

## Relatório do Laboratório 5 - Estratégias Evolutivas

### 1 Breve Explicação em Alto Nível da Implementação

#### 1.1 Estratégia Evolutiva Simples

Implementação dos passos segundo o orientado pelo roteiro. O método ficou como apresentado a seguir:

```
class SimpleEvolutionStrategy:
    def __init__(self, m0, C0, mu, population_size):
        ...

    def ask(self):
        return self.samples

    def tell(self, fitnesses):
        indices = np.argsort(fitnesses)
        best_samples = self.samples[indices[0:self.mu], :]
        C0 = np.zeros(np.shape(self.C))
        for i in range(0, self.mu):
            sample_error = best_samples[i] - self.m
            sample_error_2d = sample_error[np.newaxis]
            sample_error_t = sample_error_2d.transpose()
            C0 += sample_error_t*sample_error
        self.C = C0/self.mu
        self.m = np.mean(best_samples, axis=0)
        self.samples = np.random.multivariate_normal(self.m,
                                                    self.C, self.population_size)
```

## 2 Figuras Comprovando Funcionamento do Código

### 2.1 Função *Translated Sphere*

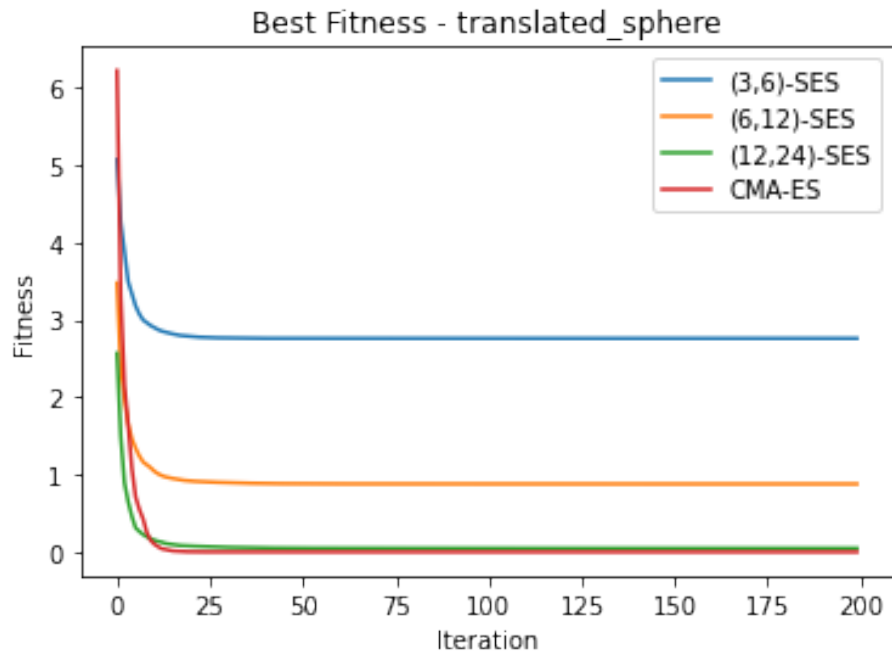


Figura 1: Evolução do melhor fitness, *Translated Sphere*.

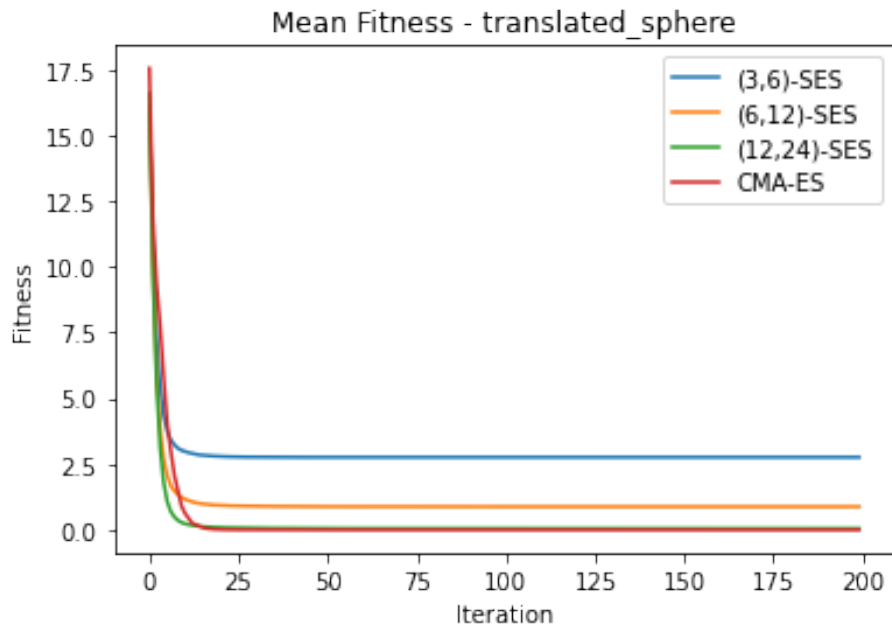


Figura 2: Evolução do fitness, média, *Translated Sphere*.

## 2.2 Função Ackley

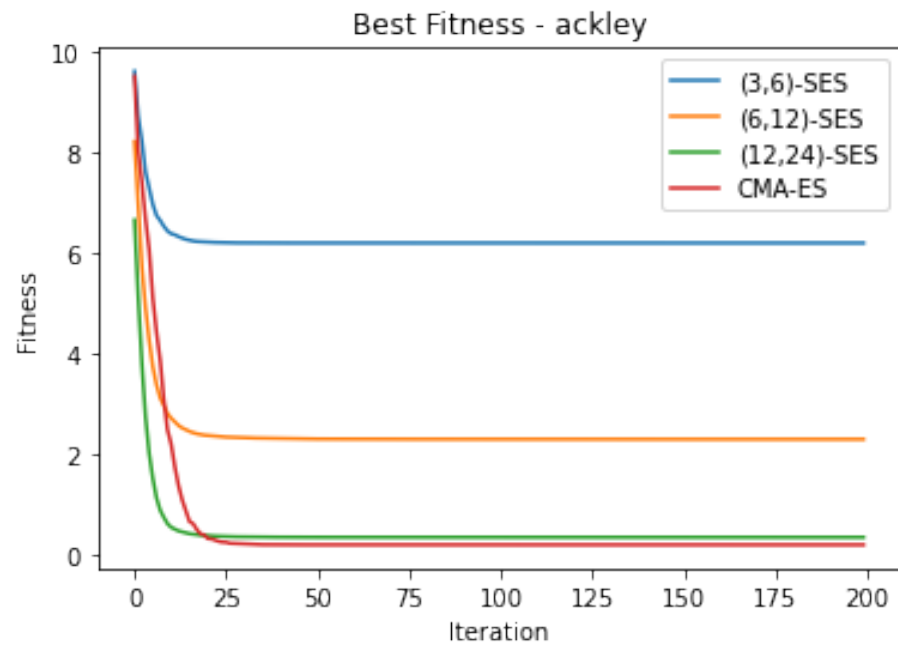


Figura 3: Evolução do melhor fitness, Ackley.

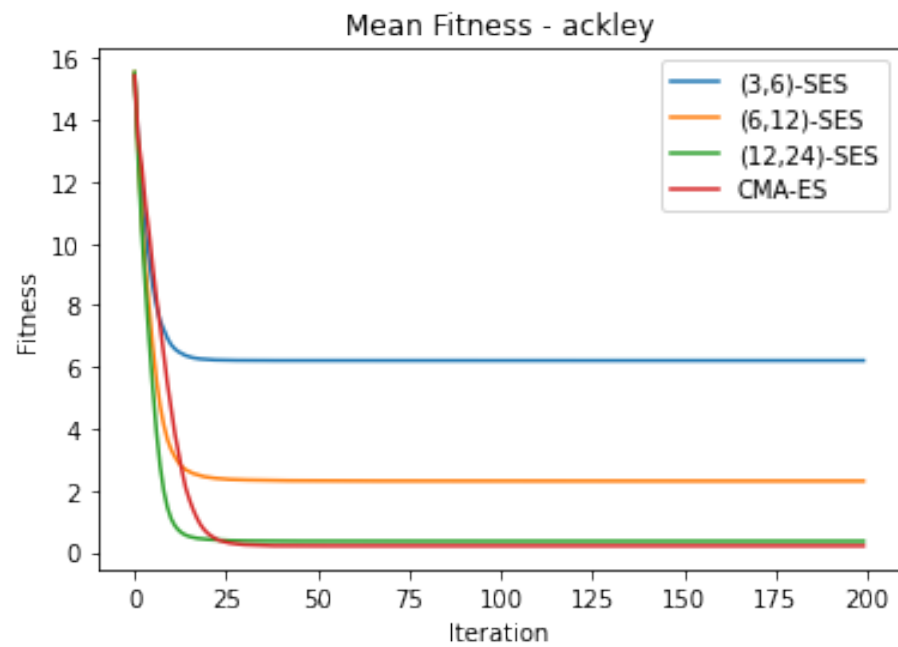


Figura 4: Evolução do fitness, média, Ackley.

## 2.3 Função Rastrigin

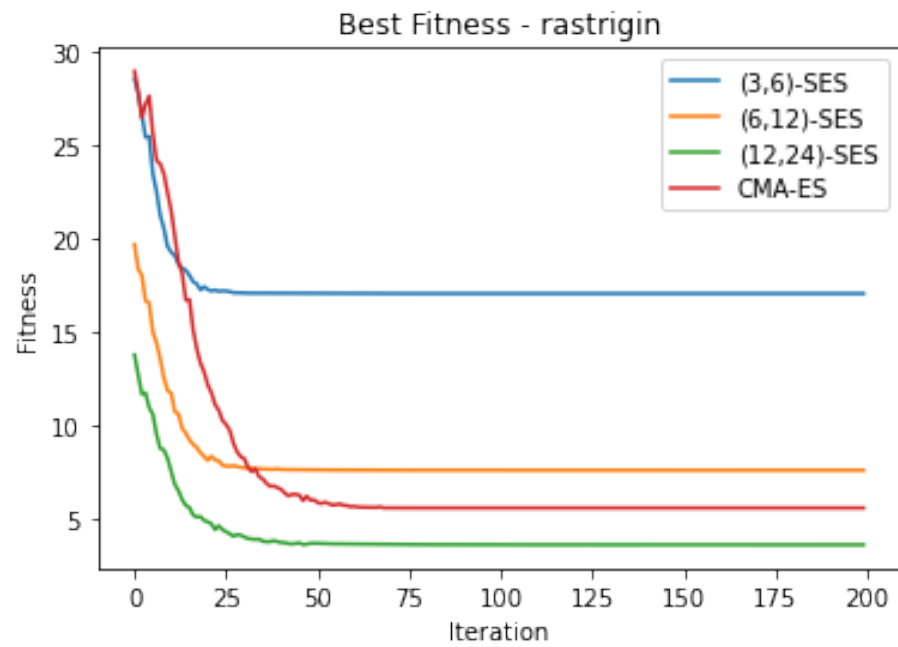


Figura 5: Evolução do melhor fitness, Rastrigin.

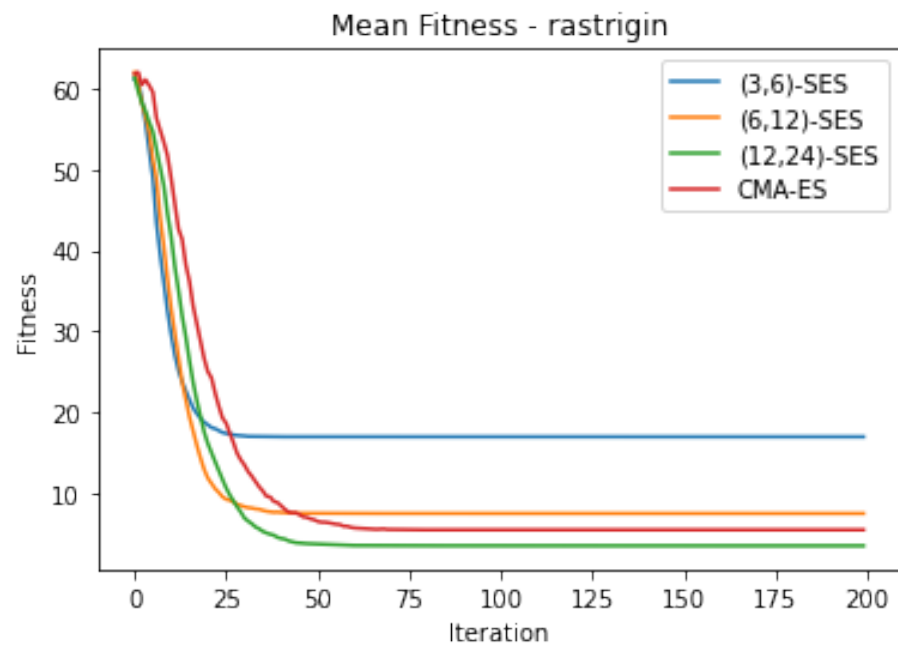


Figura 6: Evolução do fitness, média, Rastrigin.

## 2.4 Função Schaffer

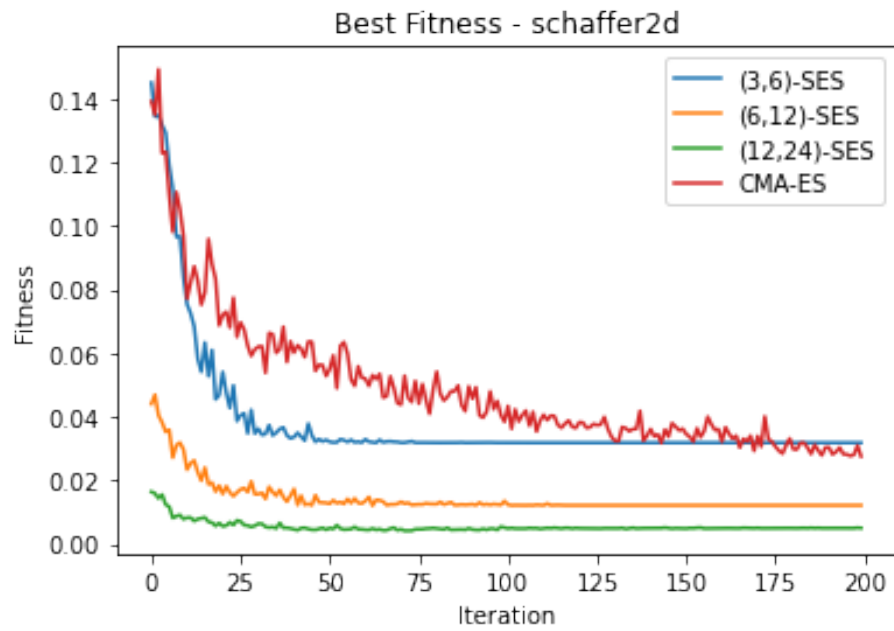


Figura 7: Evolução do melhor fitness, Schaffer.

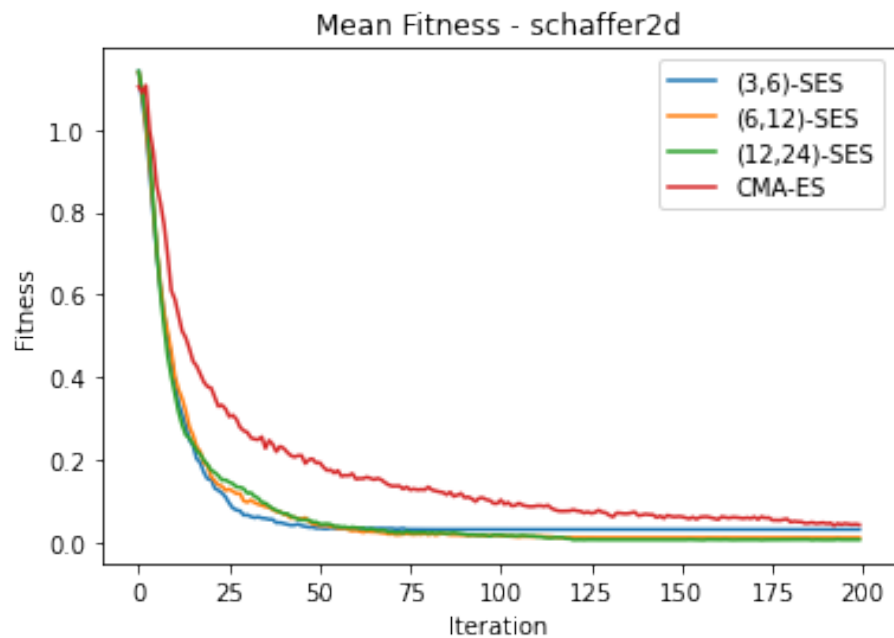


Figura 8: Evolução do fitness, média, Schaffer.

### 3 Discussões e Conclusões

- As funções de benchmark, por serem tão diversas, resultam em percursos diferentes. Por exemplo, as funções apresentadas em sala (Schaffer e Rastrigin) tem comportamentos bem distintos, com a Rastrigin sendo mais suave, enquanto a Schaffer transita de mínimos para máximo muito rapidamente. Além disso, há aleatoriedade no processo, de modo que em diferentes iterações, encontram-se diferentes mínimos locais. A escolha de várias amostragens é a melhor forma de ilustrar como o método em si performa em um caso médio.
- Na função *Translated Sphere*, observou-se convergência em mínimo local em alguns casos para o SES, enquanto o CMA-ES em média chega a um mínimo global.
- Na função Ackley, por sua vez, observam-se dois casos de convergência para mínimos locais no SES, enquanto o CMA-ES, novamente, encontra um mínimo global.
- A função Rastrigin tem resultados mais interessantes. Observam-se dois fatos importantes. Um deles é que, a depender do chute inicial, o SES pode convergir para um mínimo melhor que o encontrado pelo CMA-ES. O outro é que o próprio CMA-ES já observa dificuldades em encontrar um mínimo global para sua convergência.
- A função Schaffer, de picos e vales muito tortuosos e com mudanças bruscas, trás como resultados o funcionamento do SES melhor que o do CMA-ES. Em todo caso, observa-se que todas as funções (mesmo de diferentes pontos iniciais) chegam em mínimos locais bem próximos, indicando que nesse caso estudado e na dimensionalidade proposta, os algoritmos mais simples performaram melhor.