



**UNIVERSIDADE  
FEDERAL DO CEARÁ**  
CAMPUS SOBRAL

**Curso de Engenharia Elétrica**

**Inteligência Computacional**

## **Relatório 3**

**Aluno:** Julio Cesar Ferreira Lima - 393849

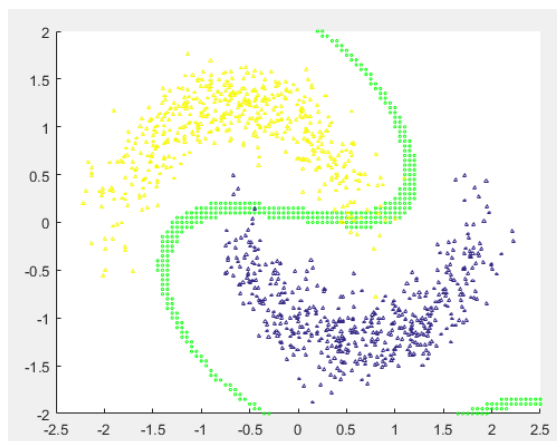
**Prof:** Jarbas Joaci de Mesquita Sá Junior

Sobral, **Junho de 2018.**

### 1. *Extreme Learning Machine*

O algoritmo executa os passos de acordo com a teoria envolvendo ELM. Não há necessidade de explicação sobre a interação com o usuário, este deverá apenas rodar o código.

**Figura 1 – Em azul e amarelo os dados, em verde o correspondente classificador.**



**Fonte:** Autor

### 2. *Radial Basis Function*

O usuário deve entrar com uma percentagem corresponde as amostras de teste que serão usadas. Por algum motivo a toolbox utilizada não aceita um número maior que 40%. Em seguida deverá entrar com a quantidade de épocas para obter uma média.

**Figura 2 – Entrada do usuário no MATLAB.**

```
Neural Network Radial Base Function
Percentage of sample test? (enter a number below then 0.4): .4
How much time do you wanna run before get media?: 50
```

**Fonte:** Autor

### 3. *Genetic Algorithm*

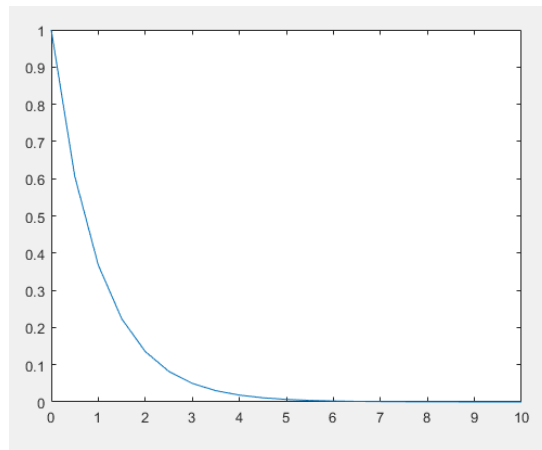
O algoritmo desenvolvido teve como base algumas premissas.

1. Combinação de bits, (*crossover*) baseado em um ponto.
2. A mutação gerada é baseada no parentesco entre os pais, que se tiverem sido gerados na mesma geração a será gerado um número que corresponde a uma probabilidade, aleatoriamente. E se esse for menor que o retorno da função  $f(\text{sameGenLvl}, \text{offsetMutation})$  abaixo, o bit sofrerá mutação.

$$f(\text{sameGenLvl}, \text{offsetMutation}) = e^{-(\text{sameGenLvl} + \text{offsetMutation})}$$

Assim, se o offset de mutação for zero, haverá sempre mutação no bit, no entanto conforme o offset aumenta a probabilidade de mutação cai. Assim como se o parentesco for alto, isso é um número próximo de zero, já que aqui o parentesco é a subtração entre gerações, a probabilidade de haver mutação aumentará. Isso tem o intuito de aumentar a mutação quando os dados convergirem.

**Figura 3 – Exponencial de menos X.**



**Fonte:** Autor

3. A função de avaliação, corresponde a função que se deseja encontrar o máximo local.
4. Há duas formas de se eliminar as gerações a primeira é por meio da idade, estipula-se uma idade máxima a cada robô, fazendo assim com que filhos convivam com os pais e possam gerar novos robôs a partir da combinação entre eles. A segunda forma é por meio de buffer, uma vez que se determina uma quantidade máxima de robôs, sempre que o número de robôs gerados for superior a esses todos os antigos robôs serão eliminados. Desse modo, conforme a escolha da quantidade de robôs a renovação de material genético é elevada, ou não.
5. A reprodução de robôs é levada em conta a avaliação de cada robô os que tiverem melhor resultado em maximizar a função obterão maior probabilidade de gerar descendentes.

### **3.1 Menu**

Todo o menu é mostrado ao usuário assim que iniciado o código.

- 1) Devido a partição do DNA, utilizando-se 20 bits como DNA, logo cada cromossomo terá 10 bits. Foi verificado que cada cromossomo deve iniciar com no mínimo um bit. Então, para iniciar ambos cromossomos com 1 o DNA de Deus deve ser 1024, que corresponde a 00000000010000000001.
- 2) O tamanho do DNA deve ser 20, como pedido.
- 3) O número de robôs é aleatório, e deve ser definido pelo usuário de modo a maximizar a acurácia em tempo.
- 4) As gerações ficam a cargo do usuário.
- 5) O número máximo de um robô deve ser determinado pelo usuário.
- 6) O offset na função de mutação é outro parâmetro que deve ser definido pelo usuário, 0 para 100% de mutação e infinito para 0.
- 7) Caso queira uma plotagem em relação ao número de interações e a saída da função opte pelo gráfico 2D, '2'. Caso haja necessidade de verificar a resposta da saída em relação ao par de DNA do robô, deve se optar pelo gráfico 3D, '3'.

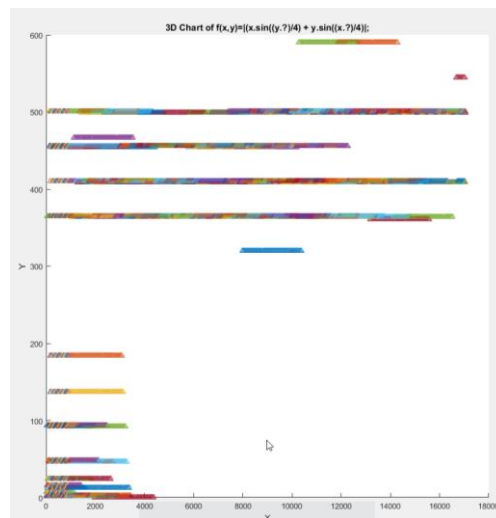
**Figura 4 – Entrada do usuário no MATLAB.**

```
This software gonna try to find the max value for the function:
xsen(y?/4) + ysen(x?/4)
Using Genetic Algorithm, what stop condition do you wanna use?
Type the God DNA: 1025
Type the max lenght of God DNA: 20
Type max number of robots: 100
Type how much generations do you wanna run: 1000
Type the max age of each robot: 2
Type the mutation offset (0 to infinite): 3
Type the type of chart that you wanna see (2 or 3): 2|
```

**Fonte:** Autor

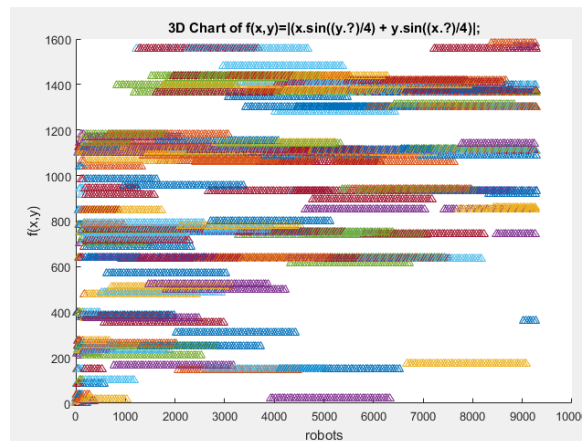
### 3.2 Alguns resultados

**Figura 5 – Baixa taxa de mutação, elevada idade .**



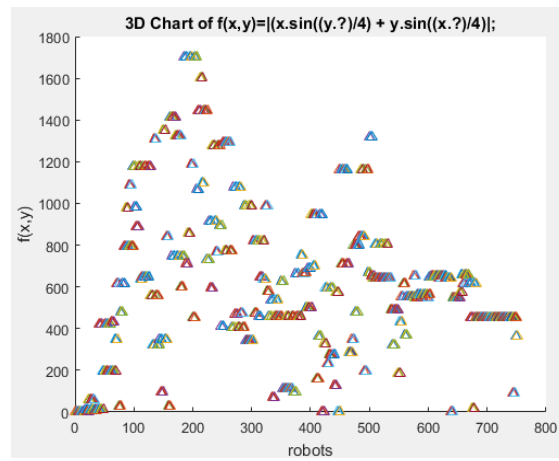
**Fonte:** Autor

**Figura 6 – Média taxa de mutação, elevada idade.**



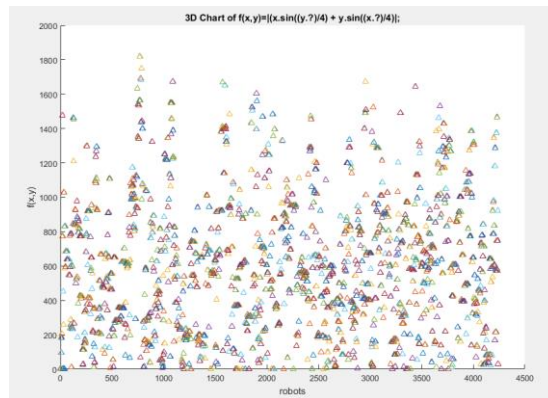
**Fonte:** Autor

**Figura 7 – Alta taxa de mutação, baixa idade.**



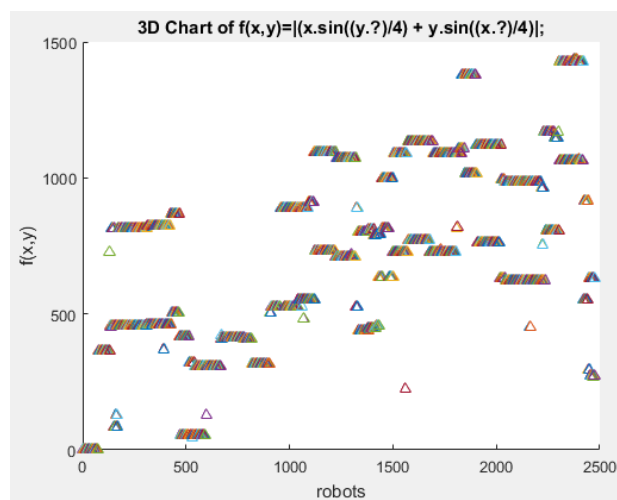
Fonte: Autor

**Figura 8 – Alta taxa de mutação, zero idade.**



Fonte: Autor

**Figura 9 – Alta taxa de mutação, idade 2.**



Fonte: Autor

#### **4. Metodologia**

A base para todo o procedimento experimental desenvolvido nesse experimento/simulação está demonstrada na descrição de código. Portanto, de modo a não onerar este relatório, apenas será mostrado a saída do MATLAB seguido pela resposta em curva do gráfico para cada resposta do sistema.

OBS:.. Visto que a língua padrão de toda a comunidade científica é o inglês e que os códigos produzidos neste trabalho serão disponibilizados na plataforma GitHub, todos os comentários do código, incluindo tomadas de entrada e saída foram escritos em inglês, tornando-o não capaz, no entanto, mais possível, que esse trabalho se torne útil a outrem.

#### **5. Bibliografia**

[2] LINDEN, Ricardo. **Algoritmos Genéticos**. 2ª ed.

#### **6. Repositório**

[1] LIMA, Julio. **Computacional Intelligence**. Disponível em:

<<https://github.com/juloko/computacional-inteligence>>. Acesso em abril de 2018.