

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA

GUSTAVO ALVES PACHECO

IMPLEMENTAÇÃO TORNEIO E ELITISMO

Uberlândia-MG

2019

GUSTAVO ALVES PACHECO

IMPLEMENTAÇÃO TORNEIO E ELITISMO

Trabalho apresentado ao curso de Engenharia de Computação da Universidade Federal de Uberlândia, como requisito parcial de avaliação da disciplina de Algoritmos Genéticos.

Prof.: Keiji Yamanaka

Uberlândia-MG

2019

SUMÁRIO

1.	INTRODUÇÃO	4
2.	OBJETIVOS	4
3.	MATERIAIS E MÉTODOS	5
4.	RESULTADOS E DISCUSSÕES	5
5.	CONCLUSÃO	9
6.	REFERÊNCIAS BIBLIOGRÁFICAS	11

1. INTRODUÇÃO

Anteriormente, foi desenvolvido um algoritmo genético simples, que utilizava a roleta com aptidão cumulativa para seleção. Além disso, o algoritmo prévio visava minimizar uma função, exibindo os indivíduos sobrepostos ao gráfico da mesma. Tais cromossomos eram representados como strings de tamanho fixo (10), com uma precisão de 0.5 para o range escolhido.

Entretanto, nota-se alguns pontos que podem melhorar nesse design. O primeiro é a resolução dos indivíduos. Ao fazer com que o tamanho da string varie, fica possível aumentar ou diminuir a precisão dos resultados, aumentando significativamente a confiabilidade das saídas.

Outra problemática que a estratégia adotada traz, é o fato de que há uma probabilidade (relativamente alta) de que haja cruzamento entre os indivíduos e, da forma como foi implementado, uma extinção dos indivíduos pais. Isso pode representar um malefício, visto que os filhos podem apresentar características piores que os progenitores e, na pior situação, ocasionar uma geração que possua um desempenho menor que a anterior. É aí que a ferramenta chamada “Elitismo” atua. A cada geração, se garante que os ‘N’ melhores indivíduos sejam copiados para a próxima, ainda permitindo que eles participem do processo de seleção e cruzamento.

Um outro ponto é o método de seleção em si. Embora fácil de implementar, o método da roleta força uma perpetuação dos melhores indivíduos, e pior, garante que super-indivíduos monopolizem o espectro de seleção. Tal fato pode acarretar em convergências incorretas à mínimos locais, sem possibilidades de varreduras mais abrangentes. Uma solução proposta para isso é o método de “Torneio”, no qual um número ‘K’ de cromossomos são escolhidos e, entre eles, há a seleção do melhor, ou dos ‘N’ melhores, que passarão normalmente pelos próximos processos evolutivos, de crossover e mutação, no caso em questão.

Outra adição a ser feita para melhorar o monitoramento da evolução é a implementação de duas curvas de desempenho. Uma exibe o melhor indivíduo, enquanto a outra mostra a média geral da população, a cada geração.

A função a ser maximizada passa a ser a seguinte:

$$f(x) = x * \sin x^4 + \cos x^2 ; 0 \leq x \leq 3$$

2. OBJETIVOS

- Aprimorar o conhecimento sobre algoritmos genéticos e obter experiência prática na implementação dos mesmos.
- Implementar o método de seleção Torneio, permitindo ao usuário selecionar entre o mesmo e a roleta, fornecendo o valor de K.
- Implementar o Elitismo, permitindo ao usuário escolher se será utilizado ou não, e ainda obter o valor de N.
- Implementar as curvas de desempenho.
- Adaptar o algoritmo anterior para maximização.
- Realizar testes modificando os métodos de seleção, bem como os coeficientes dos mesmos.
- Verificar a influência e impacto do elitismo.
- Analisar as curvas de desempenho.

3. MATERIAIS E MÉTODOS

Os materiais e métodos utilizados foram os mesmos dos usados anteriormente. Tais técnicas estão reescritas abaixo:

Para implementação do algoritmo genético foi utilizada a linguagem de programação Common Lisp, compilando-a com o SBCL (Steel Bank Common Lisp). Como interface de desenvolvimento, foi utilizado o Emacs, configurado com a plataforma SLIME (The Superior Lisp Interaction Mode for Emacs) para melhor produtividade.

Para confecção da interface gráfica, utilizou-se o pacote LTK, uma *binding* para CL (Common Lisp) do kit de desenvolvimento TK, que utiliza a linguagem TCL.

Como métodos, utilizou-se uma abordagem *top-down*, com auxílio de um framework SCRUM para desenvolvimento ágil. O código produzido segue majoritariamente uma abordagem funcional, mas possui elementos procedurais, para facilitar a leitura de algumas funções.

Durante o desenvolvimento, a documentação de funções se mostrou fundamental. Principalmente, para que algumas estruturas de dados fossem melhor especificadas, e a execução do código fosse correta.

4. RESULTADOS E DISCUSSÕES

Primeiramente, se mostrou necessário realizar algumas alterações ao código original. As principais características dessas mudanças foram visando organização e performance. Houveram alterações na interface (Figura 1), para abarcar as novas implementações, enquanto as funções de plotagem precisaram ser refeitas, com foco em agilidade e proporção correta das exibições, levando em conta que a nova função a ser

implementada possui um range muito menor que a anterior. Prezou-se também pela portabilidade e reutilização do código, procurando a facilidade nas alterações.

Começou-se o processo de implementação dos novos requisitos pelos processos de visualização, aproveitando a lição de que a interface gráfica é de grande auxílio ao desenvolvimento. Foram confeccionadas funções que redimensionavam as impressões, além de permitir ao usuário escolher o range mínimo e máximo de variação da função. A maior dificuldade encontrada nesse ponto foi a de manter a proporção entre o tamanho exibido, e o tamanho máximo a ser plotado, sendo que tal dilema foi resolvido com a função:

$$x_{ATT} = \frac{x * tam_{TELA}}{range_{MAX}}$$

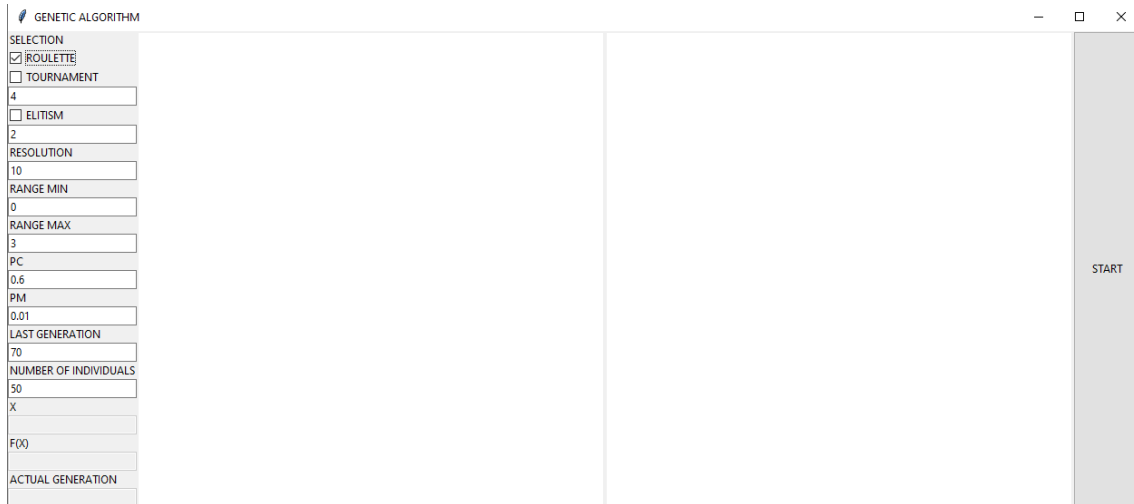


Figura 1: Interface com suporte às novas opções de configuração

Em seguida, foi implementado o método de torneio. Tal desenvolvimento foi bem simples, já que era aproveitada a população com aptidão, e indivíduos eram selecionados aleatoriamente. Tal método de seleção se mostrou bem superior à roleta, nos testes realizados. Para valores de K próximos a 3, os resultados foram bem satisfatórios (Figura 2). A convergência foi rápida e bem precisa. Entretanto, em alguns testes, o máximo global não foi encontrado, pois o algoritmo se prendeu a máximos locais. Tal fato ocorreu com uma frequência bem menor, se comparado às ocorrências na roleta. Nota: para K = 1, a busca fica totalmente aleatória (Figura 3).

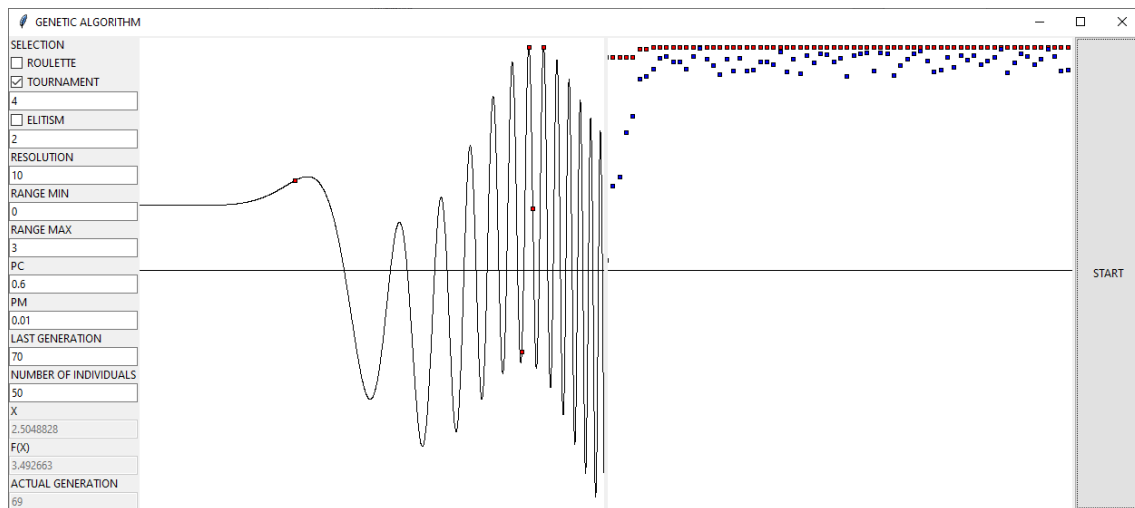


Figura 2: Execução do algoritmo usando torneio, $k = 4$. Em vermelho a curva de melhor indivíduo por geração, e em azul a média geral

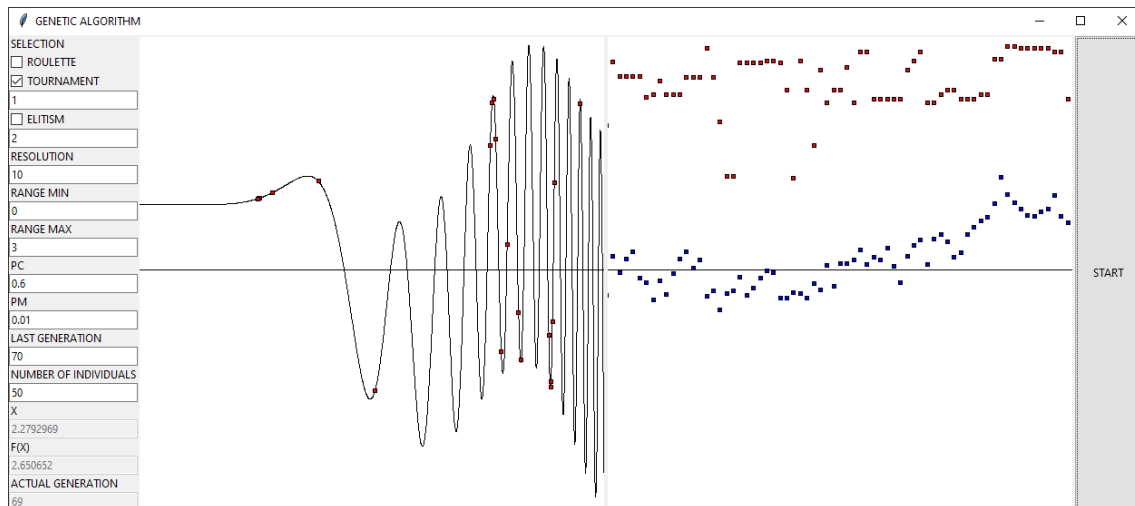


Figura 3: Um coeficiente $k = 1$ representa uma busca completamente aleatória

O processo de elitismo também foi fácil de implementar. Para ele, antes de cada seleção, os melhores indivíduos eram automaticamente copiados para a nova população, e esse número era descontado do total de novos indivíduos a ser adicionado. Tal detalhe é bem importante, pois casos com números de indivíduos, ou de indivíduos selecionados pelo elitismo, ímpares, deve-se levar em conta que um indivíduo não passará pelo crossover, ou que no processo de crossover, um dos filhos será descartado.

Uma estratégia fácil para observar a influência do elitismo é utilizar o torneio com $K = 1$ e ligar a opção de elitismo. Mesmo que a busca seja aleatória, os melhores indivíduos são copiados. Portanto, não há convergência, mas o melhor valor é encontrado, como mostra a figura 4, na qual a curva vermelha mostra realmente o melhor indivíduo, mas a azul fica aleatória.

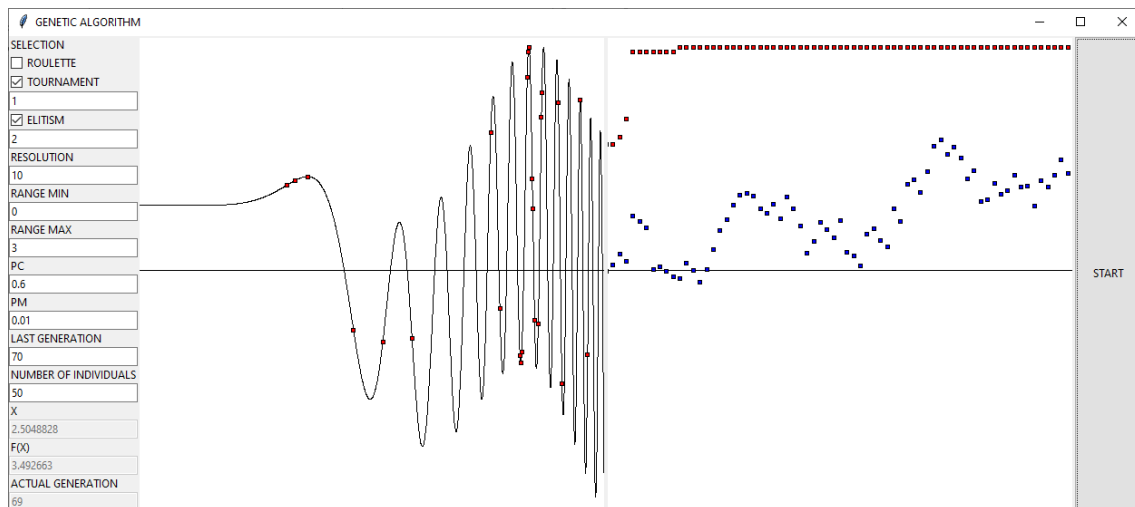


Figura 4: Torneio K=1 e elitismo

Deve-se tomar cuidado ao utilizar elitismo, visto que o mesmo acarreta em várias convergências equivocadas para máximos locais, principalmente quando utilizado com valores mais altos de N (Figura 5).

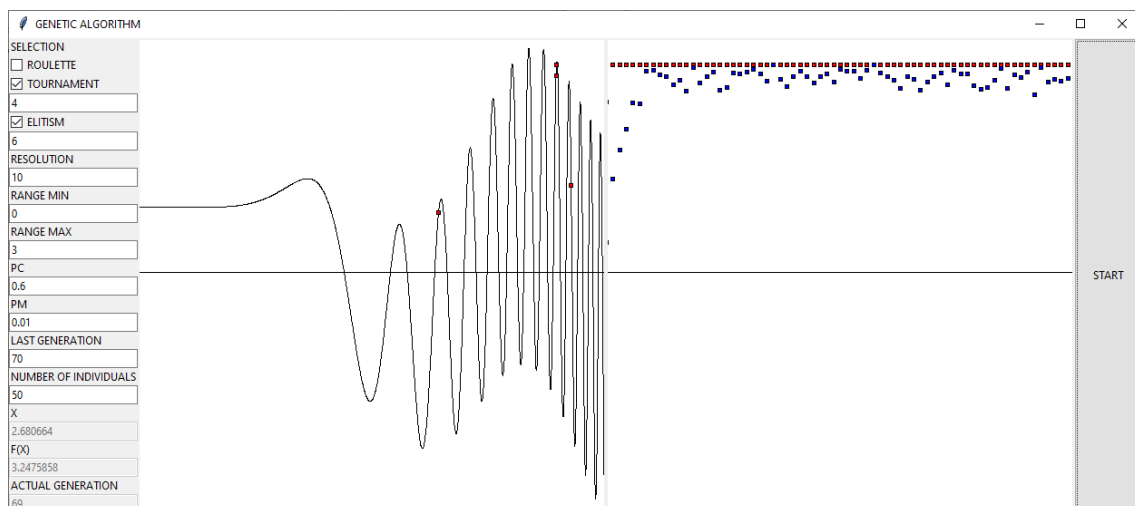


Figura 5: Altos valores de N acarretam convergências equivocadas

As curvas de aptidão foram uma ótima adição ao algoritmo. Com elas, é possível ver a convergência das populações, verificando quão rápido se dá a evolução dos indivíduos (Figura 6). Foi possível visualizar algumas vezes a problemática discutida na seção 1, de que algumas gerações apresentam aptidão absoluta menor que as anteriores, que é resolvido com o elitismo, como exibido na figura 7, abaixo.

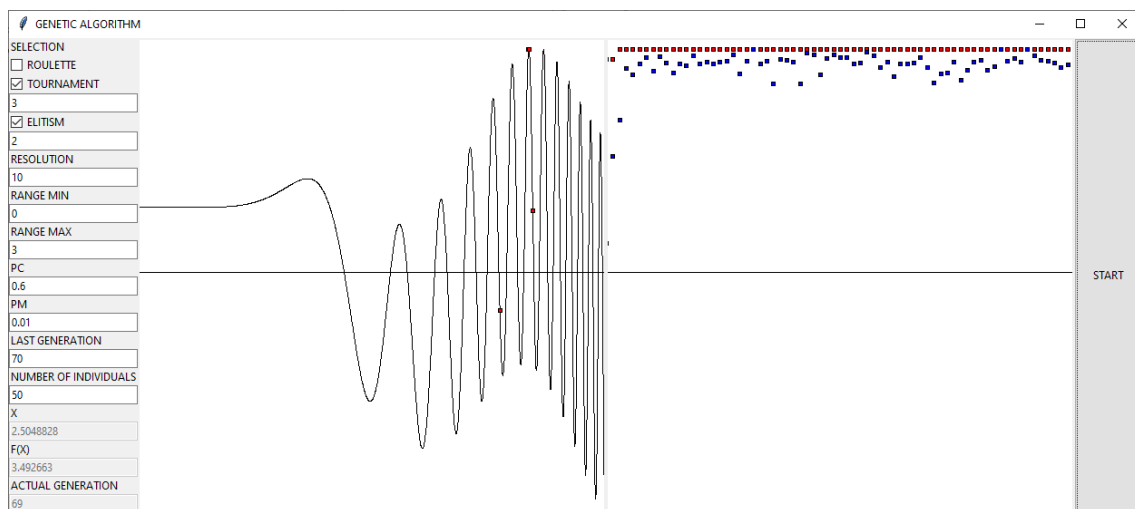


Figura 6: Execução do algoritmo utilizando torneio ($K=3$) e elitismo ($N=2$).

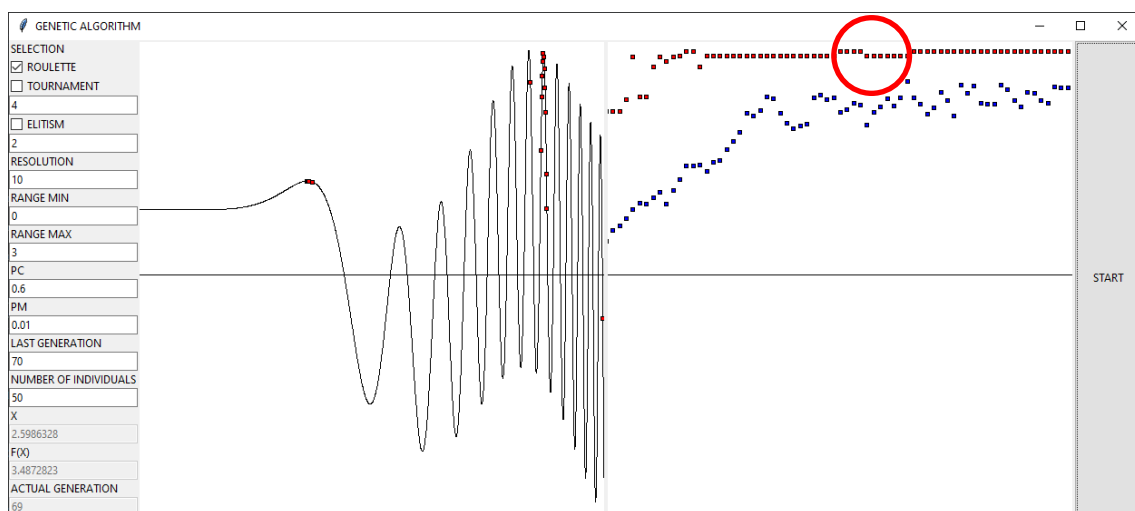


Figura 7: Execução do algoritmo utilizando roleta. Detalhe para o retrocesso da aptidão da população.

5. CONCLUSÃO

Assim como anteriormente, é fato que essas técnicas são espelhadas diretamente do processo evolutivo natural, embora mais simplificadas. Até o momento, tem-se mostrado que a precisão e exatidão do algoritmo aumenta à medida que o modelo se assemelha mais ao da natureza.

Com a adição das duas técnicas citadas, o torneio e o elitismo, é notório o aumento na eficácia do algoritmo. Apesar de que em algumas vezes, esse ganho em eficácia represente uma perda de eficiência, no geral o benefício que elas trazem justificam sua

utilização. Percebe-se que o torneio facilita bastante em casos em que é possível o surgimento de super indivíduos, pois ele permite que todos os cromossomos tenham, inicialmente, a mesma chance de competição. Já o elitismo se mostra muito vantajoso, principalmente em cenários com vários indivíduos, pois garante que as próximas gerações sejam, no mínimo, iguais à anterior.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- Mallawaarachchi, V. (7 de Julho de 2017). *Introduction to Genetic Algorithms*. Fonte: Towards Data Science: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>
- Montesanti, J. d. (s.d.). *Seleção natural*. Fonte: InfoEscola: <https://www.infoescola.com/evolucao/selecao-natural/>
- Tomassini, M. (s.d.). A Survey of Genetic Algorithms. *Annual Reviews of Computational Physics, Volume III*.
- Yamanaka, P. K. (Agosto de 2017). *ALGORITMOS GENÉTICOS: Fundamentos e Aplicações*.
- Zini, É. d., Neto, A. B., & Garbelini, E. (18 de Novembro de 2014). ALGORITMO MULTI OBJETIVO PARA OTIMIZAÇÃO DE PROBLEMAS RESTRITOS APLICADOS A INDÚSTRIA. *Congresso Nacional de Matemática Aplicada à Indústria*.