

## 1 Introdução

O objetivo desse laboratório foi implementar um algoritmo de estratégias evolutivas. O algoritmo mais completo de tais estratégias é o *CMA-ES*, o qual converge rapidamente para um ótimo global ou local. No entanto, pela complexidade de tal algoritmo, nesse laboratório, foi escolhido um algoritmo mais simples para se implementar. Após feita a implementação, o objetivo principal foi comparar a convergência do algoritmo mais simples com o *CMA-ES*.

## 2 Código

```
def tell(self, fitnesses):  
    best_samples = melhores vetores em self.sample  
    self.samples=np.random.multivariate_normal(self.m,self.C,self.population_size)  
  
    sum1=1/self.mu*sum(best_samples)  
    sum2=1/self.mu*sum((best_samples-self.m)*(best_samples-self.m).transpose)  
  
    self.m=1/self.mu*sum1  
    self.C=1/self.mu*sum2
```

## 3 Teste das estratégias evolutivas

A partir daqui, este relatório será focado em discutir os resultados obtidos. O programa rodou em várias funções pré-disponibilizadas pelo professor (*translated sphere*, *ackley*, *schaffer2d* e *rastrigin*).

### 3.1 translated sphere

Tanto o algoritmo simples quanto o *CMA-ES* convergiram para o ótimo local (1, 2) (vide a figura 1) . O segundo, como esperado, obteve uma convergência mais rápida para tal valor.

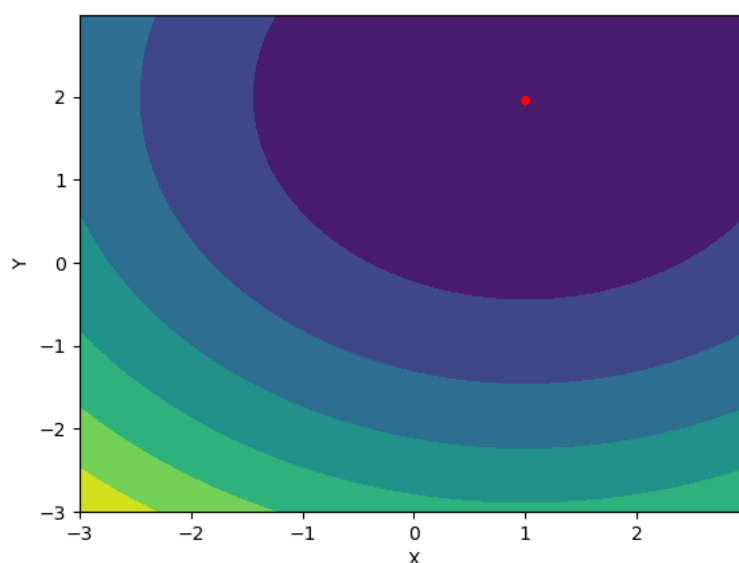


Figura 1: Resultado da otimização da função *translated sphere*. Observe a convergência para o ótimo global

Ao rodar no Monte Carlo, mostrou-se a velocidade de convergência do método mais complexo em comparação ao mais simples, de acordo com as figuras 2 e 3.

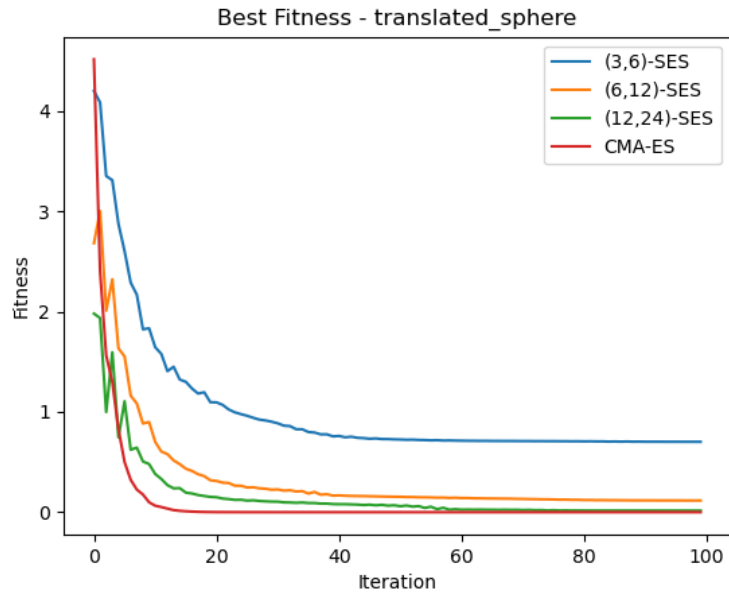


Figura 2: Resultados de *Best* para a primeira função

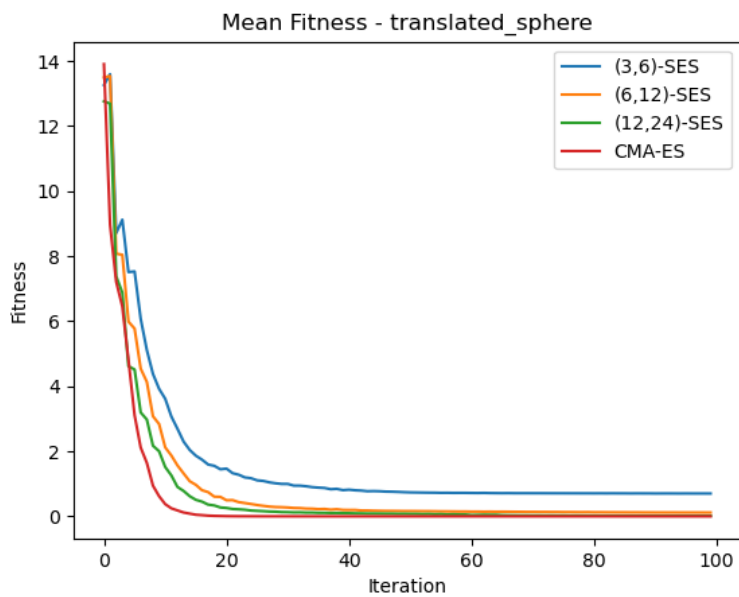


Figura 3: Resultados de *Mean* para a primeira função

### 3.2 ackley

Igualmente ao 3.1, os dois métodos convergiram sem dificuldades para o ótimo global (0,0) (vide a figura 4).

Os resultados do Monte Carlo mostram uma convergência, em geral, mais rápida para o *CMA-ES*. Veja as figuras

### 3.3 schaffer2d

A função *schaffer2d* teve dificuldades para convergir tanto utilizando o algoritmo mais simples quanto com o *CMA-ES*. O resultado do Monte Carlo está na figura 7, 8 e 9

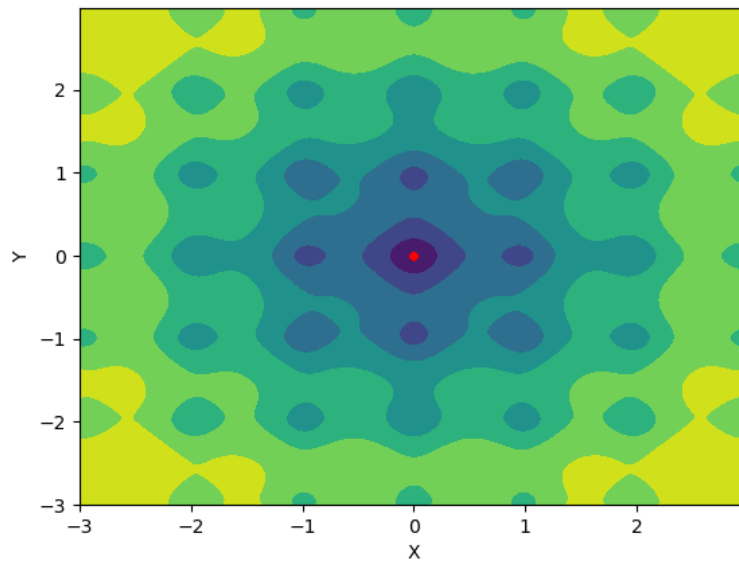


Figura 4: Resultado da otimização para a função *ackley*. Note que a convergência foi para o ótimo global

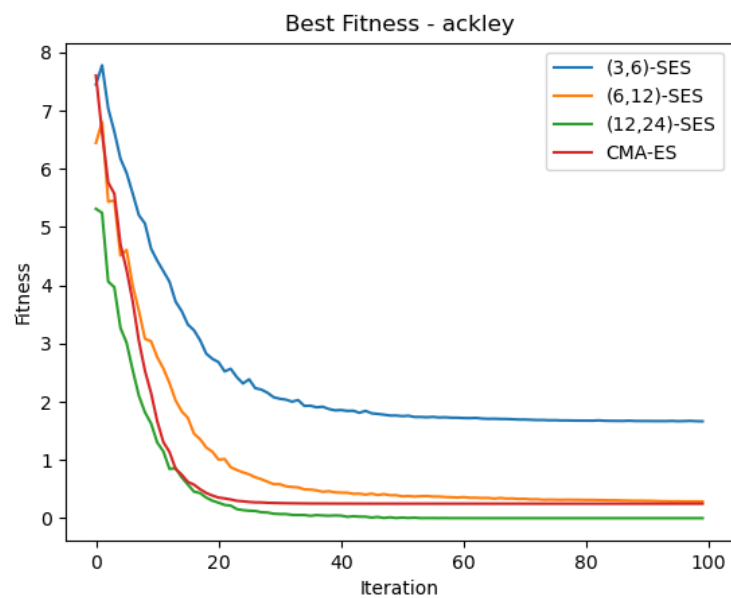


Figura 5: Resultados de *Best* para a função *ackley*

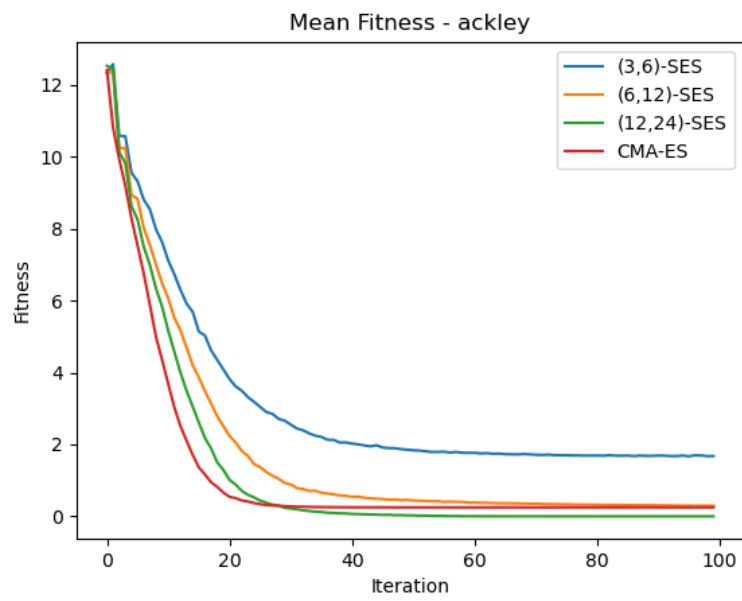


Figura 6: Resultados de *Mean* para a função *ackley*

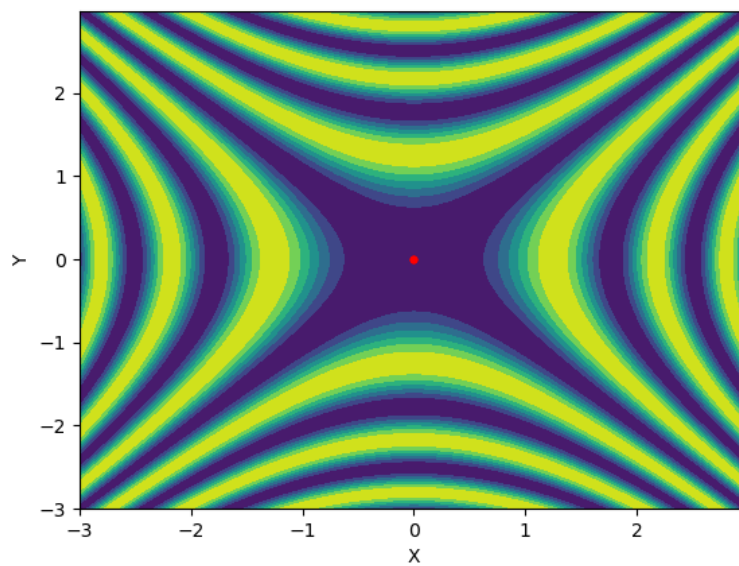


Figura 7: Convergência da otimização para a função *schaffer2d*.

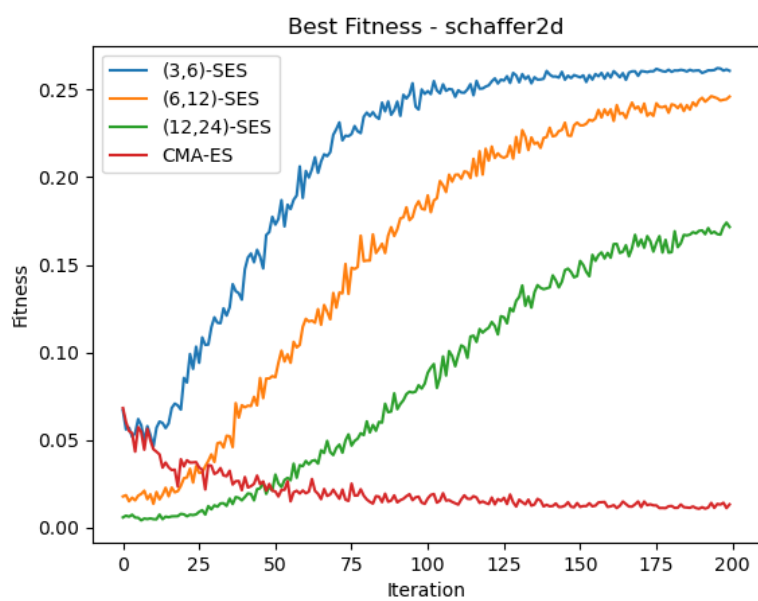


Figura 8: Resultados de *Best* para a função *schaffer2d*

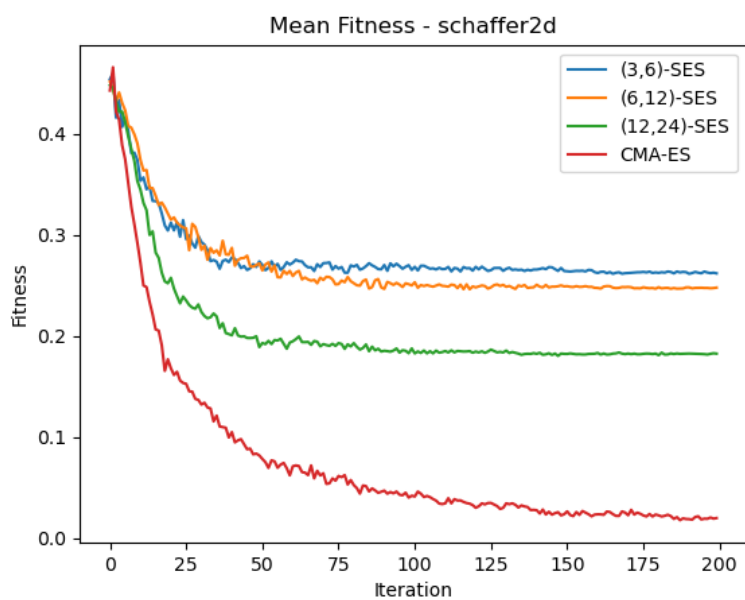


Figura 9: Resultados de *Mean* para a função *schaffer2d*

### 3.4 rastrign

Essa foi a função que convergiu para vários valores diferentes (ótimos locais), como mostram as figuras 10 e 11. Os resultados

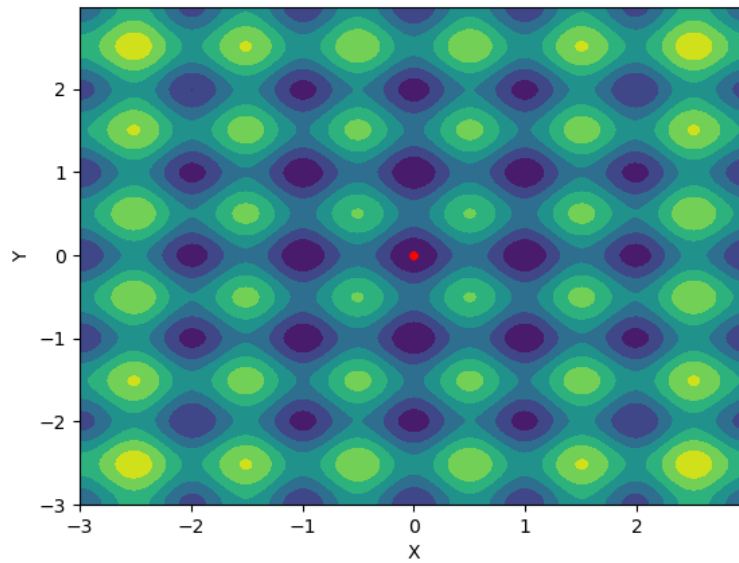


Figura 10: Iteração em que o algoritmo convergiu para o ótimo global (0,0). Obs: foi utilizado o algoritmo simples

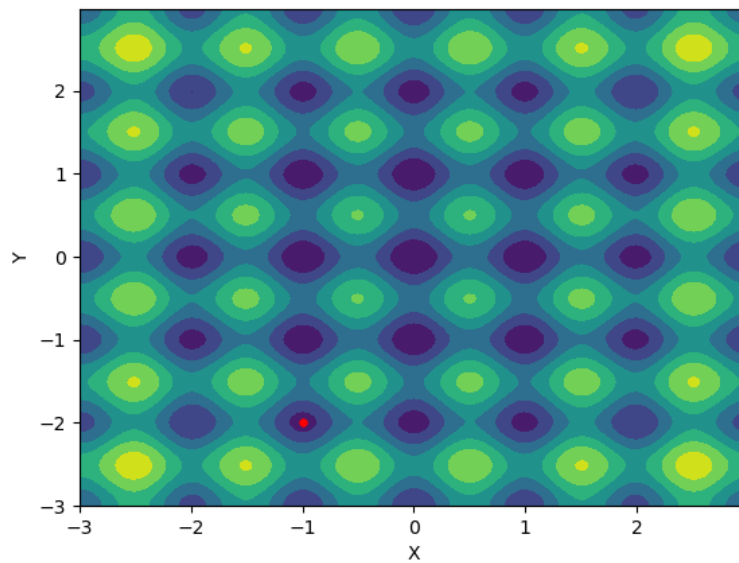


Figura 11: Iteração em que o algoritmo convergiu para um ótimo local. Obs: foi utilizado o algoritmo *CMA-ES*

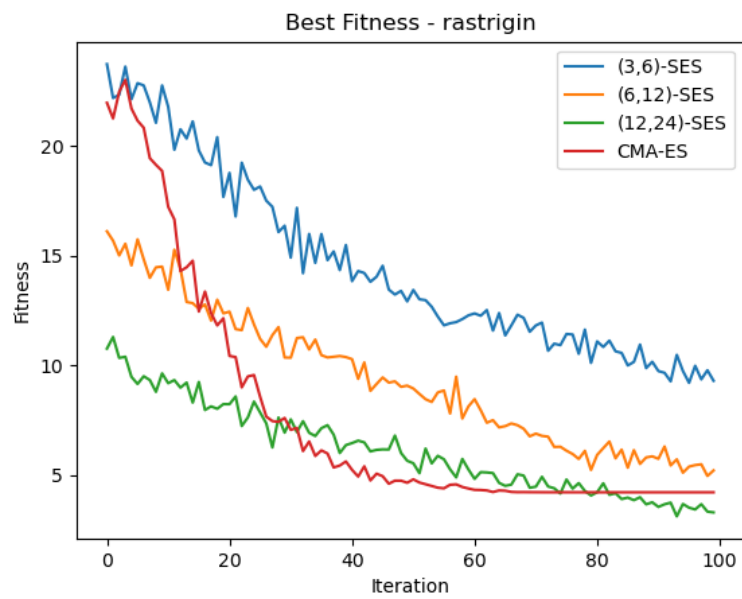


Figura 12: Resultados de *Best* para a função *rastrigin*

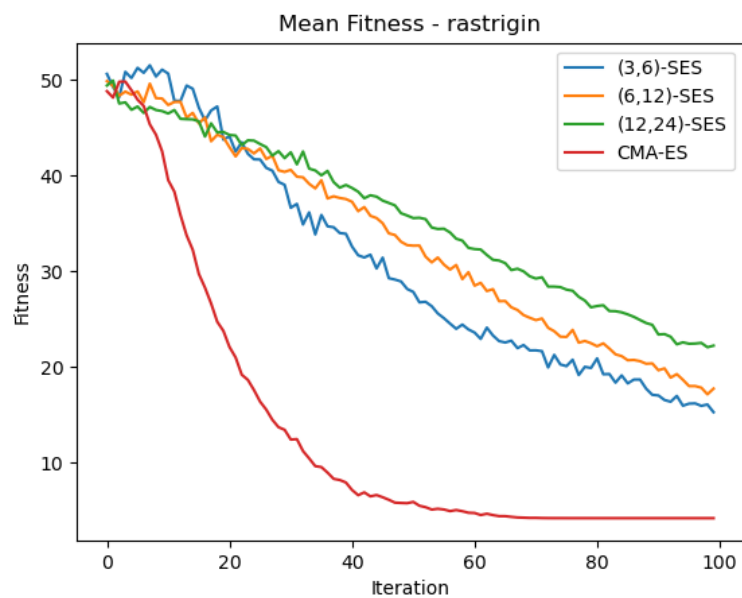


Figura 13: Resultados de *Mean* para a função *rastrigin*