

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA

GUSTAVO ALVES PACHECO

PARÂMETROS CONTÍNUOS

Uberlândia-MG

2019

GUSTAVO ALVES PACHECO

PARÂMETROS CONTÍNUOS

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	6
5.	CONCLUSÃO	11
6.	REFERÊNCIAS BIBLIOGRÁFICAS	13

1. INTRODUÇÃO

Até o presente momento, os indivíduos apresentavam uma forma de representação em valores discretos. Um exemplo era o uso de uma string binária, sendo que cada posição só poderia assumir dois valores: 0 ou 1. Mesmo que outras técnicas de design fossem utilizadas, os parâmetros dos indivíduos sempre representavam valores discretos, com uma certa precisão fixa, de acordo com a quantidade de bits ou dados usados para armazenamento.

A proposta desse projeto é implementar alternativas para os parâmetros, de forma a deixar os valores contínuos, adquirindo uma precisão e exatidão consideravelmente maiores. Tais características são essenciais em aplicações comerciais, e o aprendizado de tais técnicas se mostrará vantajoso na otimização de diversos problemas.

Para crossover, duas técnicas serão adicionadas, o Crossover de Radcliff e o Crossover de Wright. A mutação irá se alterar para abranger geração de valores dentro de todo o intervalo disponível. Tais técnicas serão discutidas com mais detalhes em [3].

Como teste, deve-se otimizar duas funções:

Maximizar:

$$f(x, y) = x \sin + 1.1 \sin (2y)$$

$$0 \leq x \leq 10; 0 \leq y \leq 10$$

E minimizar a função de Rastrigin:

$$f(x_1, x_2, \dots, x_n) = 10n + \sum_{i=1}^n x_i^2 - 10 \cos (2\pi x_i)$$

$$-5.12 \leq x_i \leq 5.12$$

2. OBJETIVOS

- Desenvolver um algoritmo genético que trabalhe com parâmetros contínuos
- Desenvolver novo design para os cromossomos

- Implementar novos operadores de Crossover e Mutação
- Adequar a interface gráfica utilizada no projeto de ‘Maximização de uma função de duas variáveis’, de forma a abranger os novos operadores, além de possibilitar escolha de maximização/minimização e suporte à função de Rastrigin.
- Verificar impacto dos parâmetros do algoritmo genético, no processo de *tuning*

3. MATERIAIS E MÉTODOS

A linguagem utilizada foi o Racket, uma linguagem de uso geral, baseada em Scheme e Lisp. Como interface de desenvolvimento, foi utilizado o DrRacket.

Para confecção da interface gráfica, utilizou-se a linguagem *racket/gui*, um recurso nativo da própria linguagem Racket. O processo de plotagem foi realizado através das funções do pacote *plot*.

Como métodos, utilizou-se uma abordagem *top-down*, com auxílio de um framework SCRUM para desenvolvimento ágil. Em algumas situações, uma visão *bottom-up* foi adicionada ao projeto, na parte de incremento da própria linguagem, ao criar novos operadores e funcionalidades. 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.

Os novos operadores genéticos trabalham da seguinte forma.

Radcliff:

Sejam os pais:

P1: [pm1, pm2, ..., pmN]

P2: [pd1, pd2, ..., pdN]

sendo N = Número de variáveis da função. Os filhos F1 e F2 serão gerados como:

F1: $p_{novo} = \beta * pmn + (1 - \beta) * pdn$

F2: $p_{novo} = (1 - \beta) * pmn + \beta * pdn$

sendo que n = Enésimo parâmetro e β = Valor aleatório entre 0 e 1.

Wright:

Sejam os pais:

P1: [pm1, pm2, ..., pmN]

P2: [pd1, pd2, ..., pdN]

sendo N = Número de variáveis da função. Serão gerados três filhos, F1, F2 e F3, da forma:

F1: $p_{novo1} = 0.5 * p_{mn} + 0.5 * p_{dn}$

F2: $p_{novo2} = 1.5 * p_{mn} - 0.5 * p_{dn}$

F3: $p_{novo3} = -0.5 * p_{mn} + 1.5 * p_{dn}$

sendo que n = Enésimo parâmetro.

Deve-se ficar atento ao fato de que, nessa situação, é possível gerar indivíduos fora do intervalo disponível. Logo, devem ser substituídos pelos valores corretos dos extremos do intervalo. Além disso, um dos filhos deve ser descartado. Isso é feito aleatoriamente, na implementação adotada.

Mutação Contínua:

No caso de mutação, serão gerados novos valores dentro do intervalo, que substituirão a informação antiga.

4. RESULTADOS E DISCUSSÕES

Em primeira instância, a interface desenvolvida no projeto ‘Maximização de uma função de duas variáveis’ foi adequada. Alguns parâmetros foram removidos, como o de velocidade de animação, e novos foram colocados. Abaixo estão as imagens (Figura 1) de comparação entre as duas interfaces, no quesito de configuração de parâmetros. Optou-se por eliminar a animação das gerações, por já não representar um ganho tão grande à visualização do desenvolvimento, sendo melhor verificar o gráfico de performance. Para 2 ou 3 dimensões, ainda é plotado o gráfico da função, em conjunto com o ponto de máximo ou mínimo encontrado.

FUNCTION	$(\lambda(x y) (+ (* x (\sin (* 4 x))) (* 1.1 (\sin (* 2 y)))))$	FUNCTION	$(y) (+ 21.5 (* x (\sin (* 4 \pi x))) (* y (\sin (* 20 \pi y)))))$
NUMBER OF VARIABLES	2		
MODE	<input checked="" type="radio"/> MAXIMIZATION <input type="radio"/> MINIMIZATION		
<input type="checkbox"/> CONTINUOUS PARAMETERS		<input checked="" type="checkbox"/> 3D	
THRESHOLD	500	THRESHOLD	500
SELECTION	<input checked="" type="radio"/> TOURNAMENT <input type="radio"/> ROULETTE	ANIMATION SPEED	0.03
TOURNAMENT K	4	SELECTION	<input checked="" type="radio"/> ROULETTE <input type="radio"/> TOURNAMENT
ELITISM	20	K	4
STRING SIZE FOR EACH VARIABLE IN TRADITIONAL MODE	10	ELITISM	2
RANGE	'((0 10) (0 10))	RESOLUTION	10
<input checked="" type="radio"/> TRADITIONAL		X MIN	-3.1
CROSSOVER	<input type="radio"/> RADCLIFF <input type="radio"/> WRIGHT	X MAX	12.1
PC	0.6	Y MIN*	4.1
MUTATION	<input checked="" type="radio"/> TRADITIONAL <input type="radio"/> CONTINUOUS	Y MAX*	5.8
PM	0.01	PC	0.6
LAST GENERATION	70	PM	0.01
NUMBER OF INDIVIDUALS	500	LAST GENERATION	70
(X, Y, \dots)		NUMBER OF INDIVIDUALS	50
$F(X, Y, \dots)$		X	
		Y*	
		$F(X, Y^*)$	
START		START	

Figura 1: À esquerda, interface atualizada. À direita, a utilizada no projeto de Maximização de Duas Variáveis.

Em seguida, o código anterior foi adaptado para funcionar corretamente com qualquer número de variáveis. Isso envolveu uma padronização das posições do fitness como parte do indivíduo, e na representação. Indivíduos novos passavam a ser gerados dentro do intervalo disponível, para o caso de parâmetros contínuos, e como string de tamanho n , para descontínuos.

Após essa etapa, os operadores genéticos foram implementados. Tais algoritmos só podem ser utilizados em parâmetros contínuos. Além disso, no cálculo do fitness, foi adicionada a opção de minimização ou maximização, que apenas muda a forma como a função é aplicada (negativa para minimização e positiva para maximização).

Depois disso, alguns ajustes foram necessários, para abarcar os resultados do algoritmo, adequando os valores de threshold, e eliminando a resposta que iria para a animação do processo.

Em ultima etapa, foi desenvolvida a função de Rastrigin, por meio de macros. Ao entrar com um determinado N , a função correspondente era criada, contendo todas as N variáveis necessárias, além de todos os termos da soma.

Alguns dos testes realizados estão listados abaixo.

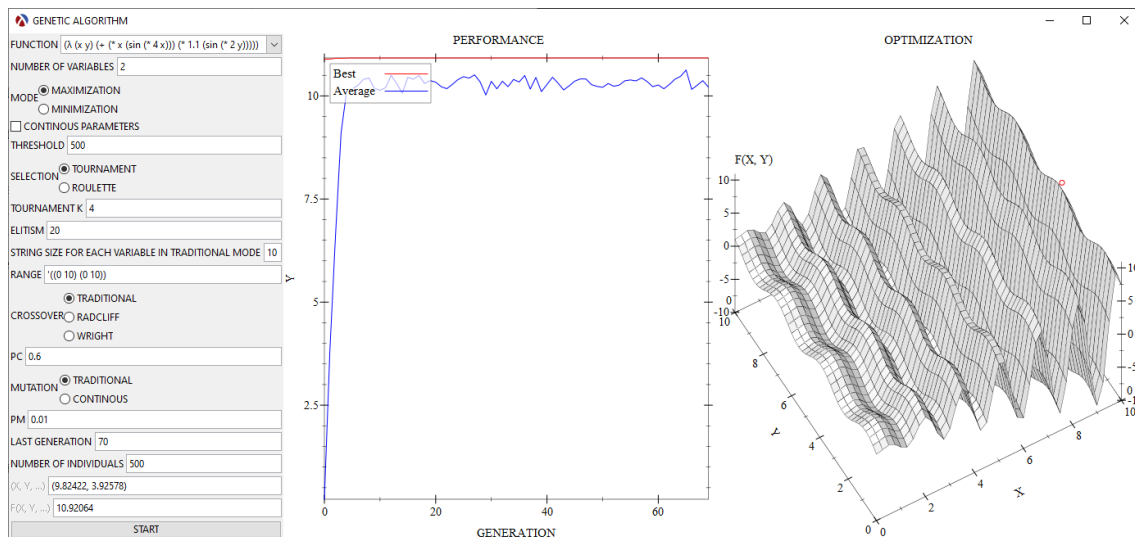


Figura 2: Maximização da função 1, com intervalo especificado. Parâmetros discretos são utilizados

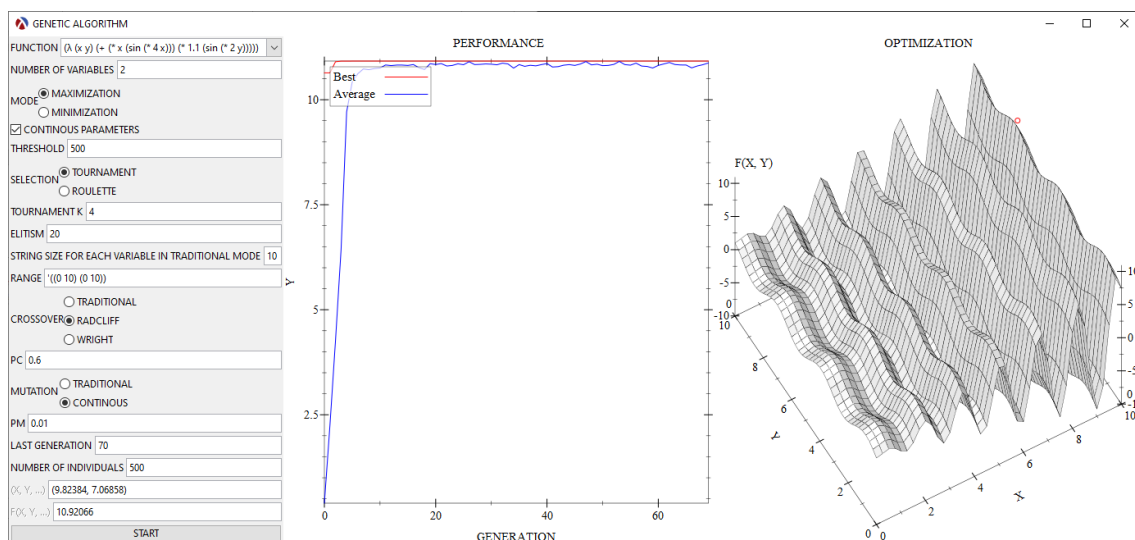


Figura 3: Utilizando parâmetros contínuos, o mesmo valor foi encontrado, mas observa-se que a média da população subiu consideravelmente

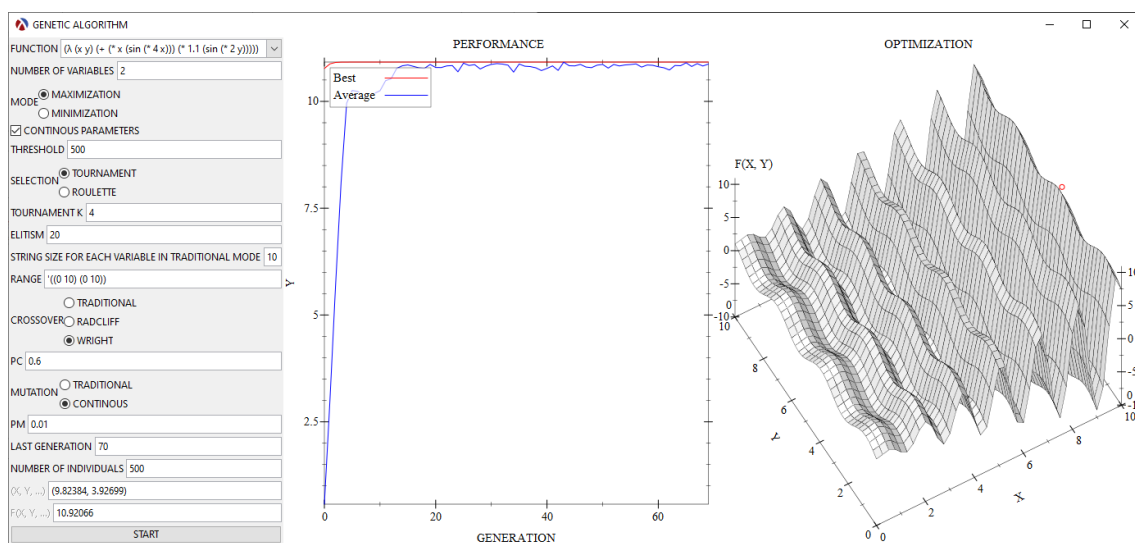


Figura 4: Mesma configuração, utilizando Crossover Wright. O resultado foi semelhante, mas a execução foi mais rápida.

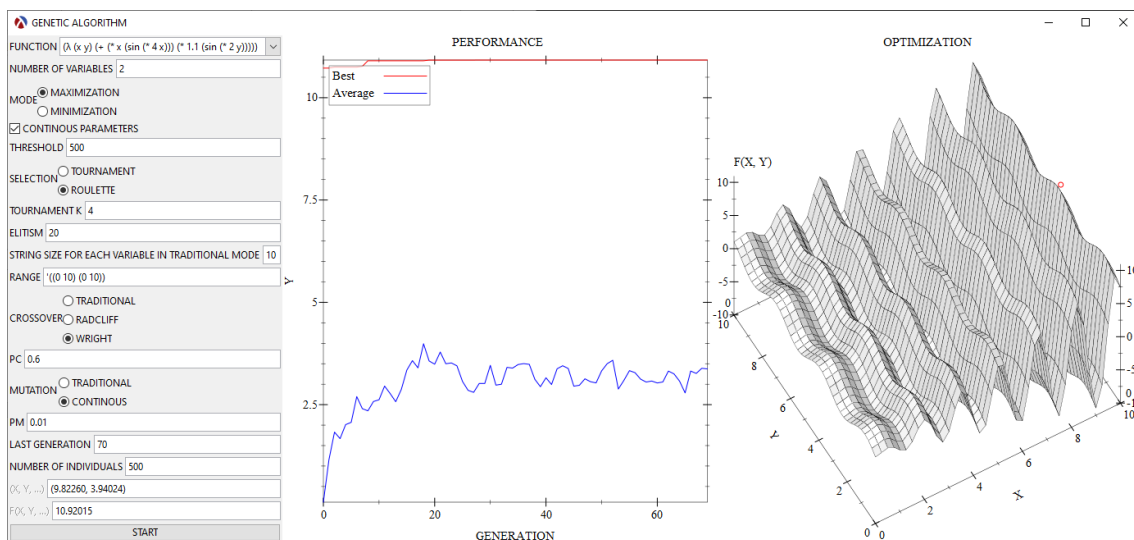


Figura 5: Utilizando roleta, observa-se que a média fica muito baixa, além de representar um tempo de execução consideravelmente maior.

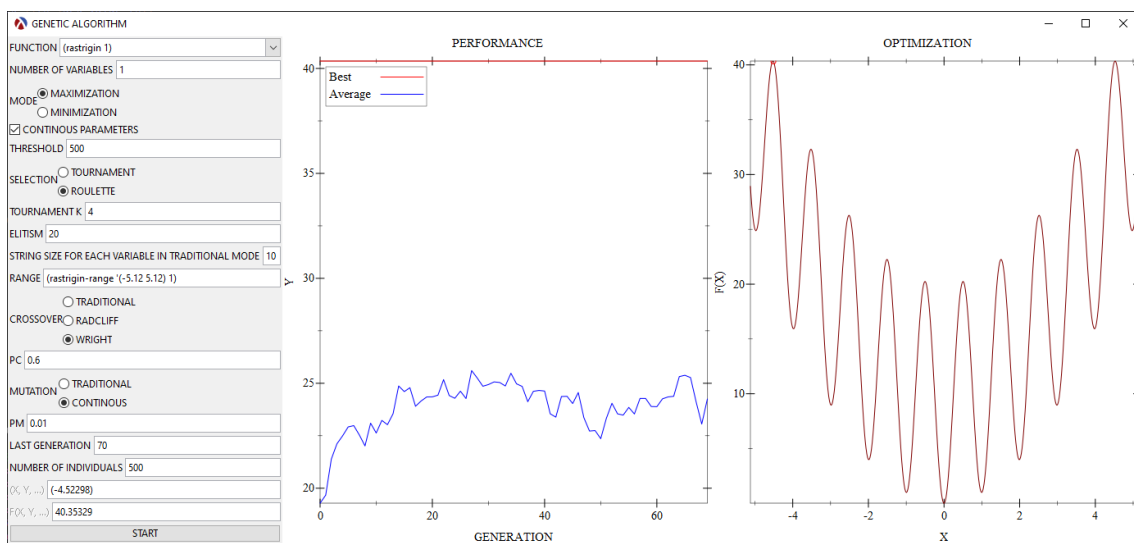


Figura 6: Maximização da função de Rastrigin utilizando roleta e parâmetros contínuos, nota-se a média baixa.

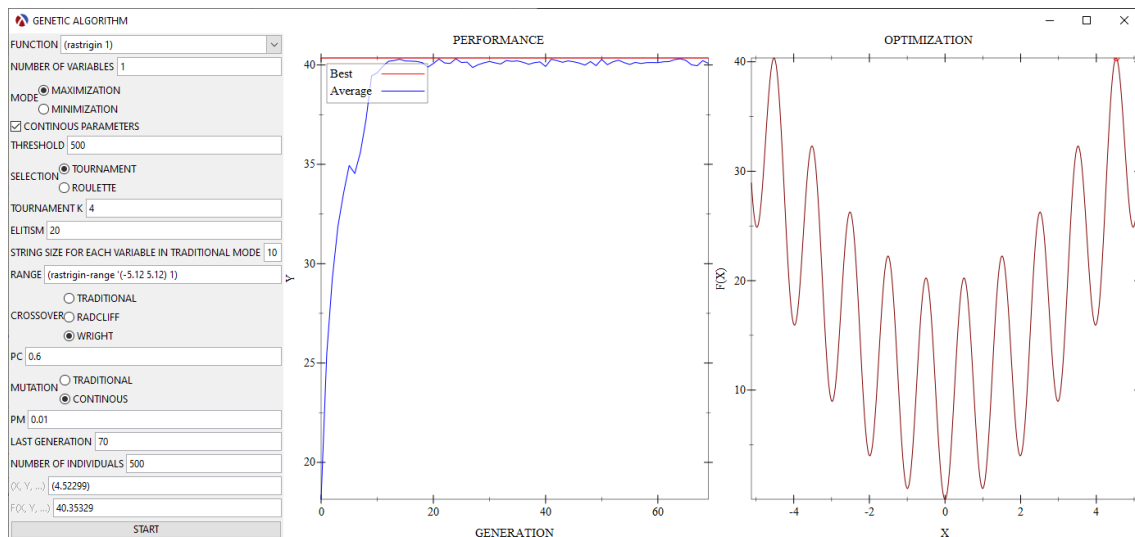


Figura 7: Mesma configuração, utilizando roleta.

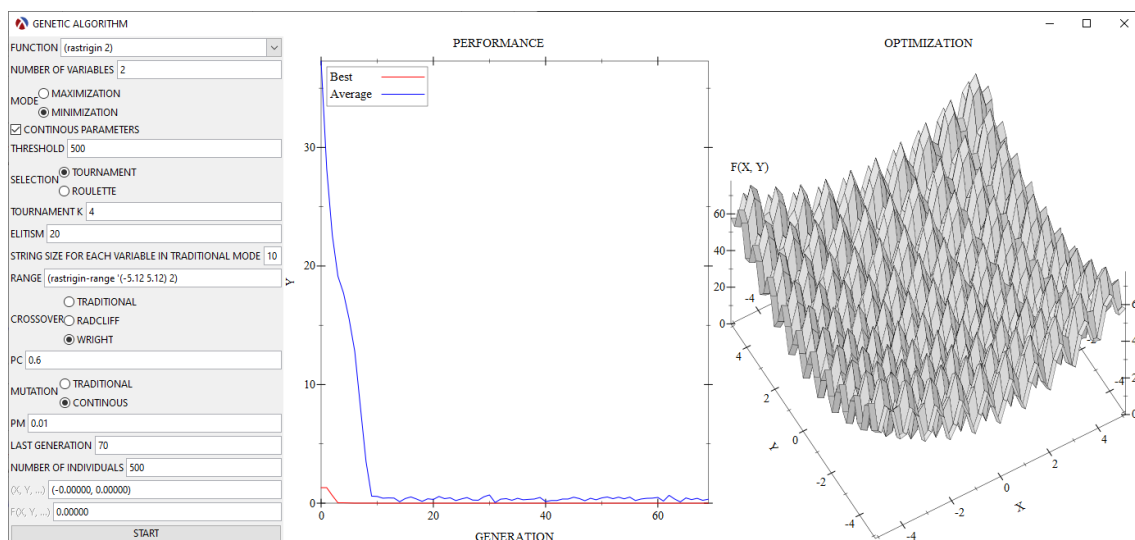


Figura 8: Função de Rastrigin para duas variáveis. Minimização encontrou o valor 0, como esperado. A média foi boa, e o tempo de execução foi pequeno.

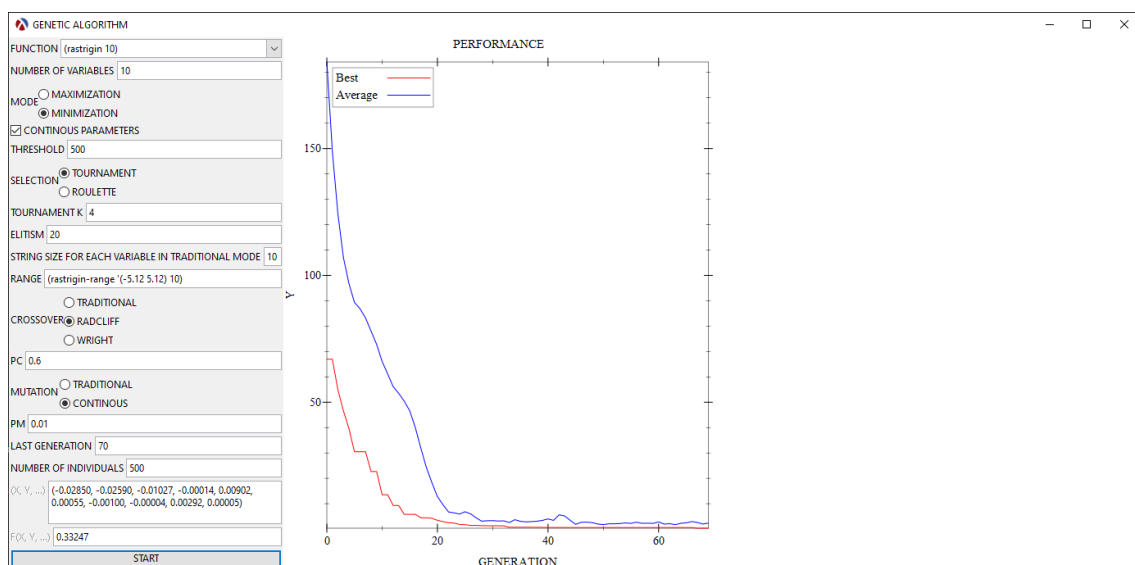


Figura 9: Minimização de Rastrigin para 10 dimensões. Percebe-se que a convergência demora mais para acontecer, além de que o valor de 0 não foi encontrado, embora um valor bem próximo tenha sido a resposta.

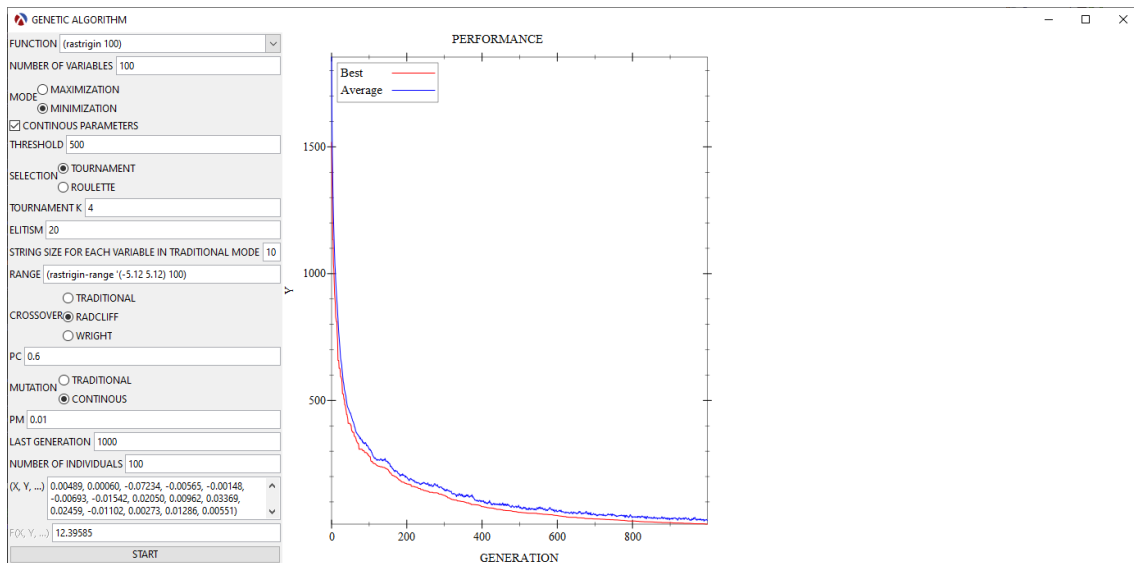


Figura 10: Para 100 variáveis, é necessário aumentar bastante o número de gerações. Nesse caso, o tempo de execução foi de 20099ms, sendo encontrado um valor próximo a 12.

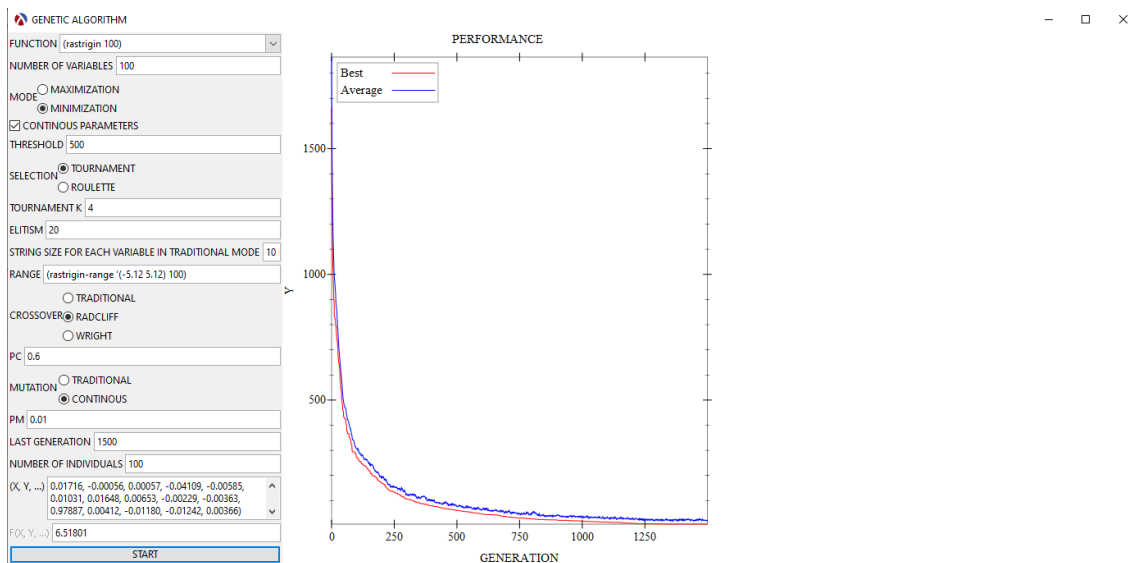


Figura 11: Aumentando o número de gerações, a resposta se aproxima de zero, mas o tempo de execução também aumenta (30941 ms).

5. CONCLUSÃO

Deve-se notar a importância desse projeto, tendo em mente dois novos aspectos principais: A utilização de parâmetros contínuos e a otimização de funções de qualquer número de variáveis.

O primeiro aspecto é extremamente importante para a obtenção de resultados confiáveis, exatos e precisos. O segundo, abre possibilidades para otimização de funções não gráficas, que necessitam de diversos parâmetros, inclusive um número variável de variáveis.

Com essas duas adições, o algoritmo passa a ser mais aceito como forma de resolução de problemas, visto que passa a abranger um grau de confiabilidade à altura de algoritmos não evolutivos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- Al-Gharaibeh, J., Qawagneh, Z., & Al-Zahawi, H. (29 de Outubro de 2015). Genetic Algorithms with Heuristic. *Research Gate*.
- 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 MULTIOBJETIVO PARA OTIMIZAÇÃO DE PROBLEMAS RESTRITOS APLICADOS A INDÚSTRIA. *Congresso Nacional de Matemática Aplicada à Indústria*.