

Simulated Annealing Python

October 5, 2018

1 Simulated Annealing em Python

Disciplina: Inteligência Artificial Aplicada;

Professor: Sérgio Nathan;

Grupo: Ana Karoline;

Jonathan Martins;

Lucas Solano Cadengue;

Rafael Dias;

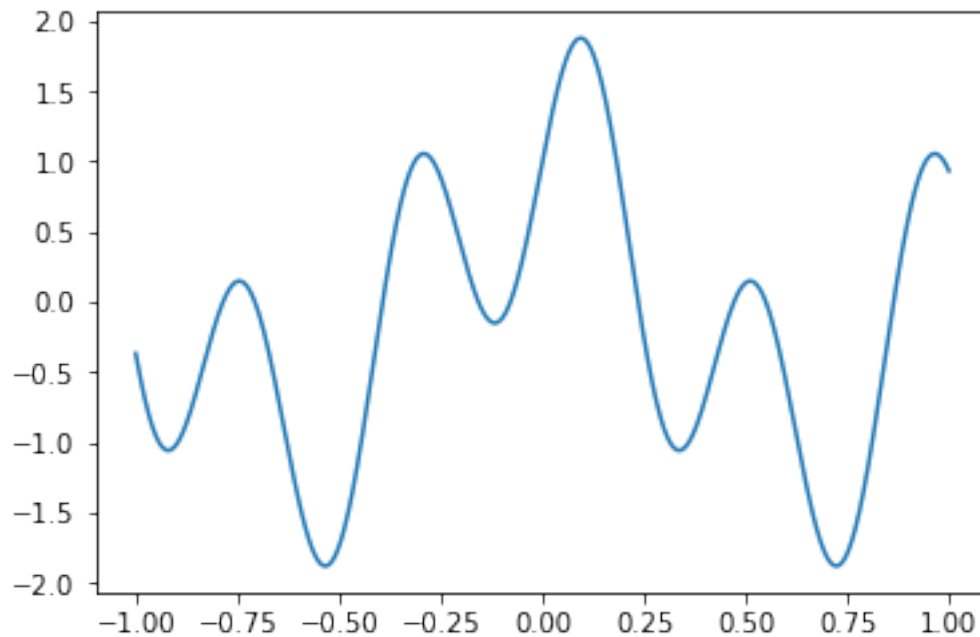
Primeiro iremos importar as bibliotecas e definir a função utilizada para obter o máximo.

```
In [2]: import numpy as np
import matplotlib.pyplot as plt

def funcao(x):
    funcao = np.sin(15*x) + np.cos(5*x)
    return funcao

funcaovetorizada = np.vectorize(funcao)
x = np.linspace(-1, 1, num = 1000)
plt.plot(x, funcao(x))
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x7f611e5399e8>]
```



Agora nós vamos efetivamente fazer a função Simulated Annealing (SA)

```
In [3]: def SA(espacobusca, func, Temperatura):
        scale = np.sqrt(Temperatura)
        inicio = np.random.choice(espacobusca)
        x = inicio * 1
        atual = func(x)
        historico = [x]
        for i in range(1000):
            prop = x + np.random.normal()*scale
            if prop > 1 or prop < 0 or np.log(np.random.rand()) * Temperatura > (func(prop) - atual):
                prop = x
            x = prop
            atual = func(x)
            Temperatura = 0.8*Temperatura
            historico.append(x)
        return x, historico
```

E agora nós vamos chamar a função e plotar os resultados (a linha azul é a função e a linha vermelha o caminho que ele fez até chegar no máximo).

```
In [4]: x1, historico = SA(x, funcao, Temperatura = 4)

        plt.plot(x, funcaovetorizada(x))
        plt.scatter(x1, funcao(x1), marker = 'x')
        plt.plot(historico, funcaovetorizada(historico))
```

Out[4]: [

