

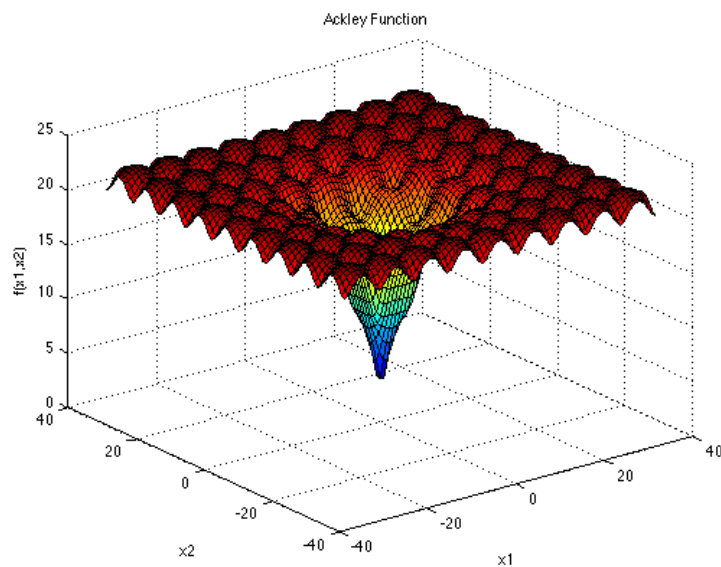
## Aula Prática 4

### Objetivo:

O objetivo desta tarefa é que você se familiarize com a implementação e análise do **ajuste e controle de parâmetros** usando como exemplo um algoritmo evolutivo.

### Descrição:

A função Ackley é amplamente usada para testar algoritmos de otimização. Em sua forma bidimensional, como mostrado na imagem, ela é caracterizada por uma região externa quase plana e um grande buraco no centro. A função representa um risco para algoritmos de otimização, particularmente algoritmos de subida de encosta, de ficarem presos em um de seus muitos mínimos locais.



$$f(\mathbf{x}) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

*Dimensões:*  $d = 30$

Os valores de variáveis recomendados são:  $a = 20$ ,  $b = 0,2$  e  $c = 2\pi$ .

### Domínio de entrada:

A função é geralmente avaliada no hipercubo  $x_i \in [-32.768, 32.768]$ , para todo  $i = 1, \dots, d$ , embora também possa ser restrita a um domínio menor. Utilizar  $-30 \leq x_i \leq 30$

### Mínimo global:

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \dots, 0)$$

O problema que você está tentando resolver é usar um AE para encontrar o mínimo global.

### Algoritmo:

Estratégia de Evolução ( $\mu, \lambda$ )

Critério de Parada do algoritmo: após realizar 200.000 avaliações do fitness

### Ajuste de Parâmetros:

$\mu = \{20, 30, 40\}$

$\lambda = \{100, 200, 300\}$

### Resultados:

- Realizar 100 execuções
- Aplicar um teste estatístico para comparar as diferentes instâncias do algoritmo

### ANOVA



```
python Copiar Editar  
  
import scipy.stats as stats  
  
# Simulação de 9 conjuntos de dados (substitua pelos seus dados reais)  
dados = [  
    [1.2, 1.5, 1.3, 1.4, 1.6, 1.7, 1.8, 1.9, 2.0, 1.5], # Abordagem 1  
    [1.4, 1.6, 1.5, 1.7, 1.9, 2.1, 1.8, 1.5, 1.3, 1.4], # Abordagem 2  
    # ... Adicione os outros 7 conjuntos  
]  
  
estatistica, p_valor = stats.f_oneway(*dados)  
print(f"Estatística F = {estatistica:.4f}, p-valor = {p_valor:.4f}")  
  
if p_valor < 0.05:  
    print("Há diferença significativa entre pelo menos duas abordagens.")  
else:  
    print("Não há diferença significativa entre as abordagens.")
```

Se o **p-valor da ANOVA** for **menor que 0.05**, significa que pelo menos uma abordagem é diferente das outras, mas **não diz qual**. Para identificar quais abordagens são diferentes, usamos um **teste post-hoc**.

## TUKEY HSD

Esse teste faz comparações **par a par**, mas **controla o erro estatístico**.

python

 Copiar  Editar

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import numpy as np

# Criar um array com todas as observações
valores = np.concatenate(dados)
grupos = np.array(["Abordagem 1"]*10 + ["Abordagem 2"]*10 + ...) # Repita para 9 a

tukey = pairwise_tukeyhsd(endog=valores, groups=grupos, alpha=0.05)
print(tukey)
```

Esse teste retorna quais abordagens têm diferenças significativas entre si.