

Laboratório 5 - Estratégias Evolutivas

Marcelo Buga Martins da Silva

CT-213 - Professor Marcos Ricardo Omena de Albuquerque Máximo

21/04/2021



1 Implementação da Estratégia Evolutiva Simples

Nesse laboratório foi implementado um algoritmo evolutivo simples, *Simple Evolution Strategy* (SES), utilizado para otimização de funções mais complexas. A implementação desse algoritmo, basicamente, consiste em criar uma série de amostras a partir de uma distribuição gaussiana multivariada que tem sua média (m) e matriz de covariância (C) atualizadas a cada interação, de forma que o algoritmo converge para um mínimo (não necessariamente o mínimo global). As fórmulas utilizadas para a atualização da média e da matriz de covariância na geração $g + 1$ são apresentadas nas Equações 1 e 2, em que μ é o número de melhores amostras escolhidas e λ é o tamanho da população, ambos hiperparâmetros.

$$m^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} s_{i:\lambda}^{(g+1)} \quad (1)$$

$$C^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} \left(s_{i:\lambda}^{(g+1)} - m^{(g)} \right) \left(s_{i:\lambda}^{(g+1)} - m^{(g)} \right)^T \quad (2)$$

Para as operações com vetores e matrizes, foi utilizada a biblioteca *NumPy*, que torna mais intuitiva as manipulações. Uma pequena especificidade no código é que, diferentemente da matemática em que os vetores

são coluna por padrão, para *NumPy* os vetores são em linha, de forma que a ordem da multiplicação na Equação 2 é invertida.

2 Teste das Estratégias Evolutivas

Após a implementação do SES, prosseguiu-se para testar esse algoritmo, assim como o já implementado CMA-ES para diferentes funções, sendo elas Translated Sphere, esfera centrada em $(1,2,3,\dots,n)$ sendo n a dimensão do problema e as funções de Ackley, Schaffer e Rastrigin, todas em 2D.

Para esses problemas, para SES foi utilizado $\mu = 12$ e $\lambda = 24$ e para CMA-ES foi utilizado $\mu = 3$ e $\lambda = 6$. Cada uma das funções e a performance de cada algoritmo é apresentada a seguir:

2.1 Translated Sphere

Para essa função, ambos algoritmos convergiram muito rapidamente na maioria dos casos, tendo sido a maior diferença entre eles a dispersão das amostras, que pareceu diminuir mais rapidamente para o algoritmo SES em relação ao CMA-ES. A maioria dos testes convergiu para o resultado esperado $(1,2)$, como apresentado pelas Figura 1.

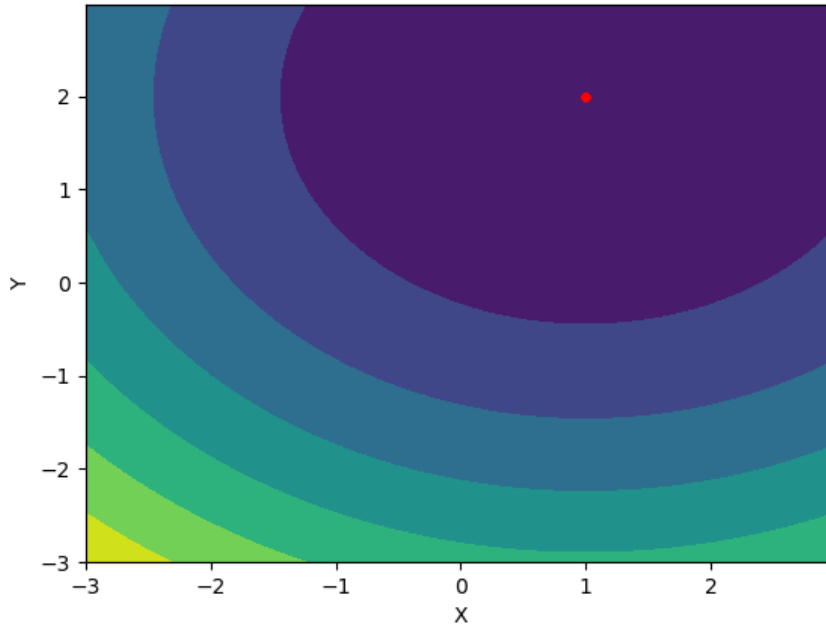


Figura 1: Resultado final para minimização de Translated Sphere via CMA-ES

Para essa função, porém, o algoritmo SES teve alguns (raros) casos em que convergiu para um ponto que

claramente não é um mínimo (nem mesmo local, dado que essa função configura um paraboloide em que o único mínimo local é, também, global). Esse comportamento atípico é apresentado na Figura 2

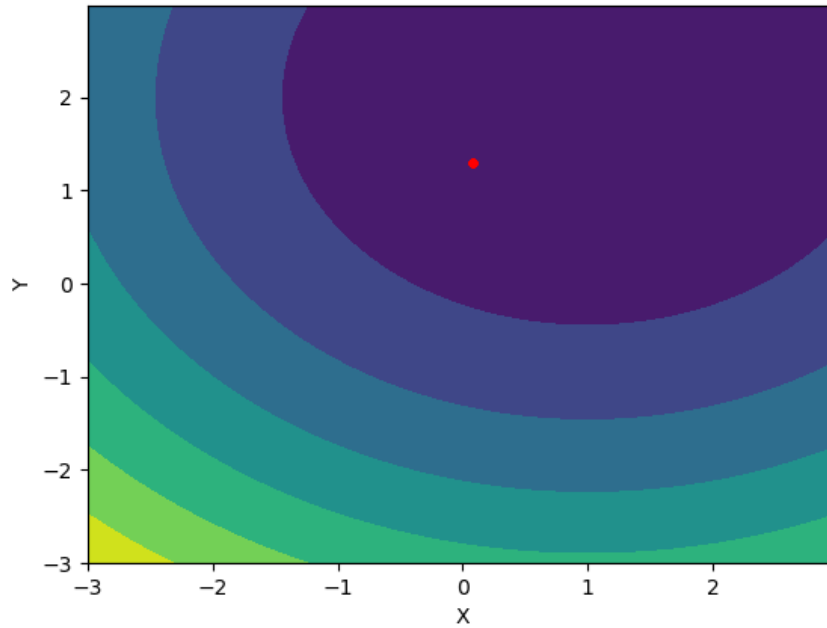


Figura 2: Resultado atípico para o algoritmo SES

2.2 Ackley

Para essa função ocorreu algo muito semelhante ao que ocorreu em Translated Sphere. Na grande maioria das vezes, ambos convergiram para o mínimo global da função em $(0,0)$. Entretanto, em geral, a convergência foi mais rápida para o método SES implementado. Os resultados mais comuns são apresentados na Figura 3

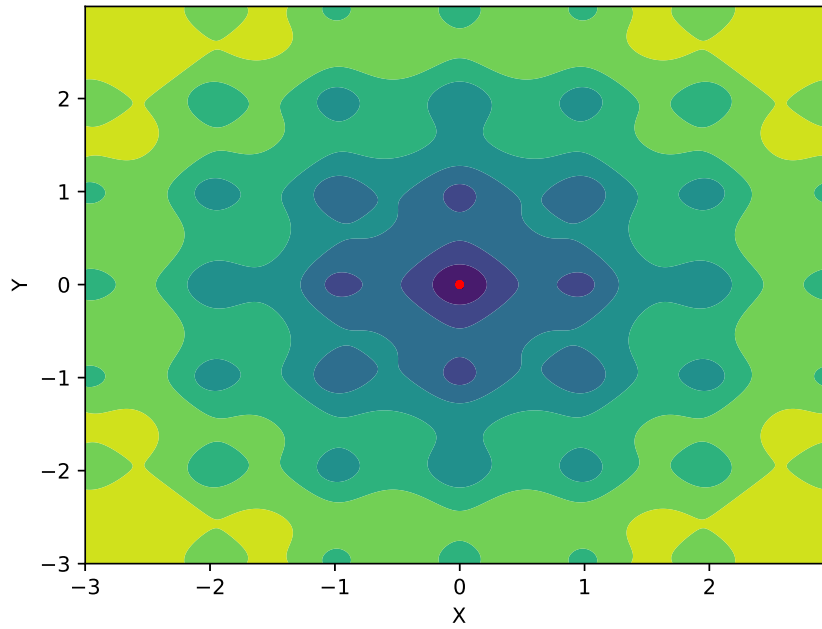


Figura 3: Resultado final mais comum para minimização de Ackley via CMA-ES

Nessa função, ambos algoritmos tiveram muito raras falhas, convergindo para um mínimo local. Isso provavelmente se dá por uma estimativa inicial distante da origem que acabou causando a convergência para mínimos local. Exemplos dessa convergência atípica são apresentados nas Figuras 5 e ??:

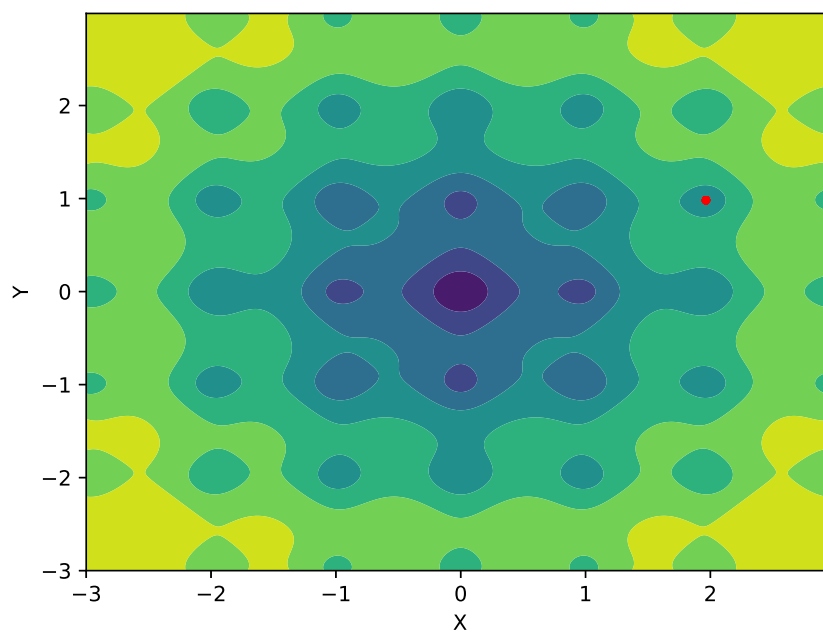


Figura 4: Convergência para mínimo local de Ackley via SES

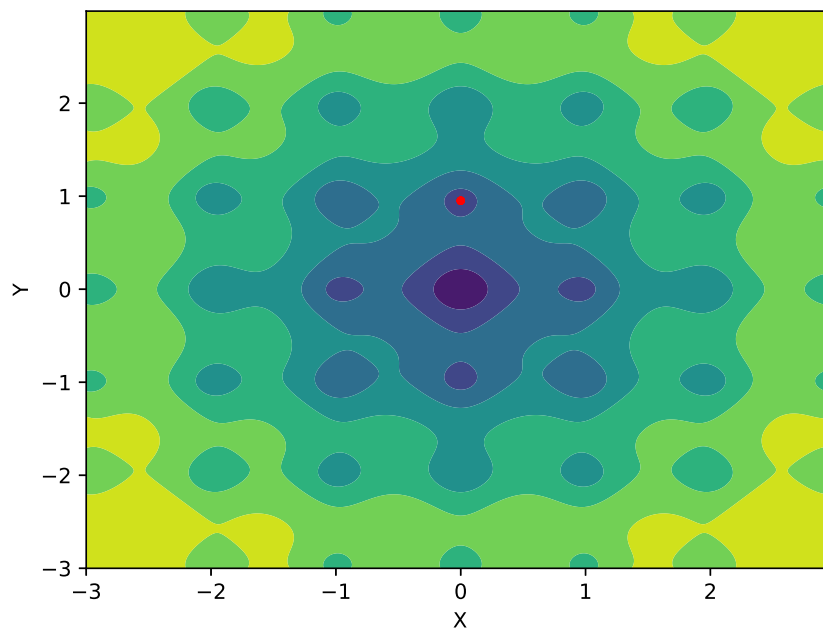


Figura 5: Convergência para mínimo local de Ackley via CMA-ES

2.3 Schaffer

Para essa função, a convergência para o mínimo global em $(0,0)$ foi bem mais rara para ambos os algoritmos. Convergência para mínimos locais foi muito comum e, em algumas situações, a convergência foi externa ao quadrado apresentado nas Figuras, provavelmente devido a estimativa inicial estar mais próxima da “borda” do quadrado. Ainda assim, algumas vezes o resultado convergiu para o mínimo global para ambos os algoritmos, como evidenciado na Figura 6. Para os casos em que o algoritmo convergiu para uma coordenada interna ao quadrado, essa convergência pareceu mais rápida via SES.

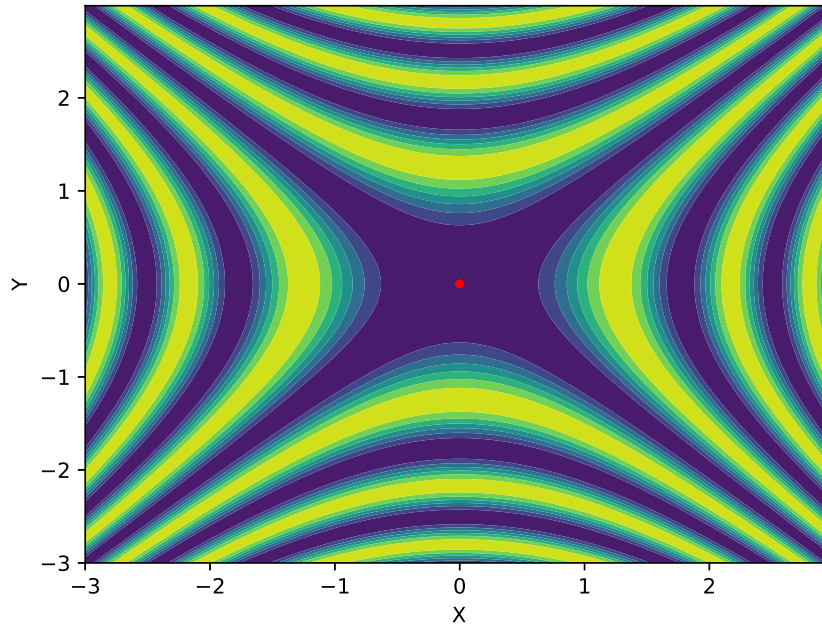


Figura 6: Obtenção do mínimo global de Schaffer 2D via SES

Os casos de convergência para mínimos locais são exemplificados pela Figura 7

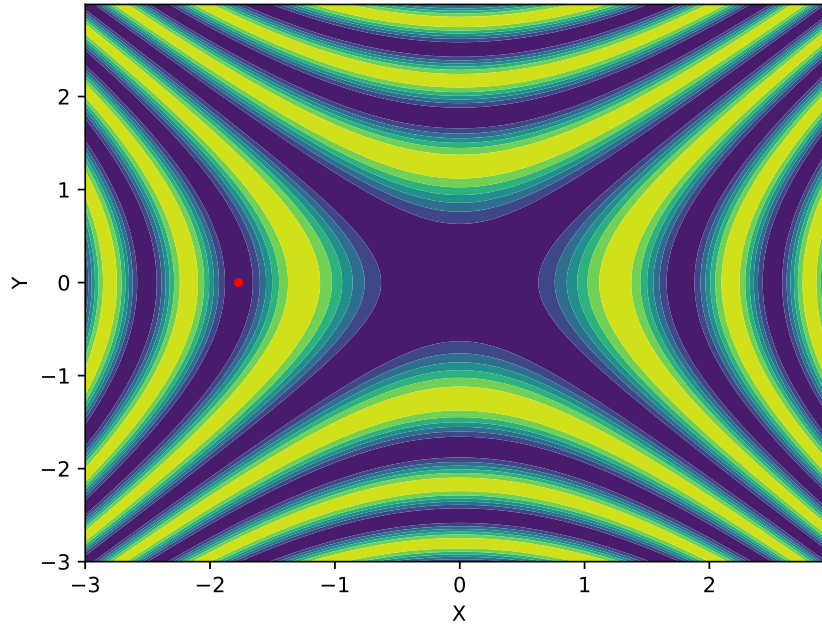


Figura 7: Convergência para um mínimo local em Schaffer2D via CMA-ES

2.3.1 Rastrigin

Os resultados para a função de Rastrigin foram muito dependentes das posições iniciais das amostras, sendo que, para ambos os métodos, foi muito comum a convergência para o mínimo local mais próximo da maioria das amostras. Novamente, nessa função, SES pareceu convergir mais rápido. Nenhum dos algoritmos pareceu mais favorável a encontrar o mínimo global em $(0,0)$. A convergência para o mínimo global foi possível nos dois algoritmos, representada pela Figura 8. Uma convergência para mínimo local é apresentada pela Figura 9

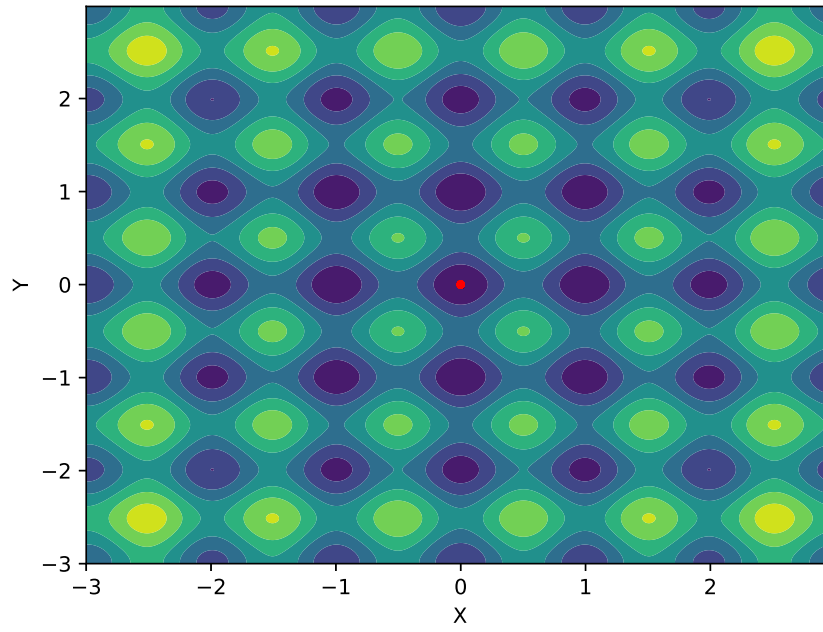


Figura 8: Convergência para mínimo global em Rastrigin via SES

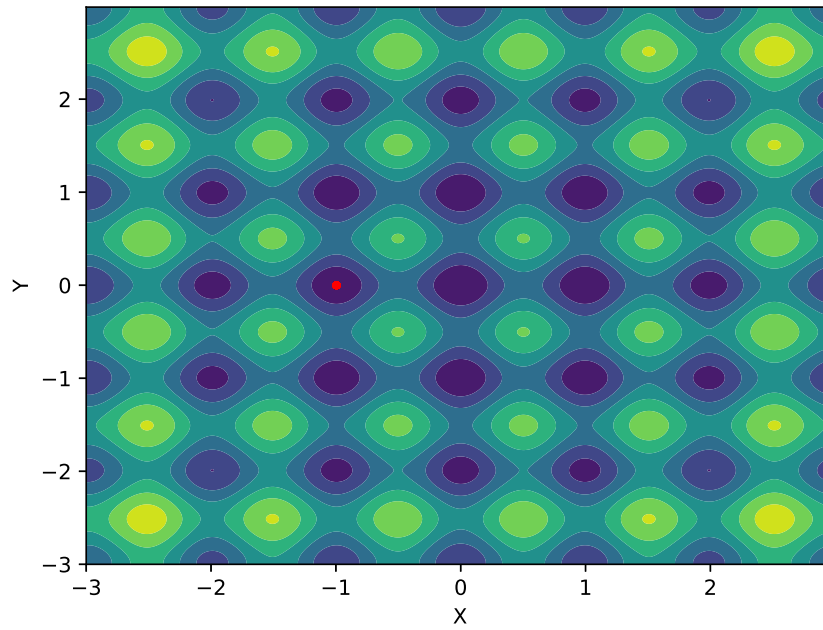


Figura 9: Convergência para mínimo local em Rastrigin via CMA-ES

3 Benchmark

Para de fato comparar os algoritmos, foi feito um *Benchmark* usando simulações de Monte Carlo. Dessa forma, para cada função, comparou-se o *Fitness* médio das amostras em cada geração e o melhor *Fitness* para cada geração. Foram, então, feitos gráficos para avaliar as diferenças. Para essa comparação, foram testados:

- (3,6)-SES: estratégia evolutiva simples com $\mu = 3$ e $\lambda = 6$.
- (6,12)-SES: estratégia evolutiva simples com $\mu = 6$ e $\lambda = 12$.
- (12,24)-SES: estratégia evolutiva simples com $\mu = 12$ e $\lambda = 24$.
- CMA-ES: CMA-ES com estratégia padrão, que usa $\mu = 3$ e $\lambda = 6$.

Foram utilizados 200 *trials* e 100 iterações para cada função, com exceção da função Schaffer 2D, em que, por conta dos ruídos iniciais, foram utilizadas 500 *trials* e 200 iterações. Os resultados para *Fitness* médio são apresentados pelas Figuras 10, 11, 12 e 13.

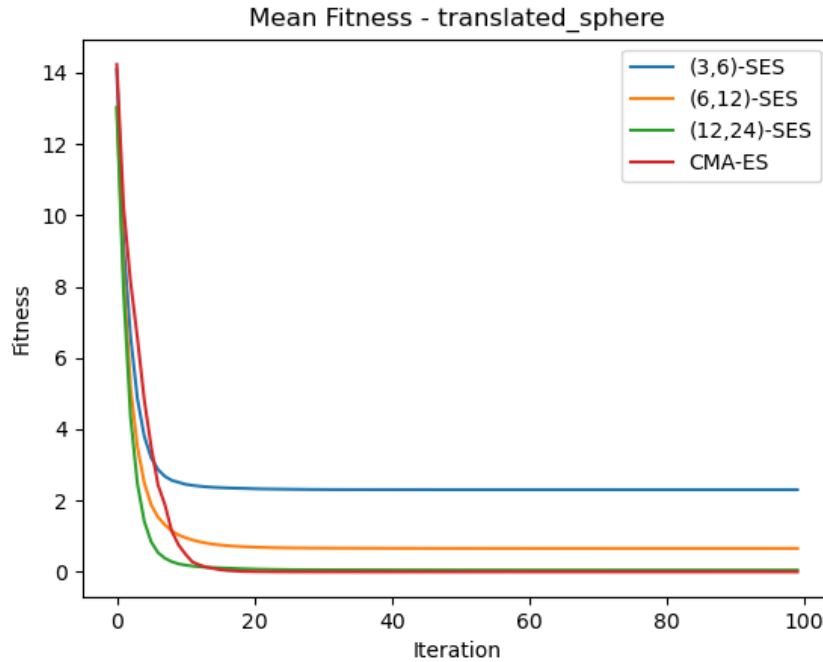


Figura 10: *Fitness* médio por geração de Translated Sphere

Pelos resultados do *Fitness* médio, pode-se perceber que CMA-ES tende a ser melhor que (3,6) -SES e (6,12)-SES e chegando a resultados próximos de (12,24)-SES, apenas convergindo um pouco mais lentamente. Entretanto, para a função de Schaffer, CMA-ES tem a pior performance. Essa análise é confirmada pelos gráficos de melhor *Fitness*, apresentados pelas Figuras 14, 15, 16 e 17.

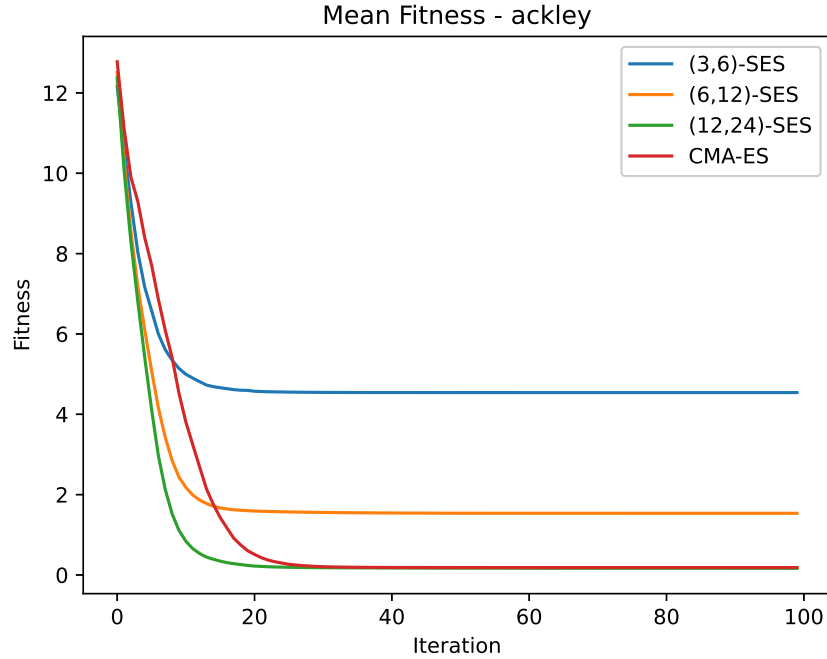


Figura 11: *Fitness* médio por geração de Ackley

Em geral, percebe-se que CMA-ES realmente tende a ser um algoritmo muito superior ao algoritmo evolutivo mais simples implementado, uma vez que para este ter um resultado melhor que CMA-ES, foi necessário que seus hiperparâmetros fossem 4 vezes maiores do que os utilizados em CMA-ES. Um possível motivo para o pior desempenho de CMA-ES para essa função em específico pode ser o fato desse algoritmo considerar várias gerações anteriores e, como essa função tem uma grande amplitude próxima à origem, considerar valores com *Fitness* muito altos pode atrapalhar a convergência. Isso justificaria o comportamento mais ruidoso da curva presente na Figura 16.

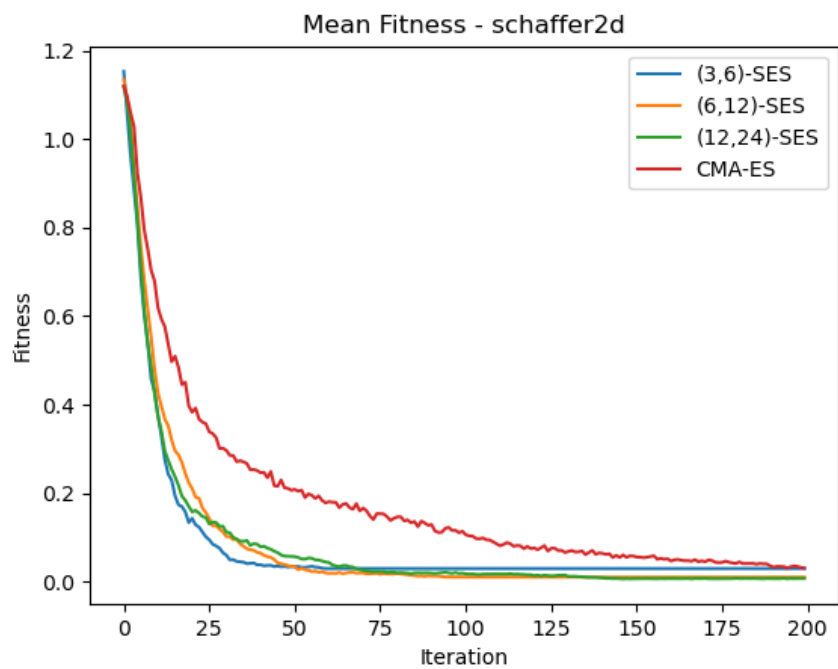


Figura 12: *Fitness* médio por geração de Schaffer

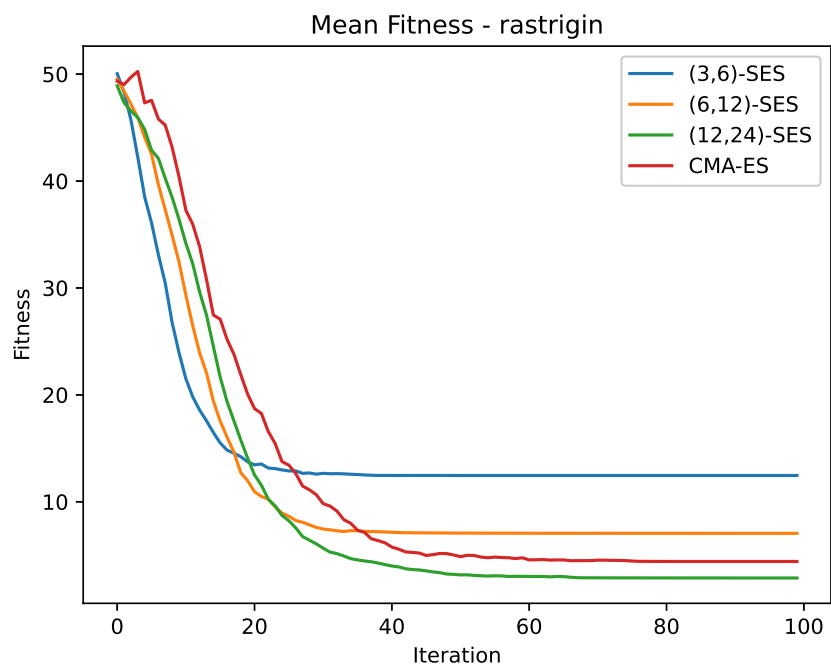


Figura 13: *Fitness* médio por geração de Rastrigin

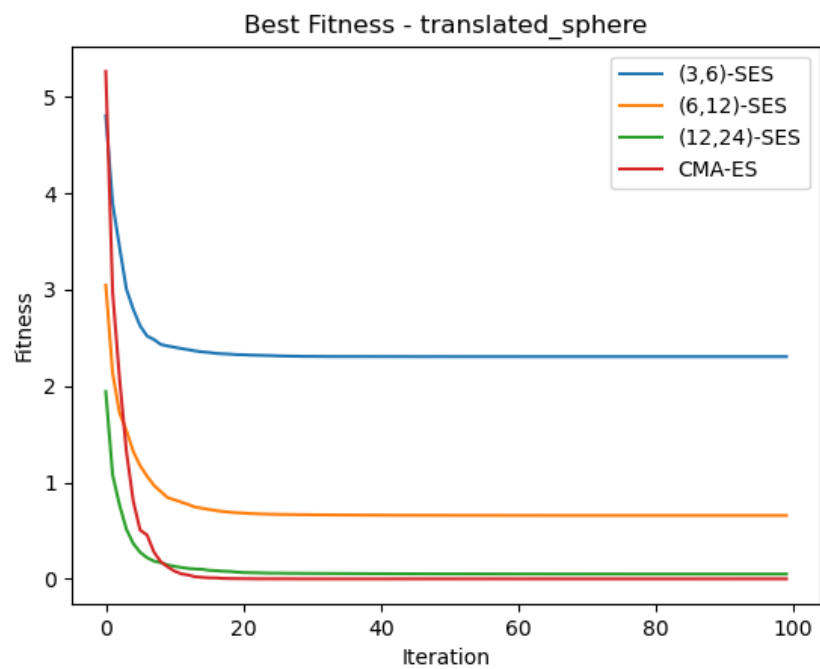


Figura 14: Melhor *Fitness* por geração de Translated Sphere

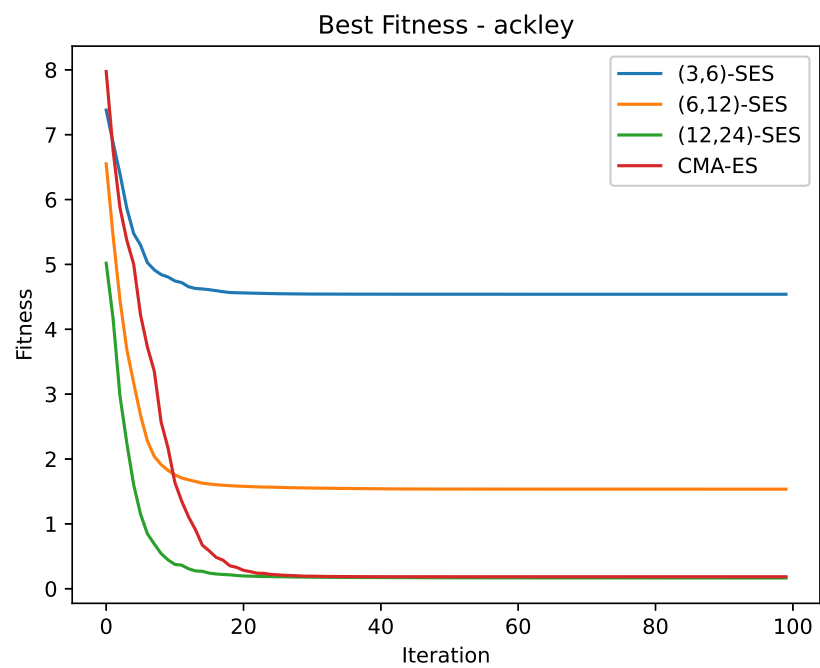


Figura 15: Melhor *Fitness* por geração de Ackley

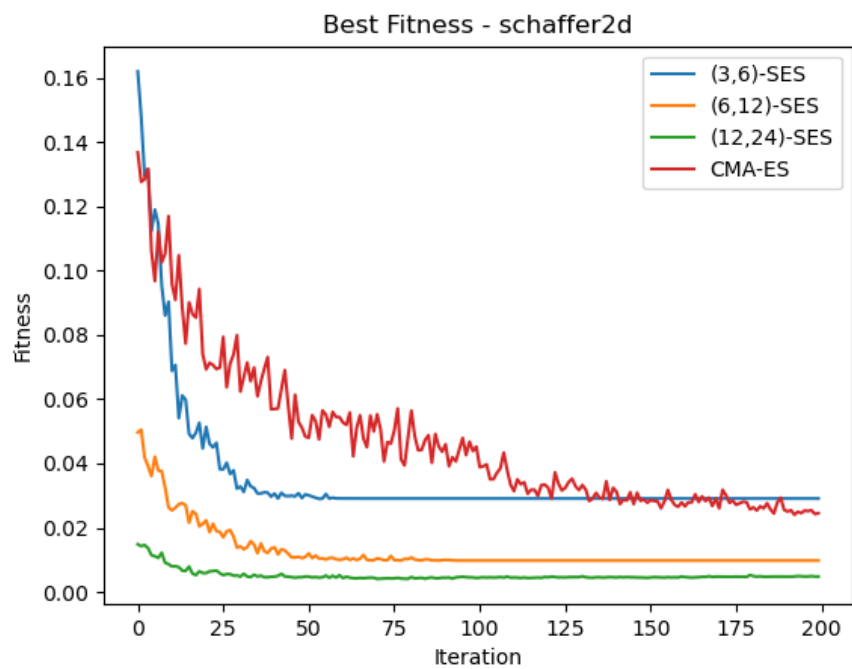


Figura 16: Melhor *Fitness* por geração de Schaffer

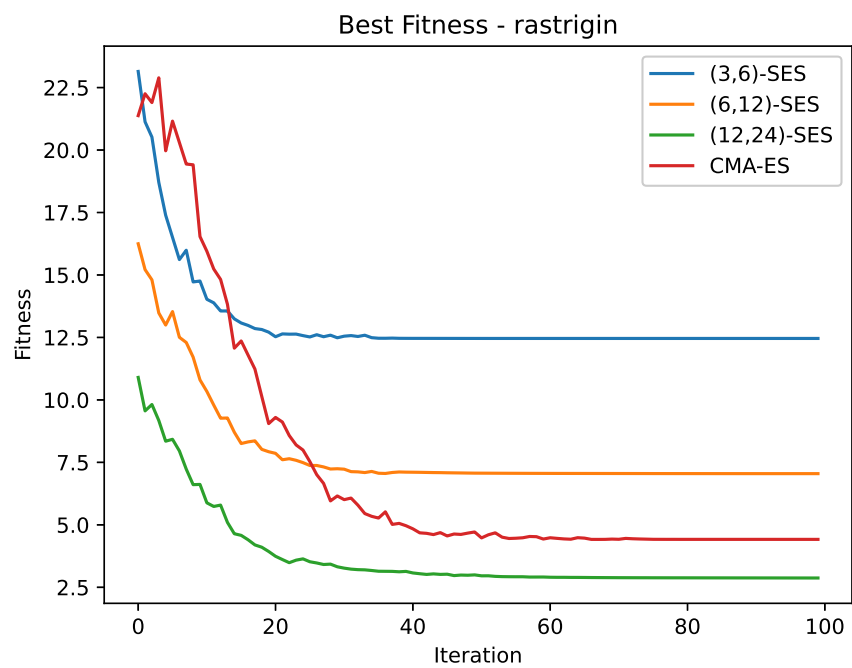


Figura 17: Melhor *Fitness* por geração de Rastrigin