemática ou cognição humana, mas também outros baseado em conceitos encontrados na natureza. Foi desenvolvido uma área da IA para a resolução de problemas de maneira heurística chamada de Computação Evolucionária(CE). A primeira técnica de CE(Rechenberg,1973), chamada de Estratégias Evolutivas, no qual utiliza dois indivíduos para a resolução do problema. Fogel, Owens e Walsh(1966) desenvolveram o conceito de *Programação Genética*, no qual cada indivíduo iria gerar uma máquina de estados finitos para a solução específica do problema. Holland (1992) em seu livro "*Adaptation of Natural and Artificial Systems*" faz um estudo sobre a adaptação, e define características principais sobre o *Algoritmo Genético(AG)*, no qual se baseia em aspectos encontrados na natureza, sobrevivência do mais apto, genética, definidos para uma população de tamanho N, sendo N a quantidade total de indivíduos.

Trabalhos anteriores com a movimentação de um braço mecânico simulado utilizando AG, foram estudados, mas a maior parte utiliza conceitos matemáticos(cálculo de cinemática inversa) para a execução dos AG e os aplica em ambientes simulados normalmente compostos por duas dimensões(Junior, 2013). O foco deste trabalho é desenvolver e mostrar a capacidade do AG fora de conceitos matemáticos, utilizando valores aleatórios para gerar a movimentação de um braço mecânico simulado de um ponto fixo A até o ponto fixo B.

## 2. Algoritmo genético

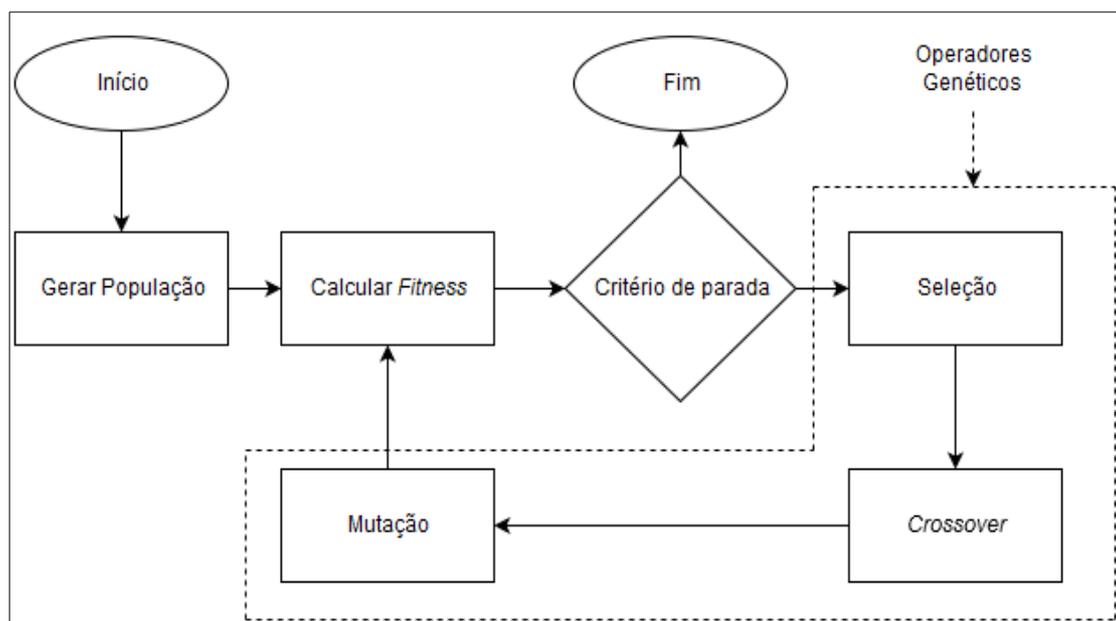


Figura 1: AG

De acordo com Mitchell(1999), qualquer AG possui a composiç o semelhante a da figura 1 acima, onde inicialmente   gerado uma populaç o de indiv duos(genomas), seguido pela categorizaç o de cada genoma baseado na qualidade da resoluç o dos problemas, este valor de categorizaç o   chamado de fitness. O algoritmo segue por um teste para verificar se um crit rio de parada foi encontrado, caso n o seja encontrado, o AG faz realizaç o dos operadores gen ticos, seleç o, crossover(cruzamento) e mutaç o, retornando para a categorizaç o de cada genoma, e iniciando uma nova geraç o.

*Seed*   um valor utilizado na equaç o de geraç o de n meros pseudo-aleat rios(*RNG*),   uma vari vel no qual pode alterar drasticamente a maneira

em que as operação do AG, irão retornar algum resultado.

000110110100101101001011101011000101010001010

Figura 2: Exemplo de um Genoma

Todas os seres orgânicos encontrados na natureza possuem *ADN*, uma espécie de banco de memória, no qual é armazenado as características, como a cor dos olhos a cor da pele, etc. Em *AG*, o genoma é definido por um vetor de *string* contendo apenas os caracteres 0 ou 1, valores numéricos do tipo binário para ser mais exato. A figura 2 acima mostra um exemplo de um genoma a ser encontrado em *AG*.

Tabela 1: Tabela de codificação de *AG*

Tabela: Codificação de <i>AG</i>	
Código	Valor
000	1
001	2
010	3
100	4
011	5
110	∅
101	∅
111	∅
0	Movimento para Esquerda
1	Movimento para Direita

A codificação das características ocorre de maneira semelhante ao encontrado no *ADN*.

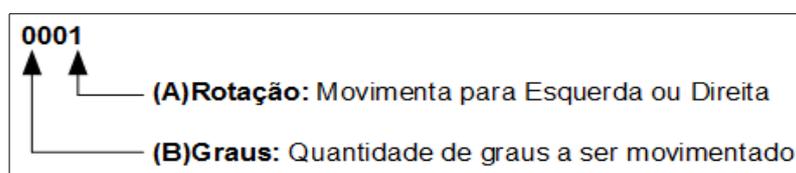


Figura 3: Codificação de um Genoma

Em AG as definições das características podem ser valores binários singulares ou um conjunto dos mesmos. As definições para cada conjunto podem suportar  $N$  tipos de dados, de texto a valores numéricos ou simbólicos, como mostrado na tabela 1. Na figura 3 acima, o genoma exemplo apresenta duas codificações; a primeira (A), o conjunto de três valores binários o qual pode representar 8 diferentes valores decimais definidos na tabela 1, e o segundo (B), representa a direção no qual um objeto pode ser rotacionado.

Uma população de *indivíduos(genoma)*, é uma quantidade de tamanho  $N$  a ser criado, cada genoma é geradores com valores aleatórios definidos pelo algoritmo *RNG*(utilizando uma *Seed*).

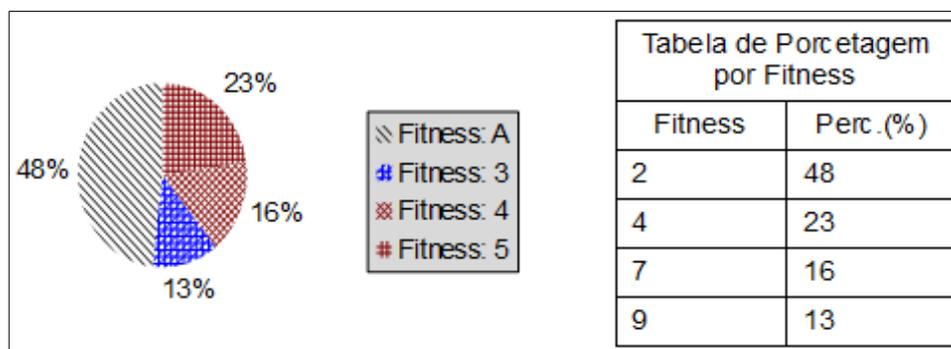


Figura 4: Exemplo de seleção por roleta

Na natureza, para que existam novas gerações de espécies, é necessário inicialmente gerações anteriores sobrevivam. Para verificar se um indivíduo irá "sobreviver", em AG, foi definido o conceito chamado de *fitness*, no qual é atribuído um valor para cada indivíduo de uma população. Este valor representa o ambiente, ou, o quão bem ele irá lidar com a tarefa designada. A grandeza deste valor, definido como valor de ajuste do quanto um genoma está adaptado ao seu ambiente, pode possuir significados variados dependendo da maneira em que o AG foi definido, sendo valores baixos de boa qualidade e altos de péssima, e vice-versa. Os valores gerados são adaptativos, pois são comparados com valores de genomas anteriores, e sua grandeza aumenta ou diminui baseado na existência atual do genomas(Mitchell, 1999).

Para o cruzamento ser realizado entre duas espécies, na natureza machos e fêmeas escolhem o melhor parceiro baseado em certas características, este evento é chamado de *seleção* em AG. A *seleção* pode ocorrer de diversas

maneiras dependendo da técnica utilizada. É possível fazer a seleção aleatoriamente entre dois *genomas* sem nenhum critério de qualidade, ou uma seleção utilizando *genomas* com *fitness* de alta qualidade, ou ainda utilizar a técnica chamada de roleta.

A técnica da roleta, atribui a todos os *fitness* um valor de porcentagem dentro de zero a cem por cento(0-100%), a grandeza deste valores estão baseados na qualidade do *fitness* de cada *genoma* dentro da população geral. Observando a *figura 4* acima, existe um total de quatro *fitness*, cada um possuindo uma porcentagem específica dentro de cem por cento, no qual representam quatro diferentes fatias de uma roleta. Depois de todos os *fitness* receberem a atribuição, a seleção ocorre da mesma maneira em que uma roleta real funcionaria.

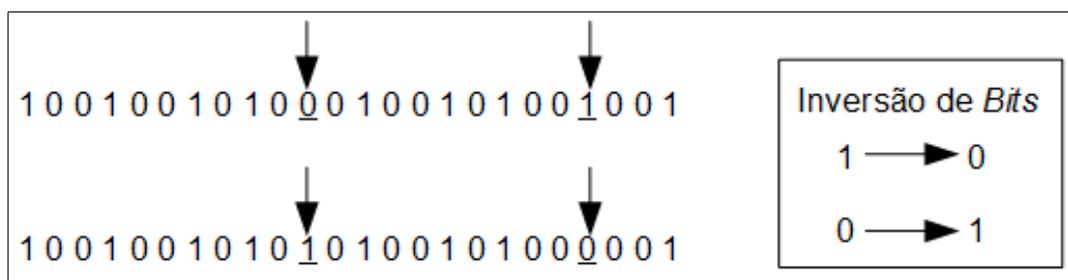


Figura 5: Exemplo de mutação sobre um genoma

O termo *crossover* pode ser encontrado em literaturas sobre biologia e genética, o qual significa a recombinação genética entre dois *genomas*. Em AG, ele possui o mesmo significado, porém definido de forma distinta. A *figura 5* acima, demonstra um exemplo de *crossover* sendo executado com dois *genomas*. Este processo ocorre após o estágio da seleção. Os *genomas* resultantes irão substituir outros dois *genomas* da população, podendo ser eles os próprios pais, os *genomas* com os piores *fitness*, ou mesmo ainda, *genomas* aleatórios encontrados na população. O ponto de cruzamento é o local onde será dividido os múltiplos *genomas* para a geração de outras duas resultantes. Alguns autores também mostram a utilização de múltiplos pontos fixos, ou ponto(s) aleatórios, para a troca de *genomas*.

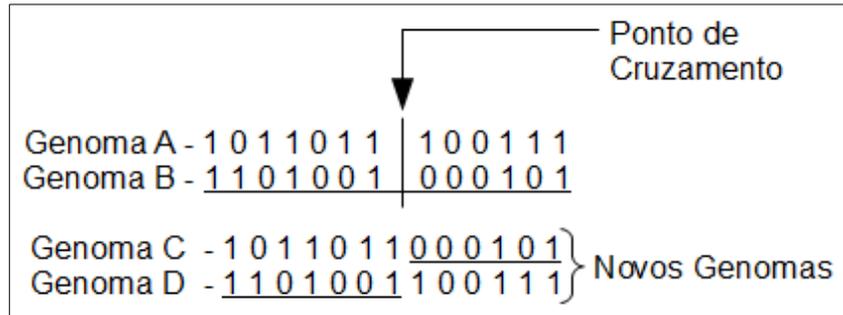


Figura 6: Execução do Crossover

A *mutação* é um evento que pode ocorrer ao final da execução do *crossover*. Certo ponto, ou múltiplos pontos, terão seus valores(*bits*) invertidos, caso o valor seja 0, irá se tornar 1 e vice-versa como é visto na *figura 6* acima. Esta alteração pode causar efeitos drásticos(bom ou ruim) sobre um *genoma*, alterando o seu *fitness*, ou podendo não afetar em nada.

### 3. Estrutura do trabalho

Este trabalho foi desenvolvido utilizando a linguagem C++, a *IDE Qt Creator* e as bibliotecas de computação gráfica e de geração de Interface GUI do mesmo. O braço em ambiente 3D foi desenvolvido em *OpenGL 4.0*(Shreiner et al., 2013) e a biblioteca *GLM*(OpenGL Mathematics) para calculo do *OpenGL*.

O braço robótico possui um total de três partes contendo um grau de liberdade cada, totalizando no braço, três graus de liberdade.

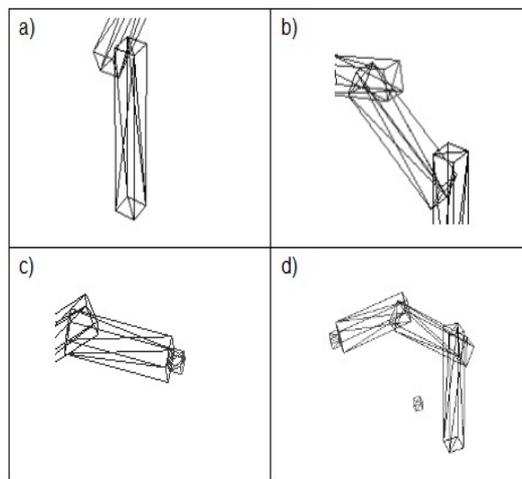


Figura 7: Composição do Braço Robótico - a) Base do braço. b) Braço médio. c) Braço Superior. d) Visualização integral do ambiente

Na *figura 7*, o quadro a), mostra a base do braço com possível rotação de *trezentos e sessenta graus* em si mesmo, o quadro B mostra a parte média do braço, tem uma rotação de sessenta graus a partir de sua base, com posição inicial de sessenta graus a partir de sua base no sentido horário. O quadro C o braço superior, tem uma rotação de *sessenta graus* a partir de sua base com uma posição inicial de *quarenta graus* no sentido horário, o braço inteiro pode ser visualizado no quadro D. Os pequenos cubos visto no quadro C (topo do braço), e no quadro D (próximo a base), representam sensores, sendo o cubo próximo a base o alvo a ser atingido, e o cubo ao topo o objeto a atingir o alvo.

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2} \quad (1)$$

Para avaliar o *genoma* foi utilizado como *fitness* a minimização do cálculo de *distância euclidiana*(1) visto acima, entre ambos os cubos, e o resultado é arredondado e aplicado ao genoma como *fitness*, não sendo levado em conta a tempo de rotação. Os valores não são adaptativos, sendo eles fixos (1 para melhor).

O genoma é composto por um total vinte e oito caracteres binários, possuindo 3 divisões principais no qual representam cada parte do braço, sendo a primeira para a base (quadro A), o segundo o braço médio (quadro B) e o terceiro o braço superior (quadro C). Dentro de cada divisão ocorre subdivisões, onde os primeiros conjuntos (2<sup>1</sup> | 2<sup>2</sup>) binários sempre irão representar a ação de rotação e os outros conjuntos representam os graus no qual devem ser rotacionadas.

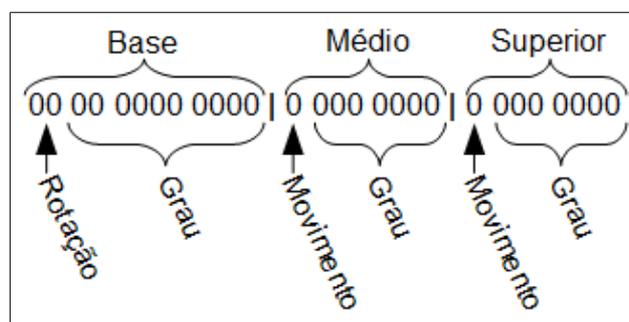


Figura 8: Composição do genoma no AG

Observando o primeiro conjunto da *figura 8* (base), e utilizando o seguinte exemplo : 00 01 0100 1101. Pode-se dividir a cadeia de caracteres 00 01 0100

1101 na forma a)[00] b)[01] c)[0100] d)[1101]. O valor a) representa a rotação(direção e movimento), como demonstrado na *tabela 2*.

*Tabela 2: Tabela de Rotação*

<b>Tabela de Rotação</b>	
<b>Código</b>	<b>Valor</b>
<b>00</b>	Direita
<b>01</b>	Parado
<b>10</b>	Parado
<b>11</b>	Esquerda

O valor em b) representa a parte da centena da grandeza do grau de rotação, representado na *tabela 3*.

*Tabela 3: Tabela de Grau - Centena*

<b>Tabela de Grau: Centena</b>	
<b>Código</b>	<b>Valor</b>
<b>00</b>	0
<b>01</b>	1
<b>10</b>	2
<b>11</b>	3

O valor c) representa a dezena e o valor d) representa a unidade, mostrado na *tabela 4*.

<b>Tabela de Grau : Unidade e (A)Dezena</b>			
<b>Código</b>	<b>Valor</b>	<b>Código</b>	<b>Valor</b>
<b>0000</b>	0	<b>1000</b>	4
<b>1000</b>	0	<b>1011</b>	4
<b>0001</b>	1	<b>0011</b>	5
<b>1001</b>	1	<b>1101</b>	5
<b>0010</b>	2	<b>0110</b>	6
<b>1010</b>	2	<b>1100</b>	7
<b>0100</b>	3	<b>0111</b>	8
<b>1110</b>	3	<b>1111</b>	9

*Tabela 4: Tabela de Grau: Unidade e (A) Dezena*

No caso do exemplo, observando as *tabelas 2 a 4*, o valor binário combinado de *abcd*(exemplo acima), representa a rotação para a direita em 135°. Caso a esteja parado, os valores em *a*) serão {01,10}, e nenhum grau gerado terá efeito.

*Tabela 5: Tabela de Movimento*

<b>Tabela de Movimento</b>	
<b>Código</b>	<b>Valor</b>
<b>0</b>	Parado
<b>1</b>	Movimento

*Tabela 6: Tabela de Grau - (B)Dezena*

<b>Tabela de Grau:(B)Dezena</b>	
<b>Código</b>	<b>Valor</b>
<b>000</b>	0
<b>001</b>	1
<b>101</b>	1
<b>010</b>	2
<b>100</b>	3
<b>011</b>	4
<b>110</b>	5
<b>111</b>	6

Para o braço médio e o braço superior, é utilizado apenas as *tabelas 5, 6 e 4* respectivamente, enquanto o primeiro conjunto utiliza as *tabelas 2, 3, 4 e 4* respectivamente. Em certos casos, é possível que a rotação do braço médio e superior e seus graus somados superem os limites definidos do braço integralmente, isso automaticamente forçar o *fitness* a receber um valor de tamanho máximo do tipo inteiro, ajustando o *AG* para que possa remove-los na geração posterior. Em outro caso específico, como um grau gerado maior que 360, ele continuara com a sua rotação sem nenhuma alteração do *fitness*.

Os parâmetros de *seed*, tamanho total de população e total de gerações a serem utilizados no *AG*, não são fixos, podendo ser alteradores na inicialização

do algoritmo. O método de seleção aplicado, foi o método da roleta, e o *crossover*, faz a substituição dos piores genomas ou últimos genomas encontrados na lista.

#### 4. Resultados e discussões

	Fitness	Genome	Base	Middle Arm	Upper Arm	Sensor X	Sensor Y	Sensor Z
1	1	0010110010101011010010011001	272	43	11	-3.52948	1.11937	0.123251
2	1	0011000010111011110111100010	304	45	52	-2.69929	1.0179	1.8207
3	1	0011000010111011010010011001	304	43	11	-2.92785	1.11937	1.97486
4	1	0010110010101011110111100010	272	45	52	-3.25395	1.0179	0.11363
5	1	0011000010111011010010011001	304	43	11	-2.92785	1.11937	1.97486
6	1	0011000010111011010010011001	304	43	11	-2.92785	1.11937	1.97486
7	1	0011000010111011010010011001	304	43	11	-2.92785	1.11937	1.97486

Figura 9: Janela do Software

Na figura 9 acima, é mostrado uma *screenshot* do software desenvolvido para teste do AG com configuração de parâmetros pelo usuário.

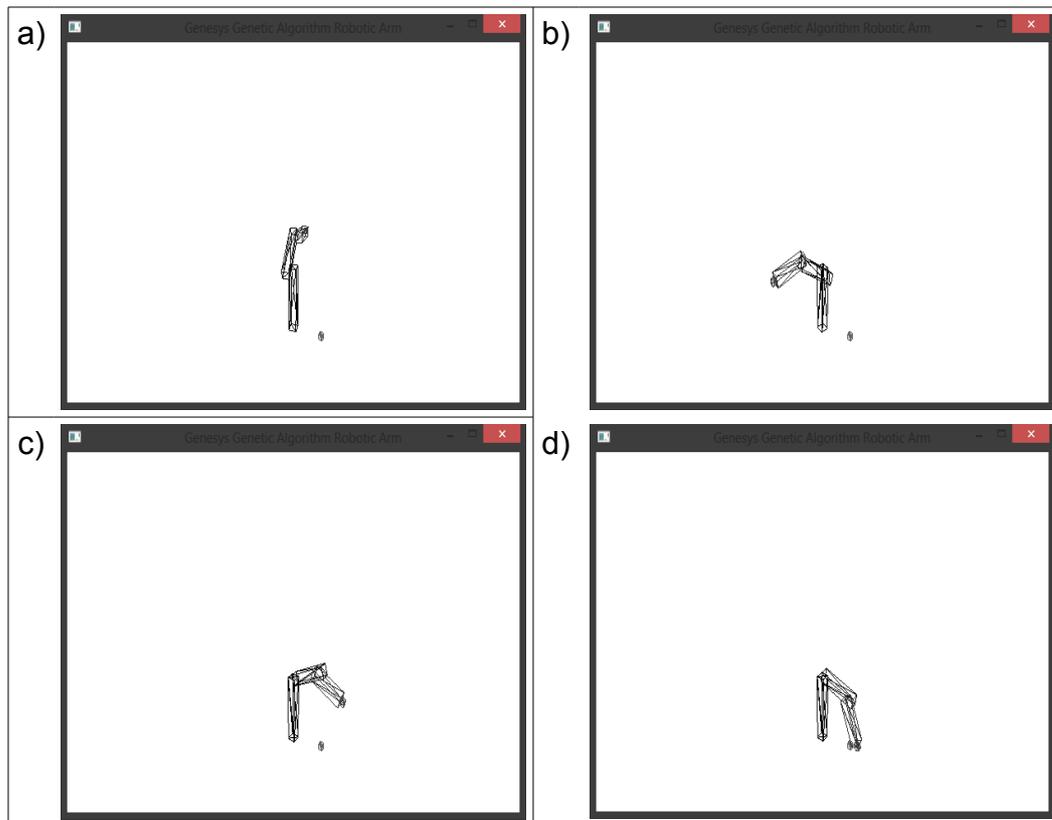


Figura 10: Exemplo de resultado de um genoma. a) Início: Posição Inicial do braço robótico. b) Movimentação em direção ao alvo. c) Sobre o alvo. d) Tocando o alvo.

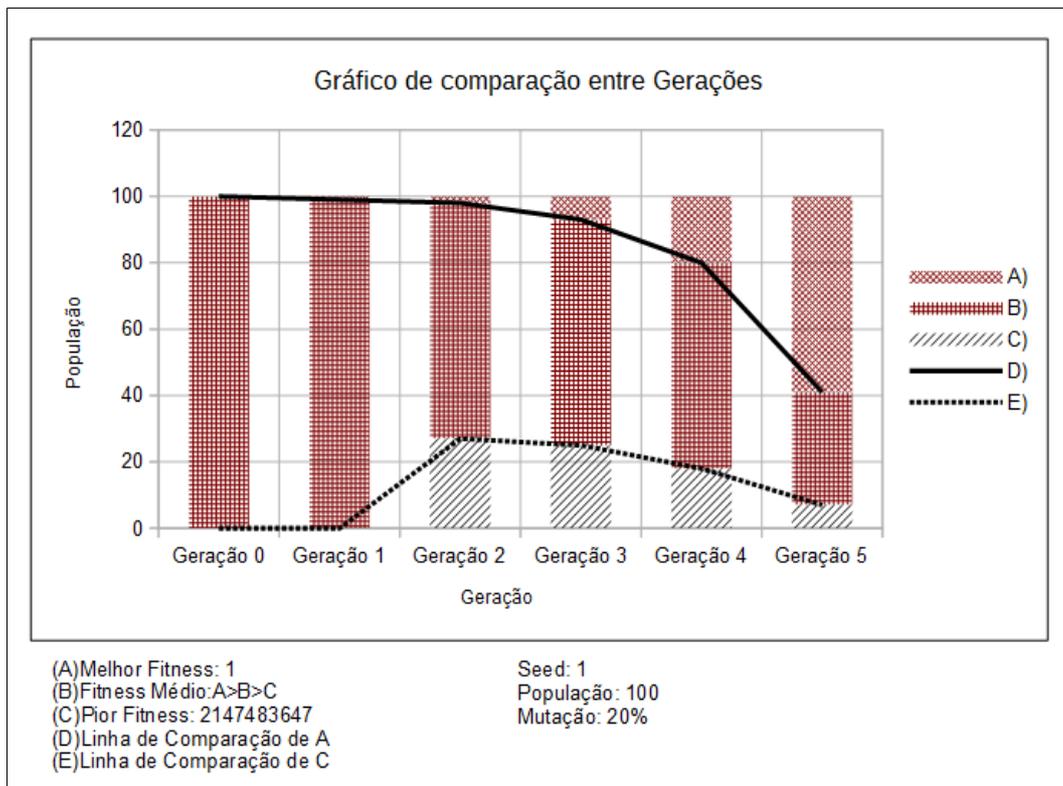


Gráfico 1: Gráfico de Comparação de fitness por População e Geração

O gráfico 1 acima apresenta uma população de cem genomas, utilizando *seed* de valor 1, e uma taxa de mutação de 20%. Baseado no gráfico, com a evolução dos genomas é possível visualizar o crescente aumento dos genomas compostos pelos melhores *fitness*(A), também é possível visualizar um aumento dos piores *fitness*(B) na geração 2 e a sua queda nas gerações posteriores, isso é causado pelo substituição dos novos genomas gerados pela *crossover*.

A melhor solução, levando-se em consideração um espaço delimitado entre zero e cem para cada um dos parâmetros(*seed*, tamanho da população, numero de gerações), foi (10, 100, 0), sendo encontrando o resultado na geração inicial.

## 5. Conclusão

A AG pode ser considerada uma técnica poderosa. Como é um método heurístico, oferece soluções de qualidade variada de acordo com geração de novos indivíduos, encontrando uma solução subótima ou ótima para um determinado problema. No corrente trabalho a AG foi usado para solucionar o

problema da busca do movimento de uma braço robótico articulado com três graus de liberdade, e como resultado foi encontrado um conjunto de soluções após testes empíricos, sendo a melhor delas para os parâmetros (seed: 10, população: 100, geração: 0, mutação: 20%).

O sistema desenvolvido permite que o usuário controle os parâmetros de entrada de forma a buscar a melhor solução, e apresentando os dados encontrados como solução ao simulador do braço robótico, para que o mesmo demonstre graficamente o resultado.

Como sugestão de trabalhos futuros, propõe-se o teste em um ambiente instável, com obstáculos, estáticos ou dinâmicos, para uma melhor análise sobre o AG.

## **6. Agradecimentos**

Agradeço ao meu orientador pela paciência que teve com o desenvolvimento deste trabalho, a minha família, aos amigos, ao tempo em que o Projeto Rede Ciranda ofereceu para a conclusão deste trabalho, ao Project Coco, a FEMA e ao Dr. Alex Sandro Romeo De Souza Poletto pela chance de desenvolver um Projeto de Iniciação Científica.

## **7. Bibliografia**

FOGEL, Lawrence Jerome; OWENS, Alvin J; WALSH, Michael John. Artificial Intelligence Through Simulated Evolution, Hoboken, Nova Jersey. John Wiley & Sons, 1966.

HOLLAND, John H. *Adaptation in Natural and Artificial Systems*, Cambridge, Massachusetts: MIT Press, 1992.

JUNIOR, Wedson G. Silveira. Cálculo Cinemático Inversa para Manipuladores Utilizando Algoritmos Genéticos. Instituto Federal do Sul de Minas Gerais. Disponível em <http://ocs.ifsuldeminas.edu.br/index.php/jcinc/jcinc/paper/viewFile/104/96>.

Acesso em: 27 de julho de 2015.

MITCHELL, Melanie. *An Introduction to Generic Algorithms*, Cambridge, Massachusetts: MIT Press, 1999.

NORVIG, Peter; RUSSEL, Stuart J. , *Artificial Intelligence A Modern Approach*. Englewood Cliffs, New Jersey : Prentice Hall, 2009.

RECHENBERG, Ingo. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Alemanha. Frommann-Holzboog, 1973.

SHREINER, Dave; SELLERS, Graham; KESSENICH, John; LICEA-KANE, Bill. *OpenGL Programming Guide*. 8<sup>o</sup> ed. Nova Jersey, Michigan, EUA. Pearson Education, 2013.