Computação Evolutiva: desvendando os algoritmos genéticos

Evolutionary Computing: unleashing genetic algorithms

Sílvio Petroli Neto[1] , FAJ, USF

Resumo

Este trabalho apresenta os Algoritmos Genéticos, uma classe diferenciada de algoritmos baseada em seleção natural, destinados à busca de soluções para problemas de otimização. Inicialmente as origens deste tipo de algoritmo são comentadas e também as técnicas necessárias para sua implementação, incluindo a codificação de cromossomos, a função de avaliação, a seleção, a reprodução, a recombinação e a mutação. Ao final é desenvolvido um exemplo e analisados os resultados obtidos, o que permite confirmar a adequação e conveniência desta técnica como ferramenta para resolução de problemas computacionais.

Palavras-chave. inteligência artificial; computação evolutiva; algoritmos genéticos.

Abstract

This paper presents genetic algorithms, a different class of algorithms based on natural selection, aimed at seeking solutions to optimization problems. Initially, the origins of this type of algorithm are discussed and also the techniques required for its implementation, including the encoding of chromosomes, the evaluation function, selection, reproduction, recombination and mutation. At the end an example is developed and analyzed the results obtained, which allows to confirm the adequacy and appropriateness of this technique as a tool for solving problems.

Keywords. artificial intelligence; evolutionary computing; genetic algorithms.

> Não é o mais forte da espécie que sobrevive, nem o mais inteligente, é o que melhor se adapta à mudança.
>
> —Charles Darwin

1.Introdução

A *Computação Evolutiva* é o termo empregado para designar o conjunto de técnicas de resolução de problemas baseados em princípios de evolução biológica, como a seleção natural e a herança genética, as quais podem ser empregadas para a solução de uma grande gama de problemas, com aplicações práticas na indústria e comércio. Dentre estas técnicas, destacam-se os algoritmos ditos *genéticos*, que empregam princípios semelhantes ao da seleção natural para busca de soluções otimizadas (EIBEN & SMITH, 2003).

Antes de se falar sobre *Algoritmos Genéticos* (AG's) deve-se abordar, mesmo que sucintamente, a área onde os mesmos estão inseridos, ou seja, precisa-se falar um pouco sobre a Inteligência Artificial (IA).

A IA pode ser dividida grosseiramente em quatro fases (BARCELLOS, 2000; RUSSEL & NORVIG, 2004):

1. *Período Subsimbólico*. Nesse período, talvez devido à limitação computacional, a representação do conhecimento era feita numericamente e não simbolicamente. É nesse período que surgem as Redes Neurais Artificiais e os primeiros Algoritmos Evolucionários (AE);

2. *Período Simbólico*. Nessa fase, surgem os algoritmos que utilizam representação simbólica do conhecimento, como a Lógica de Predicados e as Redes Semânticas;

[1]Mestrando. Docente da Faculdade de Jaguariúna-FAJ (Jaguariúna-SP) e da Universidade São Francisco-USF (Itatiba-SP).

3. *Período de Conhecimento-Intensivo*. Essa fase é caracterizada pela grande quantidade de conhecimento incorporada aos sistemas de aprendizagem;

4. *Período Atual*. Hoje os pesquisadores estudam as diversas áreas, combinando e integrando as vantagens tanto quanto possível, tentando encontrar melhores soluções.

O Algoritmo Genético surgiu rudimentarmente no período subsimbólico da IA, como método de pesquisa no qual se procurava conseguir uma boa solução para problemas nos quais o espaço de pesquisa era muito grande para uma enumeração completa (busca exata), principalmente levando-se em consideração os recursos da época.

Hoje os Algoritmos Genéticos estão presentes em diversas áreas de pesquisa, seja buscando uma solução, seja ajudando a reduzir o campo de busca (GOLDBERG, 1989A, 1989B). Infelizmente, a maioria dos artigos existentes sobre o assunto não o aborda de forma simples, com exemplos claros de como essa ferramenta pode ser utilizada na solução dos mais diversos tipos de problemas. Ao longo desse trabalho, faremos uma pequena explanação sobre as origens dos AG's e as técnicas necessárias para a implementação dos mesmos. Ao final, teremos um exemplo claro de utilização, mostrando que sua implementação é simples e deve ser utilizada, não como a ferramenta e sim como mais uma ferramenta disponível para resolução de problemas em computação.

2.Darwin, a origem de tudo

O naturalista Charles Robert Darwin (1809-1882), como membro ortodoxo da igreja Anglicana, não aceitava a teoria da evolução, já inserida anteriormente por Jean Baptist Lamark (1744-1829) que afirmava que *as formas de vida inferiores se formam continuamente a partir de matéria inanimada e que o caminho para uma maior complexidade é guiado pela natureza*. No entanto, quando o ornitólogo John Gould lhe informou que suas espécies de tordos-dos-remédios das ilhas Galápagos eram tão diferentes que pareciam uma nova espécie, o pensamento de Darwin começou a mudar. Ele começa então a aceitar a transmutação de espécies e a juntar evidências nesse sentido.

Em setembro de 1838, Darwin teve a idéia do mecanismo de seleção natural:

> "aconteceu de eu ler, como entretenimento, o ensaio de Malthus sobre a população e, estando bem preparado para avaliar a luta pela existência que prossegue em toda parte pela longa e continuada observação dos hábitos de animais e plantas, imediatamente percebi que, sob essas condições, variações favoráveis tenderiam a ser preservadas e as desfavoráveis destruídas" (*apud* FUTUYAMA, 1992).

Em 1859, Darwin lança seu livro intitulado *A origem das espécies*, que estendeu o conceito de um universo em constante mudança aos seres vivos e introduziu o conceito de mutabilidade ao acaso. Dessa forma, tudo, não só os seres inanimados, mas também os seres vivos estavam sujeitos às leis da física, criando assim uma visão muito mais simples e mecanicista, livre dos paradigmas de existir um propósito, seja ele divino ou não.

Darwin foi o primeiro a introduzir o conceito de ancestralidade comum. O conceito de *luta pela sobrevivência* já existia e até explicava o desaparecimento das espécies, mas nunca havia sido considerada como fator de aparecimento de novas, através da seleção natural. Dessa forma, as variações individuais entre organismos não eram mais, necessariamente, imperfeições, mas poderiam servir de material para moldar novas espécies, talvez melhor adaptadas e preparadas para a sobrevivência.

3.O Algoritmo genético

Os *Algoritmos Genéticos* formam uma classe de pesquisa baseada em seleção natural. Podemos dividir as técnicas de pesquisa em (BARCELLOS, 2000):

- *Técnicas Baseadas em Cálculos* – utilizam um conjunto de condições necessárias e suficientes que devem ser satisfeitas pelas soluções de um problema de otimização. Esses métodos utilizam-se de cálculos matemáticos, como derivadas de primeira e/ou de segunda ordem para dar a direção de busca do ponto ótimo a ser encontrado.

- *Técnicas Enumerativas* – Procuram o ponto ótimo pesquisando sequencialmente cada um dos pontos do espaço de busca, que deve ser finito e discreto para que o algoritmo encontre uma solução.

- *Técnicas dirigidas por pesquisa aleatória* – Utilizam técnicas enumerativas, mas usam algumas informações adicionais para dar a direção da pesquisa.

Dessa forma, podemos entender os Algoritmos Genéticos como sendo uma técnica dirigida por pesquisa aleatória que usa, como informação adicional, o processo de seleção natural visando, principalmente, resolver problemas de otimização.

John Holland, no início da década de 70 acreditou que, utilizando os postulados de seleção natural descritos por Charles Darwin, poderia criar algoritmos computacionais capazes de manipular cadeias de informações (*genes*) de forma a construir organismos complexos, melhores adaptados, para resolver o problema de sua existência. Dessa forma, apenas os organismos melhores adaptados sobreviveriam em detrimento de outros, menos adaptados. Tais organismos seriam os melhores para representar a solução de um problema complexo, desde que, o ambiente no qual estivessem inseridos fosse capaz de fazer a seleção de forma correta.

O algoritmo de Holland resolvia problemas complexos de uma forma bem simples (HAUPT & HAUPT, 2004). Assim como na natureza, o algoritmo não tinha a informação de qual problema estava resolvendo, mas conseguia, através de uma função de avaliação, fazer com que somente os organismos com maiores condições de chegar ao resultado conseguissem sobreviver e se reproduzir, passando para os descendentes a sua carga genética e aumentando assim a chance de continuar, enquanto os menos adaptados eram eliminados.

Em 1975, no seu livro *Adaptation in Natural and Artificial System*, John Holland publicou o primeiro trabalho, cunhando a expressão *Algoritmos Genéticos*. De um outro lado, alguns pesquisadores alemães lançaram uma versão um pouco diferente, conhecida por *Estratégias Evolucionárias*.

3.1.Iniciando o algoritmo

O funcionamento correto de um AG dependerá, basicamente, de duas condições iniciais:

- *Codificação do Cromossomo* – Os cromossomos podem ser codificados utilizando-se uma cadeia de *bits* ou diretamente por números inteiros. É importante nesse ponto ressaltar que eles devem conseguir representar uma solução do problema. Outra consideração importante, no que diz respeito à codificação do cromossomo, é que se deve ter em mente que o mesmo deverá possuir *genes* (material genético que poderá ser trocado entre os cromossomos na reprodução). Apesar de parecer estranho, um *gene* nada mais será que uma divisão do cromossomo, ou seja, o cromossomo criado deverá ter a propriedade de conseguir ser dividido em partes. É nesse sentido que entra a cadeia de *bits*, na qual cada *bit* é um *gene* e o conjunto é o cromossomo.

- *Função de Avaliação* – Imagine que já foram criados os cromossomos e que agora pode ser feito que os mesmos se reproduzam, criando novos cromossomos formados a partir da troca genética de seus pais. Como saber quais são os cromossomos que devem continuar a se reproduzirem e quais devem ser eliminados? Daí a importância da função de avaliação. Ela faz a ligação do algoritmo com o problema real. A função de avaliação deverá ser capaz de analisar qual cromossomo possui melhores características para continuar. Fazendo uma analogia com a natureza, seria o ambiente onde os seres vivos encontram-se inseridos e para o qual necessitam se adaptar.

3.2.Uma visão simples do funcionamento

O algoritmo proposto por Holland é uma visão simplista do funcionamento do mesmo, mas que pode ser empregada com bastante sucesso. Um AG pode ser escrito da seguinte forma (DAVIS, 1991):

1. Inicie uma população de tamanho `N` aleatoriamente, ou seja, criar `N` cromossomos com conteúdo aleatório que será a população inicial;

2. Aplique a função de avaliação em cada um desses cromossomos, obtendo uma classificação dos mais adaptados ao problema;

3. Faça o cruzamento dos cromossomos, levando-se em consideração que os mais adaptados devem ter maiores chances de reprodução. Esse procedimento poderá gerar novos cromossomos.

4. Aplique mutação a uma pequena porcentagem dos cromossomos criados. A mutação pode proporcionar o aparecimento de boas surpresas.

5. Elimine os cromossomos da população antiga, de forma que os novos cromossomos gerados possam ser inseridos sem alterar o tamanho da população inicial;

6. Aplique a função de avaliação e insira os melhores selecionados na população anterior, gerando uma nova população;

7. Se a população de cromossomos atual representar o resultado esperado ou se a quantidade de gerações máxima definida foi atingida, pare. Caso contrário, volte à etapa 3.

O tamanho `N`, da população inicial, é uma variável do algoritmo e seu valor depende da complexidade do problema. Valores típicos encontram-se entre 20 e 200 indivíduos.

Ao final do algoritmo acima se espera que a população de cromossomos gerada seja a melhor adaptada à função de avaliação, sendo dessa forma, a que melhor represente o resultado do problema. Analisando o algoritmo, consegue-se perceber a importância de uma boa codificação para o cromossomo e de uma função de avaliação que, efetivamente, seja capaz de classificar os cromossomos em melhores e piores. Essa tarefa inicial de geração do cromossomo e da função de avaliação não é uma tarefa fácil, uma vez que, para que a mesma produza bons frutos, precisa-se conhecer muito bem a estrutura do problema a ser resolvido.

3.3.Codificação, a geração dos cromossomos

Como já mencionado, a codificação do cromossomo deverá ser de tal forma que o mesmo represente uma solução do problema e deverá possuir a característica de poder ser dividida em partes menores, que chamaremos de *genes*. Pode-se codificar o cromossomo como sendo um conjunto de números reais, ou um conjunto de números inteiros, ou até mesmo, e mais comumente, um conjunto de *bits*. A importância de se utilizar um conjunto de *bits* é que isso permite a manipulação fácil e eficiente dos operadores genéticos sobre esse cromossomo.

No caso de números inteiros, a codificação para binário poderá ser feita de forma bem simples. Suponha que o cromossomo deverá representar números inteiros entre -31 e +31. Pode-se descrever tais números como um conjunto de 6 *bits*, sendo 1 para sinal e 5 para representar amplitude $2^5=32$. Esses cromossomos serias como ilustrados na Figura 1 e Figura 2.

Figura 1. Exemplo de cromossomo (inteiro negativo).

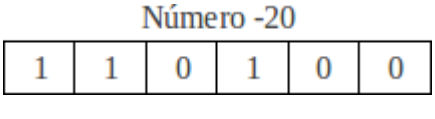
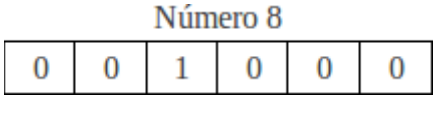


Figura 2. Exemplo de cromossomo (inteiro positivo).



No caso de parâmetros com valores reais, pode-se trabalhar com uma expressão em formato inteiro. Supondo que a faixa de valores está entre 3.56 a 10.78, pode-se multiplicar os valores dos parâmetros por 100 e representar o cromossomo com o número inteiro resultante. Quando se desejar o valor real será só realizar a operação inversa, dividindo o número inteiro presente no cromossomo por 100.

Outros cromossomos podem ser codificados, dependendo da estrutura e da complexidade do problema que se deseja resolver. O importante é que, seja qual for a codificação utilizada, ele consiga representar os parâmetros do problema real com eficiência e clareza.

3.4.Função de avaliação

A função de avaliação é basicamente um modo de avaliar o quão distante um dado cromossomo está da solução ótima. É ela quem faz a ligação entre o problema real e o Algoritmo Genético, uma vez

que os cruzamentos são feitos com os algoritmos melhores adaptados. Sem a função de avaliação seria impossível determinar se um problema está evoluindo para a solução ótima ou não.

Pode ser, por exemplo, uma equação matemática. O resultado da função de avaliação pode ser normalizado de tal modo que, quando seu valor for 1, indica um cromossomo melhor adaptado e, quando seu valor for 0, indica um cromossomo pouco adaptado, porém, a normalização não é necessária.

Vamos supor que se deseja encontrar, dentre vários números inteiros, qual o que melhor representa `x` na função `3x+6=30`. Para esse problema poderíamos ter como função de avaliação o quanto `3x+6-30` está distante de zero. Quanto mais distante de zero for o valor da função de avaliação, mais distante de sua raiz e, portanto, menos adaptado é o cromossomo que representa `x`. Por outro lado, quanto mais próximo o valor da função acima estiver de zero, mais próximo de sua raiz, então mais adaptado será o cromossomo e maior a chance do valor fornecido por esse cromossomo para `x` ser o valor que resolve o problema proposto.

É óbvio que o exemplo dado é simplista. Em problemas reais de grande complexidade, a determinação de uma função de avaliação adequada, que possa avaliar com exatidão a condição da população de cromossomos formada não é tão simples, exigindo mais uma vez o domínio do problema proposto para solução. Em virtude disso vale ressaltar que, a exemplo do que acontece na elaboração de quaisquer espécies de algoritmos, não sendo diferente no caso dos genéticos, a compreensão do problema a ser resolvido é de extrema importância e deve ser levada à exaustão antes da tentativa de se resolver o problema.

3.5.Seleção

A seleção no Algoritmo Genético é a forma como, fazendo uso da função de avaliação, serão classificados e selecionados os cromossomos. O método de seleção pode variar muito, desde que consiga rotular cada cromossomo como sendo melhor ou pior adaptado. Um método possível seria apenas ordenar os cromossomos de forma que os primeiros da lista sejam os que apresentem os melhores resultados perante a função de avaliação e os últimos sejam, por outro lado, os que apresentem os piores resultados. Assim, deve ser dada maior possibilidade de cruzamento aos primeiros colocados da lista e os últimos da lista devem ser eliminados da próxima geração.

O método proposto por Holland é conhecido como *método de seleção por roleta* (HOLLAND, 1992; DAVIS, 1991). Supondo `f` como sendo o valor da função de avaliação para um dado cromossomo e `s` como sendo a soma do valor da função de avaliação de todos os cromossomos, cada cromossomo receberá um setor na roleta de tamanho igual ao ângulo dado por `2#f/s`, ou seja, um setor com tamanho proporcional ao grau de adaptabilidade do cromossomo.

O próximo passo é selecionar um cromossomo com probabilidade igual ao do setor por ele obtido, ou seja, a probabilidade de selecionar um cromossomo para reprodução é proporcional à função de avaliação. A seleção é realizada da seguinte forma:

1. Gera-se um número aleatório entre 0 e 2#;
2. Soma-se, em qualquer ordem, o valor do setor alocado, (em qualquer ordem,) continuando a somar enquanto o resultado não ultrapassar o valor do número gerado;
3. O cromossomo cujo setor fizer a soma ultrapassar o valor do número gerado será o escolhido para a reprodução.

O algoritmo acima descreve um funcionamento semelhante ao de uma roleta, no qual os valores vão se sucedendo até parar em um determinado ponto. É óbvio que os setores maiores têm maiores chances de serem escolhidos. Não há necessidade de se converter os valores obtidos com a função de avaliação em graus ou radianos. Isso pode ser feito apenas para compreender a técnica utilizada. O que é feito na prática, é apenas somar a função de avaliação de todos os cromossomos da população e gerar um número aleatório entre zero e essa soma.

A cada dois cromossomos escolhidos dessa maneira como pais, dois novos cromossomos são gerados a partir deles. O processo de seleção termina quando o número de cromossomos formados atingir o

número de cromossomos da população anterior. Há duas maneiras de se criar a nova geração, uma delas é trocando toda a população anterior pela nova população formada; a outra é trocar apenas uma fração da população anterior pela nova geração. No segundo caso, é preciso definir quantos elementos da população serão trocados a cada nova geração.

3.6.Reprodução

A reprodução é o processo através do qual se combina dois cromossomos (pais) para formar filhos a partir deles. Tais filhos herdam as características dos pais através da herança de material genético (*genes*). Esse processo é conhecido como *recombinação*.

Além da recombinação, está envolvida no processo de reprodução, a *mutação*. Na mutação, altera-se aleatoriamente o material genético de uma pequena parcela dos filhos gerados.

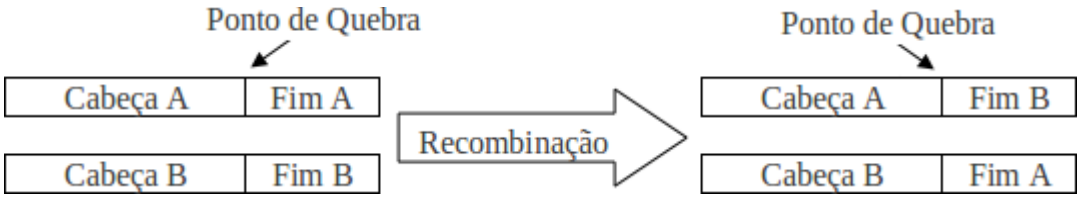
3.6.1.Recombinação

A recombinação é o processo de troca de material genético entre dois cromossomos. Pode-se definir aqui a probabilidade de recombinação, que indica a probabilidade de dois indivíduos escolhidos no processo de seleção virem a se reproduzir. Essa probabilidade será chamada de `PC`, que geralmente está entre 0.5 e 1.0. O processo de recombinação se dá da seguinte maneira (BARCELLOS, 2000; DAVIS, 1991):

1. Gera-se aleatoriamente um número entre 0 e 1;
2. Se este número for menor ou igual a `PC`, haverá recombinação; caso contrário não haverá recombinação;
3. Escolhe-se aleatoriamente um ponto no qual haverá a quebra do material genético;
4. Quebram-se os dois cromossomos envolvidos neste ponto, juntando-se a primeira parte do cromossomo `A` com a segunda parte do cromossomo `B` e a primeira parte do cromossomo `B` com a segunda parte do cromossomo `A`, gerando assim dois filhos.

A Figura 3 ilustra o processo de recombinação em um ponto de quebra.

Figura 3. Processo de recombinação.



Pode-se também quebrar o cromossomo em dois ou mais pontos. Nesses casos, o número de filhos gerados com dois cromossomos pode ser maior que dois.

Não existe uma regra fixa de como a recombinação deve ser feita. Deve-se usar de bom senso na hora da reprodução para que o processo evolua para descendentes mais adaptados e melhores que seus pais.

Geralmente, o processo utilizado é a quebra em um ou dois pontos. Uma outra técnica vem sendo utilizada para a recombinação e é conhecida como PMX (*Partially Matched Crossover*). Dados dois cromossomos pais, o PMX copia uma *substring* de *genes* de um dos pais diretamente na mesma posição do filho. As posições restantes se completam com valores que ainda não tenham sido utilizados, na mesma ordem em que se encontram em um dos pais. Supondo que Pai_1=12463758 e Pai_2=54172683 e que a substring seleciona aleatoriamente do Pai_1 seja `463` e ela tem relação com a *substring* `172` no Pai_2. Então a sequência de operação obterá um filho com `54463683` e eliminando as repetições teremos `5*463*8*`. Arrumando agora ficaria assim `51463*8*`, já que o 4 havia ocupado o lugar do 1 no Pai_2. Continuando o processo obtem-se o `Filho=51463782`.

3.6.2.Mutação

O operador de mutação é utilizado para que sejam gerados novos cromossomos distintos dos pais (BARCELLOS, 2000; KOZA, 1992). Tal fato é de grande importância, pois a população inicial é gerada aleatoriamente e as novas gerações são frutos da recombinação da população inicial. Suponha que nenhum cromossomo da população inicial possua carga genética do ponto ótimo. Se isso ocorrer, passarão muitas gerações até que surja um cromossomo que possua as características desejadas, ou ainda, pode ser que isso nunca venha a ocorrer.

O operador de mutação troca, aleatoriamente, a carga genética de uma pequena porcentagem dos cromossomos gerados. Essa porcentagem é dada através de um parâmetro fornecido *a priori*, que varia usualmente entre 0.1% e 2%. Na prática, o que ocorre é escolher, aleatoriamente, um *gene* de alguns cromossomos e trocar o seu valor por outro também aleatório. Se os cromossomos forem representados por cadeias de *bits*, isso significa trocar o valor 0 por 1 e vice-versa.

4.Um exemplo prático

Será utilizado como exemplo, a minimização de uma função, construindo passo a passo o algoritmo genético que encontrará os valores de x que produzem o menor valor possível para a função dada. Tal processo é conhecido como otimização.

A função escolhida como exemplo foi $(x_1-2)^4+(x_1-2x_2)^2$; e o problema pode ser descrito como na Equação 1.

Equação 1. Mínimo da função de avaliação

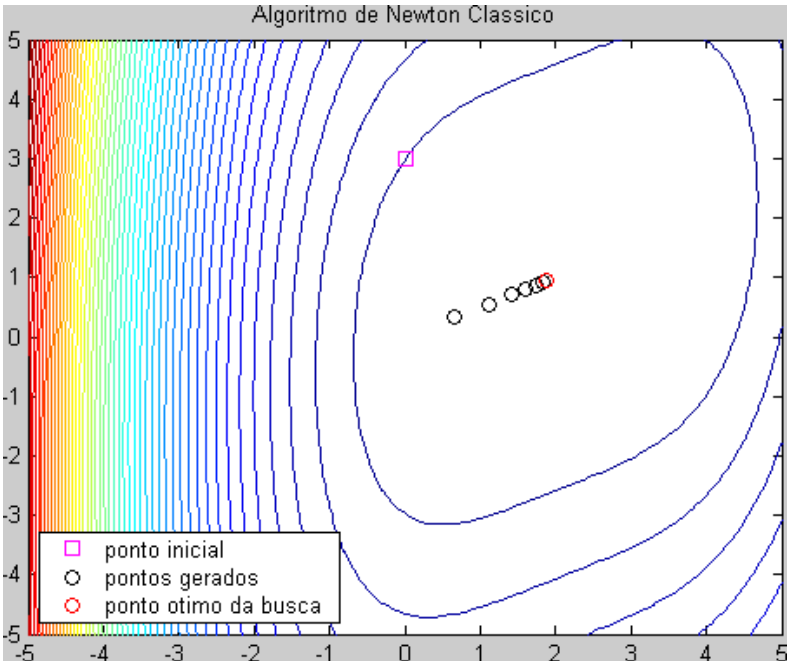
$$Min(x_1-2)^4+(x_1-2x)^2$$

Aplicando o Método de Newton Clássico, que é um conhecido método de otimização numérica, obtivemos o resultado da Equação 2, ilustrado na Figura 4.

Equação 2. Resultado da Equação 1

$$x_1=1.8829$$
$$x_2=0.9415$$

Figura 4. Resultado obtido pelo método de Newton Clássico.



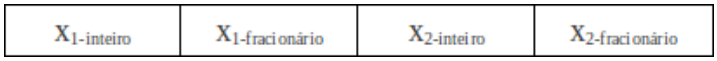
O resultado obtido pelo método exato será utilizado apenas para comparação com os resultados obtidos pelo algoritmo genético a ser desenvolvido.

Os passos descritos nessa Seção serão seguidos para construção de um algoritmo que seja realmente eficaz na resolução do problema acima.

4.1.Codificação do cromossomo

Através do método de Newton Clássico, sabemos que o valor de x_1 é aproximadamente 2.0 (dois) e o valor de x_2 é aproximadamente 1.0 (um). Com isso, não serão representados números negativos, utilizando como campo de busca apenas valores positivos entre zero e 100.999. Tendo estes números uma parte inteira e uma parte fracionária, o respectivo cromossomo terá a forma ilustrada na Figura 5.

Figura 5. Representação do cromossomo.



A divisão em inteiro e fracionário se dá pelo fato de deseja-se a representação binária do cromossomo. Agindo dessa forma, cada uma das partes do cromossomo dado pode ser representada por um número inteiro, que pode ser facilmente convertido para a base dois.

O tamanho de x_1 e x_2 dependem do campo de busca, ou seja, em qual faixa de valores o algoritmo genético deve buscar o valor ótimo. Como dito, é feita uma busca entre zero e 100.999, o que indica uma parte inteira entre 0 (zero) e 100 (cem) e uma parte fracionária entre 0 (zero) e 1000 (mil).

Para compor os números formados será utilizada a função da Equação 3.

Equação 3. Função de avaliação

$$\begin{cases} x_1 = x_{1-inteiro} + 0.001 x_{1-fracionário} \\ x_2 = x_{2-inteiro} + 0.001 x_{2-fracionário} \end{cases}$$

4.2.Função de avaliação

A função de avaliação será dada pela própria função que se deseja minimizar. Cada novo valor de x_1 e x_2 obtido através do cromossomo deverá ser aplicado à função $(x_1-2)^4+(x_1-2x_2)^2$ e o valor obtido, comparado ao valor obtido por outro par x_1 e x_2. O par que resultar em um menor valor para a função será considerado melhor adaptado e continuará na população. Para utilizar o método da roleta na seleção, atribui-se a cada cromossomo selecionado um peso. Suponha uma população inicial igual a cinco, o cromossomo melhor adaptado recebe peso cinco, o posterior recebe peso quatro, o seguinte três e assim sucessivamente até terminar a população. O último cromossomo, pior adaptado, receberá peso um.

4.3.Início da população

Como temos o cromossomo dividido em duas partes, o número de cromossomos diferentes será muito grande, igual a $100 \times 1000 \times 100 \times 1000 = 10^9 = 10$bilhões. Isso indica que, para termos uma representação coerente dos valores, precisamos de uma população inicial bem grande, da ordem de dois mil a cinco mil cromossomos. Uma vez escolhido o número de cromossomos da população inicial, ela pode ser gerada aleatoriamente. Isso será feito gerando números aleatórios entre zero e cem para as partes inteiras e entre zero e mil para as partes fracionárias.

4.4.Seleção

A seleção dos cromossomos que deverão se reproduzir será feita através do método da roleta. Como cada cromossomo, após a classificação, terá um peso, somaremos todos os pesos. Assim, se o número

de cromossomos for igual a cinco e os pesos atribuídos a cada um deles é cinco, quatro, três, dois e um respectivamente, a soma dos pesos é igual a quinze. Agora gera-se aleatoriamente um número entre zero e quinze e, se o número gerado for menor que cinco, ele representa o cromossomo um, se estiver entre cinco e nove, representa o cromossomo dois, e assim sucessivamente. Pode-se notar que a probabilidade de ser escolhido um cromossomo melhor adaptado é muito maior do que a de se escolher um pior adaptado. Por exemplo, enquanto a probabilidade de se escolher o cromossomo um é igual a `5/15`, a de se escolher o cromossomo cinco é igual a `1/15`.

4.5.Reprodução

A reprodução está dividida em duas partes: a recombinação e a mutação.

4.5.1.Recombinação

A recombinação será feita quebrando-se cada parte do cromossomo em duas partes e recombinando como já descrito. Não podemos tratar o cromossomo, nesse caso, com uma única cadeia de *bits*, e sim cada parte como sendo um cromossomo diferente, ou seja, deveremos recombinar a parte inteira de x_1 do primeiro cromossomo selecionado com a parte inteira de x_1 do segundo cromossomo selecionado; a parte fracionária de x_1 do primeiro cromossomo selecionado com a parte fracionária de x_1 do segundo cromossomo e assim sucessivamente. Não poderemos misturar parte inteira com fracionária nem sequer x_1 com x_2.

A posição de corte será escolhida aleatoriamente, ou seja, a cada recombinação o local onde as partes do cromossomo serão quebradas, será obtido aleatoriamente entre zero e o número de *bits* da parte.

Depois de cada parte recombinada separadamente, junta-se novamente as partes formadas para obter um só cromossomo. Cada casal de cromossomos gerará dois novos cromossomos filhos.

4.5.2.Mutação

A mutação será feita escolhendo-se aleatoriamente dois por cento da população. Nesses dois por cento escolhidos será escolhido um *bit* de cada parte para ser trocado, ou seja, será invertido um *bit* da parte inteira de x_1, um *bit* da parte inteira de x_2, um *bit* da parte fracionária de x_1 e um *bit* da parte fracionária de x_2.

4.5.3.Eliminação e geração da nova população

Após a reprodução, a população de cromossomos dobrará de tamanho, uma vez que foi dada a oportunidade de reprodução para metade dos cromossomos. Para se manter o mesmo número de cromossomos a cada geração, os novos cromossomos gerados foram adicionados à população anterior e todos foram avaliados. Os cromossomos são ordenados de forma que os mais adaptados fiquem no início da população. Após a ordenação a metade inferior dos cromossomos é eliminada, voltando a população ao seu tamanho inicial.

4.6.O algoritmo

O algoritmo foi desenvolvido, segundo o modelo descrito nesse capítulo e utilizando-se o software MatLab®, devido à facilidade em tratar funções matemáticas. Pode ser descrito da seguinte forma:

1. Inicia-se a população inicial;

2. Avaliam-se os cromossomos gerados, atribuindo a eles um peso;

3. Selecionam-se os melhores e faz-se a recombinação entre eles;

4. Faz-se a mutação em 2% dos novos cromossomos gerados;

5. Avaliam-se novamente os cromossomos e eliminam-se os piores adaptados;

6. Se obteve o resultado ou se findou o número de gerações esperadas, parar o algoritmo; caso contrário, voltar ao passo dois.

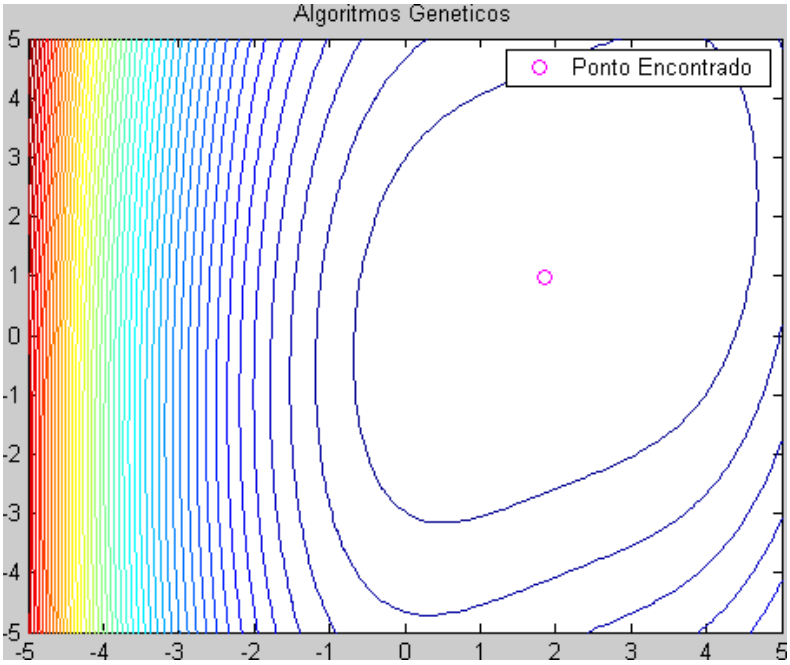
4.7.Resultados obtidos

Para uma população de dois mil cromossomos, em apenas cinco gerações, obteve-se o resultado da Equação 4, ilustrado na Figura 6.

Equação 4. Resultado do exemplo para 2000 cromossomos

$x_1 = 1.8600$
$x_2 = 0.9700$

Figura 6. Resultado obtido pelo algoritmo genético.

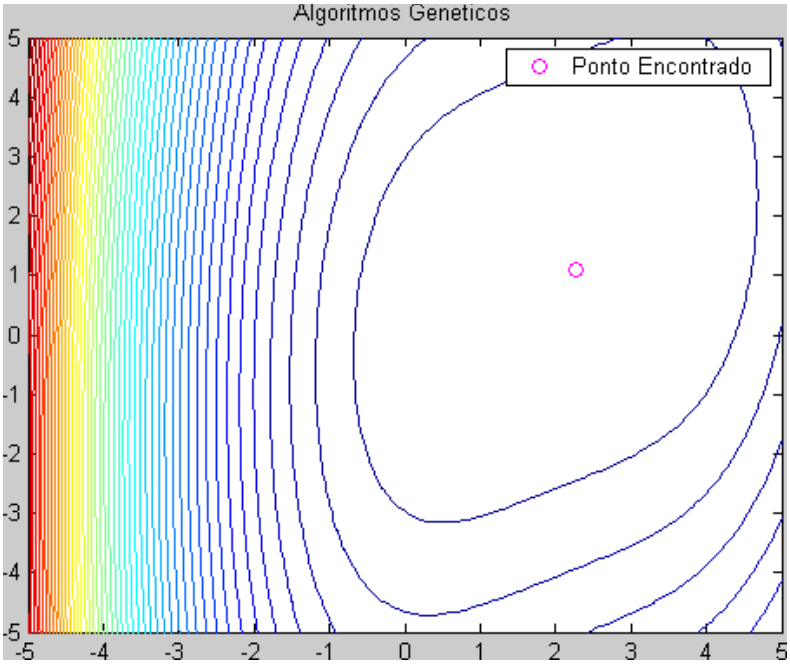


Para uma população de cinco mil cromossomos, em apenas cinco gerações, obteve-se o seguinte resultado da Equação 5, ilustrado na Figura 7.

Equação 5. Resultado do exemplo para 5000 cronossomos

$x_1 = 1.2800$
$x_2 = 1.1000$

Figura 7. Resultado obtido pelo algoritmo genético, variando-se o número da população inicial.



5.Conclusões

Como se pode observar, os Algoritmos Genéticos conseguiram resolver o problema de uma forma bastante rápida e com uma precisão semelhante a um método exato. A diferença é que o Método de Newton, utilizado aqui como exemplo, é muitas vezes mais complexo e exige muito mais poder computacional, sem falar na quantidade de conhecimento necessária para se programá-lo.

A Computação Evolutiva é uma ferramenta prática e fácil de utilizar e a idéia desse artigo é apenas deixar claro que ela existe e que não apenas pode, mas deve ser utilizada como mais uma ferramenta para resolução de problemas computacionais, sobretudo quando temos um vasto campo de busca.

Nos dias de hoje, os AG's vêem sendo amplamente utilizados nas mais diversas áreas como: otimização, mineração de dados, planejamento estratégico, entre outras.

Como perspectivas futuras temos a adaptação dos algoritmos para utilização nas mais diversas áreas, o aprimoramento das técnicas já utilizadas e o desenvolvimento de novas técnicas e regras de seleção, combinação e mutação.

6.Referências bibliográficas

BARCELLOS, J. C. H. *Algoritmos genéticos adaptativos: um estudo comparativo*. Dissertação de mestrado. Universidade Estadual de São Paulo, Escola Politécnica. São Paulo: 2000.

DAVIS, L. *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhould, 1991.

EIBEN, A.E. & SMITH, J.E. *Introduction to Evolutionary Computing*. Berlin: Springer, 2003.

FUTUYAMA, D. J. *Biologia Evolutiva*. tradução: Fábio de Melo Sene. Ribeirão Preto: Sociedade Brasileira de Genética, 1992.

GOLDBERG, D.E. *Messy Genetic Algorithms: Analysis and First Results*. In: Complex Adaptive Systems. Boston: MIT Press, 1989A.

_____. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley, 1989B.

HAUPT, R. L. & HAUPT, S. E. *Practical Genetic Algorithms*. 2nd Edition, New York: Wiley-Interscience, 2004.

HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. Boston: MIT Press, 1992.

KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. In: Complex Adaptive Systems. Boston: MIT Press, 1992.

RUSSEL S. & NORVIG P. *Inteligência Artificial*. Rio de Janeiro: Campus, 2004.