

Inteligência Artificial para Robótica Móvel

CT-213

Instituto Tecnológico de Aeronáutica

Relatório do Laboratório 5 - Estratégias Evolutivas

Leonardo Peres Dias

2 de maio de 2025





Sumário

1 Breve explicação em alto nível da implementação	3
1.1 Estratégia Evolutiva Simples	3
1.1.1 Inicialização	3
1.1.2 Procedimento ask	3
1.1.3 Procedimento tell	3
2 Figuras comprovando funcionamento do código	4
2.1 Função Translated Sphere	4
2.2 Função Ackley	5
2.3 Função Rastrigin	6
2.4 Função Schaffer	7
3 Discussão e Conclusões	7

1 Breve explicação em alto nível da implementação

1.1 Estratégia Evolutiva Simples

1.1.1 Inicialização

No construtor, os parâmetros de entrada são:

- \mathbf{m}_0 : média inicial da distribuição.
- \mathbf{C}_0 : matriz de covariância inicial.
- μ : número de pais selecionados por geração.
- λ : tamanho da população de amostras.

As amostras iniciais são geradas por

$$\{\mathbf{x}_i\}_{i=1}^{\lambda} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{C}_0).$$

1.1.2 Procedimento ask

A chamada $\text{ask}()$ simplesmente retorna o conjunto de amostras

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{\lambda} \end{bmatrix} \sim \mathcal{N}(\mathbf{m}_t, \mathbf{C}_t),$$

que será avaliado pela função de fitness externa.

1.1.3 Procedimento tell

Dado o vetor de fitnesses ordenados, selecionam-se os índices dos μ melhores indivíduos. Seja

$$\{\mathbf{y}_j\}_{j=1}^{\mu}$$

o subconjunto de amostras com menores valores de fitness. Definem-se então:

$$\mathbf{m}_{t+1} = \frac{1}{\mu} \sum_{j=1}^{\mu} \mathbf{y}_j, \quad \mathbf{C}_{t+1} = \frac{1}{\mu} \sum_{j=1}^{\mu} (\mathbf{y}_j - \mathbf{m}_t)(\mathbf{y}_j - \mathbf{m}_t)^{\top}.$$

Por fim, as amostras da próxima geração são novamente extraídas de

$$\mathcal{N}(\mathbf{m}_{t+1}, \mathbf{C}_{t+1}).$$

2 Figuras comprovando funcionamento do código

2.1 Função Translated Sphere

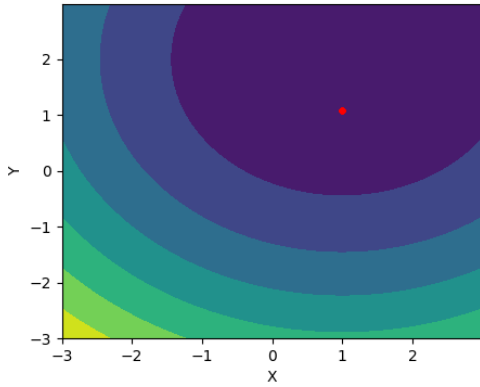


Figura 1: Otimização com SES

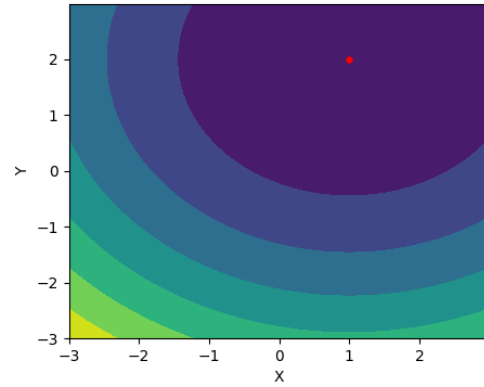


Figura 2: Otimização com CMA-ES

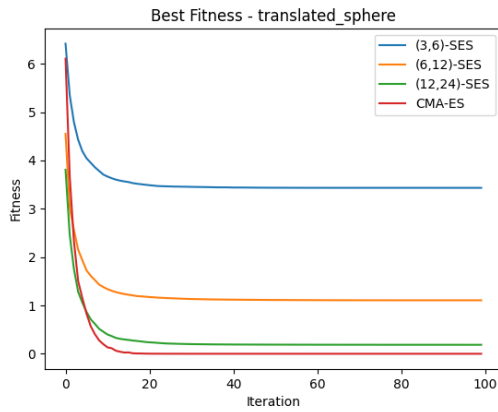


Figura 3: Evolução do melhor fitness no benchmark

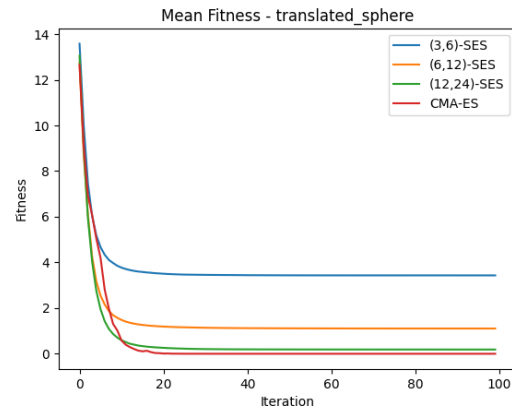


Figura 4: Evolução da média do fitness no benchmark

2.2 Função Ackley

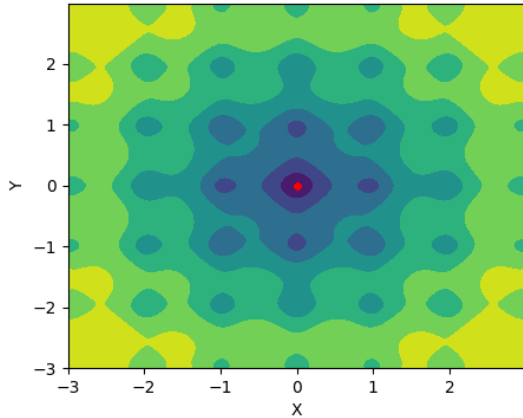


Figura 5: Otimização com SES

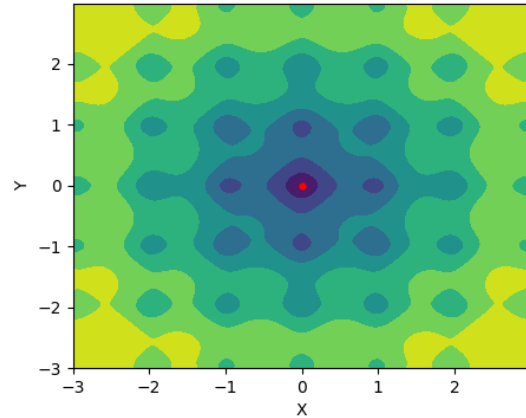


Figura 6: Otimização com CMA-ES

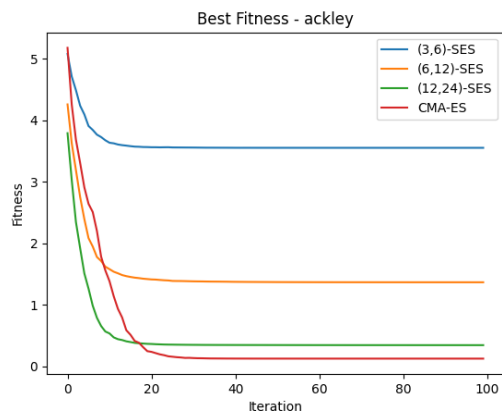


Figura 7: Evolução do melhor fitness no benchmark

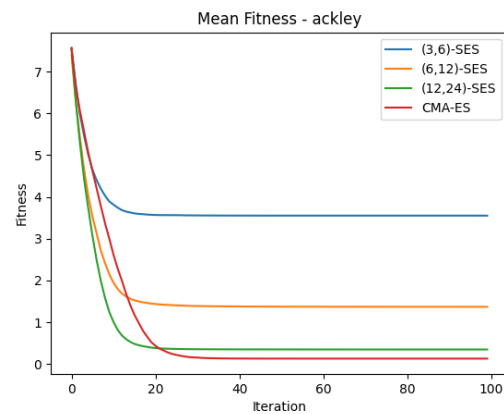


Figura 8: Evolução da média do fitness no benchmark

2.3 Função Rastrigin

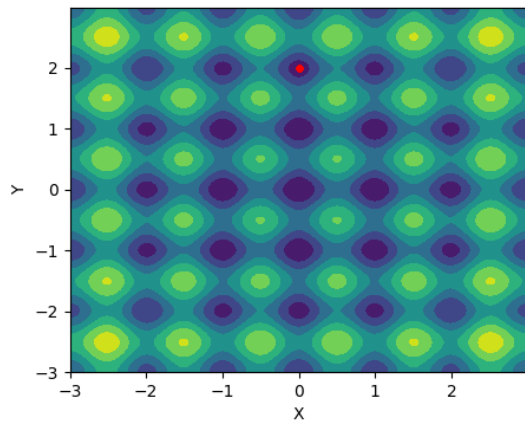


Figura 9: Otimização com SES

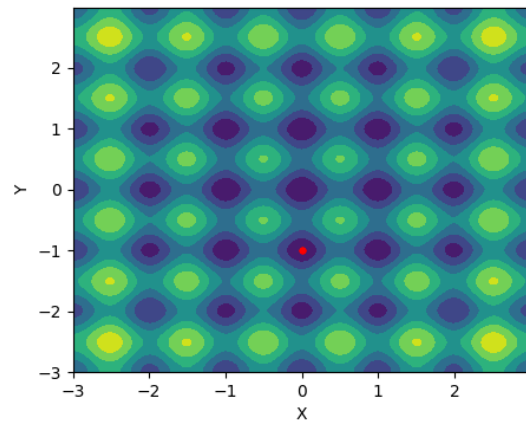


Figura 10: Otimização com CMA-ES

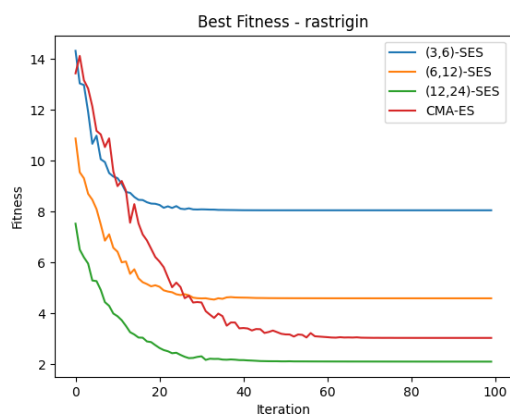


Figura 11: Evolução do melhor fitness no benchmark

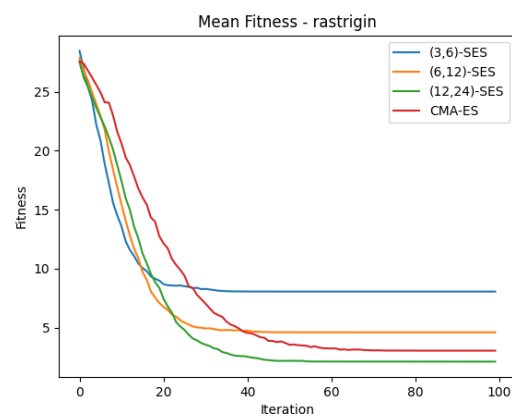


Figura 12: Evolução da média do fitness no benchmark

2.4 Função Schaffer

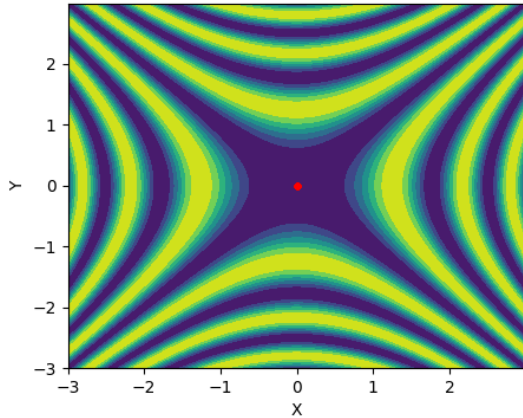


Figura 13: Otimização com SES

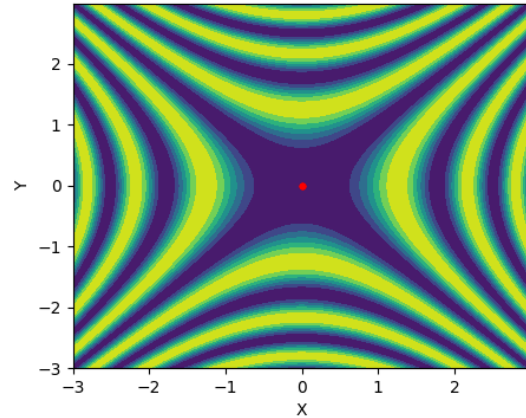


Figura 14: Otimização com CMA-ES

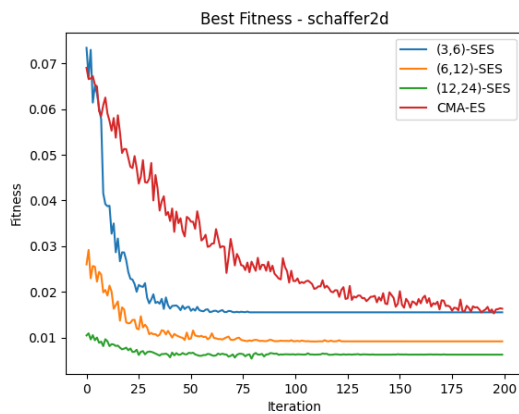


Figura 15: Evolução do melhor fitness no benchmark

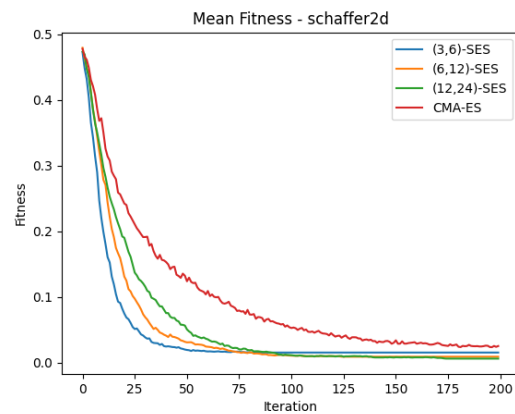


Figura 16: Evolução da média do fitness no benchmark

3 Discussão e Conclusões

- Pode-se observar que para as funções **Translated Sphere**, **Ackley** e **Schaffer**, tanto o SES quanto o CMA-ES convergiram para o mínimo global. Já no caso da função **Rastrigin**, os diferentes algoritmos convergiram para diferentes mínimos locais.
- De forma resumida, os gráficos do benchmark mostram que:



- Em funções convexas e bem comportadas (Translated Sphere, Schaffer), até o SES com população maior converge rapidamente e próximo ao mínimo global, com desempenho semelhante ao CMA-ES.
- Em funções com muitos mínimos locais (Ackley, Rastrigin), o SES tende a estagnar em mínimos locais, especialmente com populações pequenas, enquanto o CMA-ES atinge valores de fitness muito inferiores.