



Relatório do Lab 5 de CT-213

Trabalho 5 – Otimização com Estratégias Evolutivas **Comparando Estratégias Evolutivas simples com o CMA-ES** **(*Covariance Matrix Adaptation – Evolution Strategy*) em** **funções *benchmark***

Aluno:

Bruno Costa Alves Freire

Turma:

T 22.4

Professor:

Marcos Ricardo Omena de Albuquerque Máximo

Data:

28/04/2019

Instituto Tecnológico de Aeronáutica – ITA
Departamento de Computação

1. Implementação da Estratégia Evolutiva Simples (SES)

A implementação da estratégia simples foi baseada no roteiro. A classe `SimpleEvolutionStrategy` funciona armazenando as λ amostras em cada geração, bem como os valores de média e covariância das μ melhores amostras. Seus métodos `ask` e `tell` fazem a interface com o programa cliente, sendo o primeiro responsável por retornar ao cliente as amostras a serem avaliadas com a função de *fitness*. O segundo recebe como parâmetro um *array* com os valores da função de *fitness* ordenados segundo o *array* de amostras retornado em `ask`.

No método `tell` é feita uma ordenação das amostras segundo o array de fitnesses, e em seguida são tomadas as μ melhores amostras, as quais são usadas para calcular a nova matriz de covariância segundo a equação:

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

Onde $s_{i:\lambda}^{(g+1)}$ é a i -ésima melhor amostra da geração $g+1$, e $m^{(g)}$ é a média da geração g . Note que a covariância deve necessariamente ser atualizada antes da média, pois deve-se utilizar a mesma média que gerou as amostras daquela geração. Caso a média seja atualizada antes da covariância, teremos sempre resultados enviesados, prejudicando o aspecto de *exploration* da estratégia, e realizando excessivo *exploitation*.

Em seguida, é atualizado o valor da média, segundo a equação:

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

Após atualizar a média, é realizada o sorteio das amostras da nova geração, por meio de uma distribuição gaussiana multivariada com a média e covariância recém calculadas.

2. Teste da implementação

Após implementada a estratégia evolutiva simples, a mesma foi testada juntamente com o CMA-ES para as funções Esfera Transladada, Ackley, Schaffer 2D e Rastrigin, cujas expressões constam no roteiro. Nas figuras 1 a 8, podemos ver o resultado final, isto é, o ponto para o qual cada algoritmo convergiu, de uma execução de cada algoritmo para a minimização de cada uma das funções. A SES foi utilizada com $\lambda = 24$ e $\mu = 12$, enquanto o CMA-ES foi utilizado conforme o padrão para problemas 2D, com $\lambda = 5$ e $\mu = 3$.

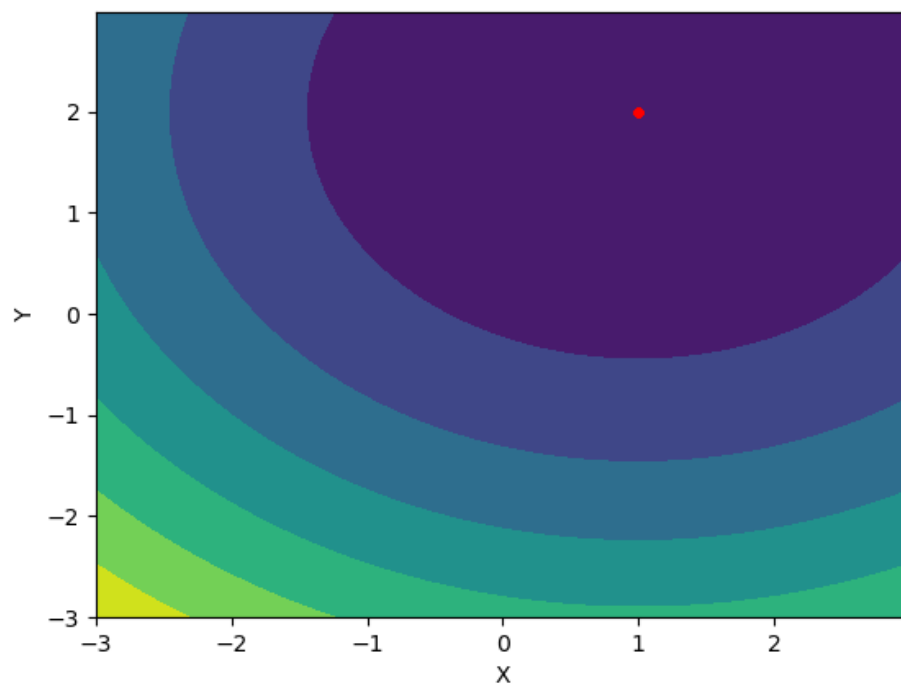


Figura 1: Gráfico da solução final encontrada pela SES para a função Esfera Transladada.

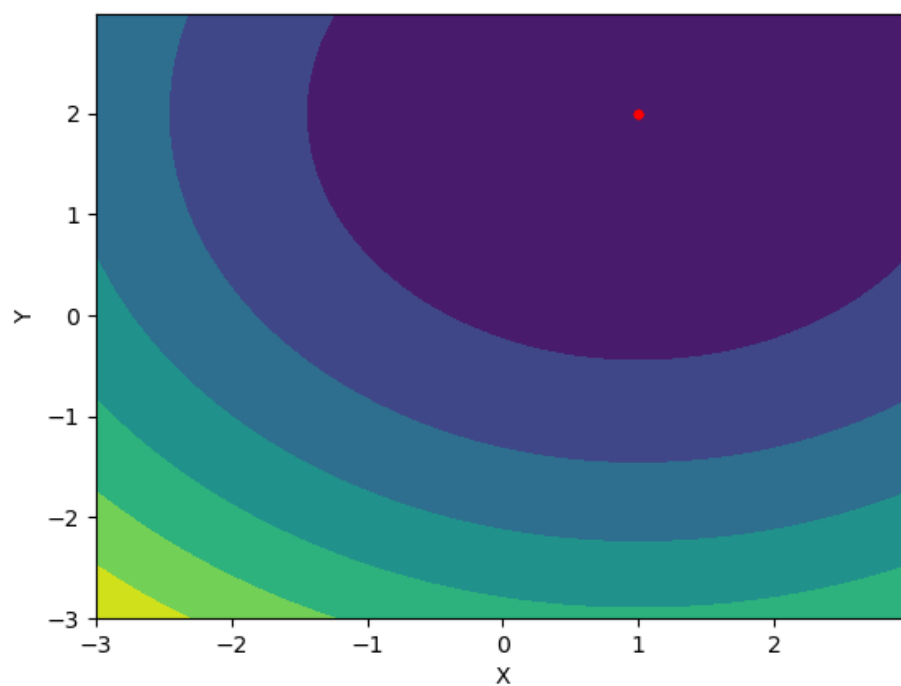


Figura 2: Gráfico da solução final encontrada pela CMA-ES para a função Esfera Transladada.

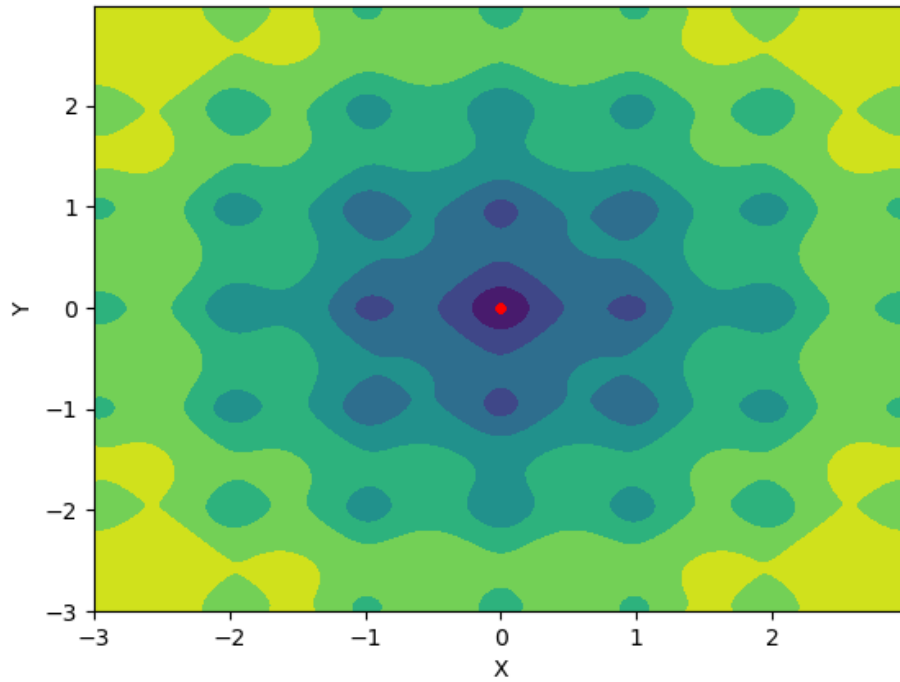


Figura 3: Gráfico da solução final encontrada pela SES para a função de Ackley.

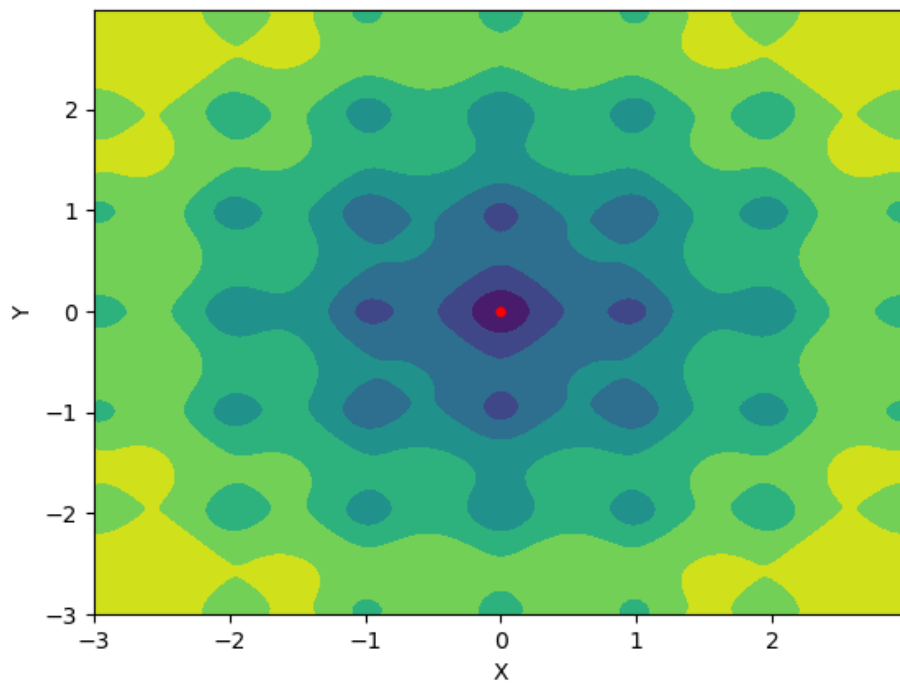


Figura 4: Gráfico da solução final encontrada pela CMA-ES para a função de Ackley.

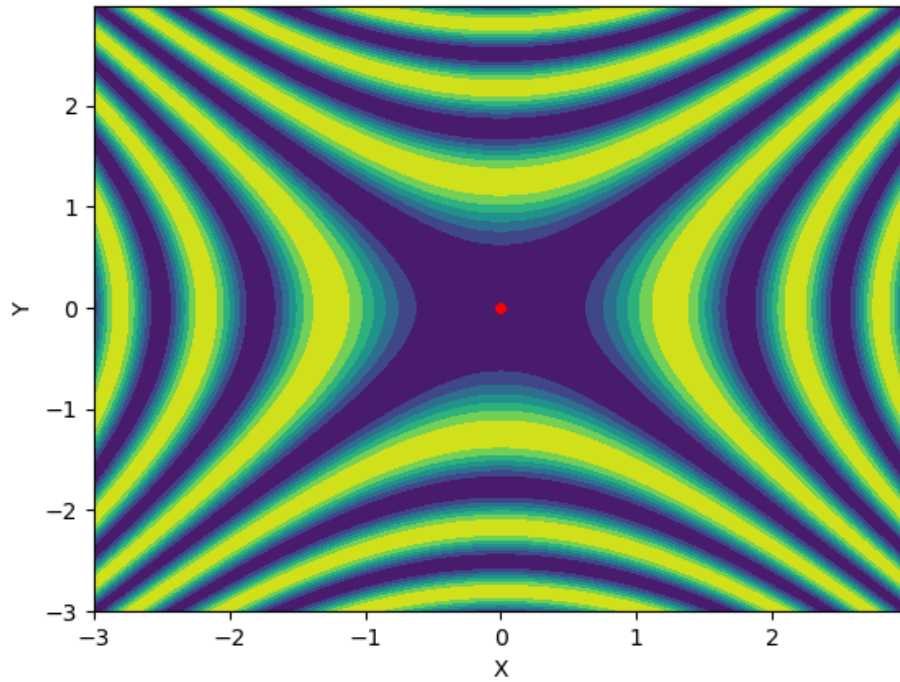


Figura 5: Gráfico da solução final encontrada pela SES para a função de Schaffer 2D.

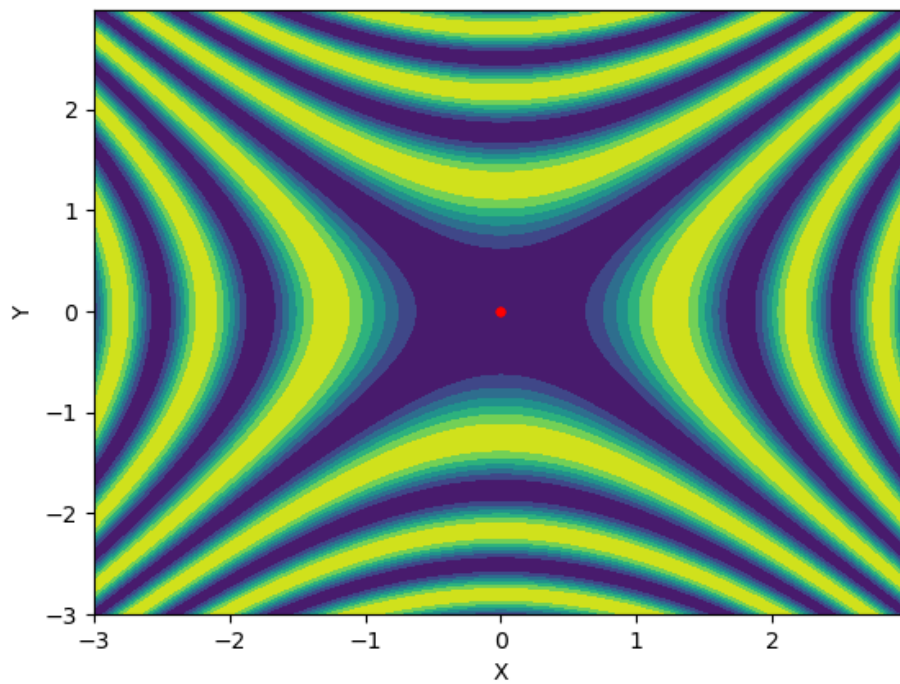


Figura 6: Gráfico da solução final encontrada pela CMA-ES para a função de Schaffer 2D.

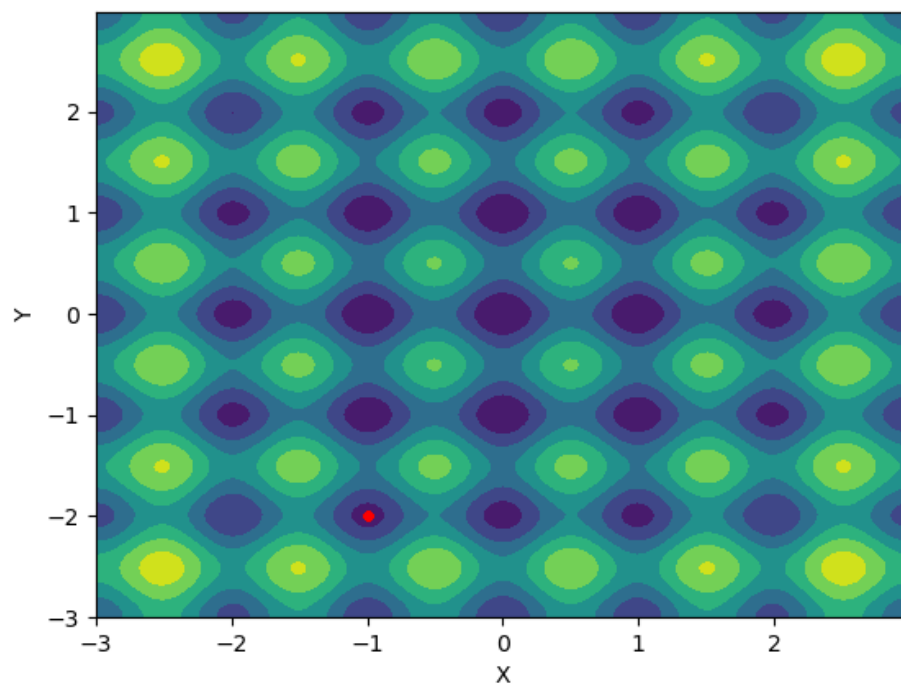


Figura 7: Gráfico da solução final encontrada pela SES para a função de Rastrigin.

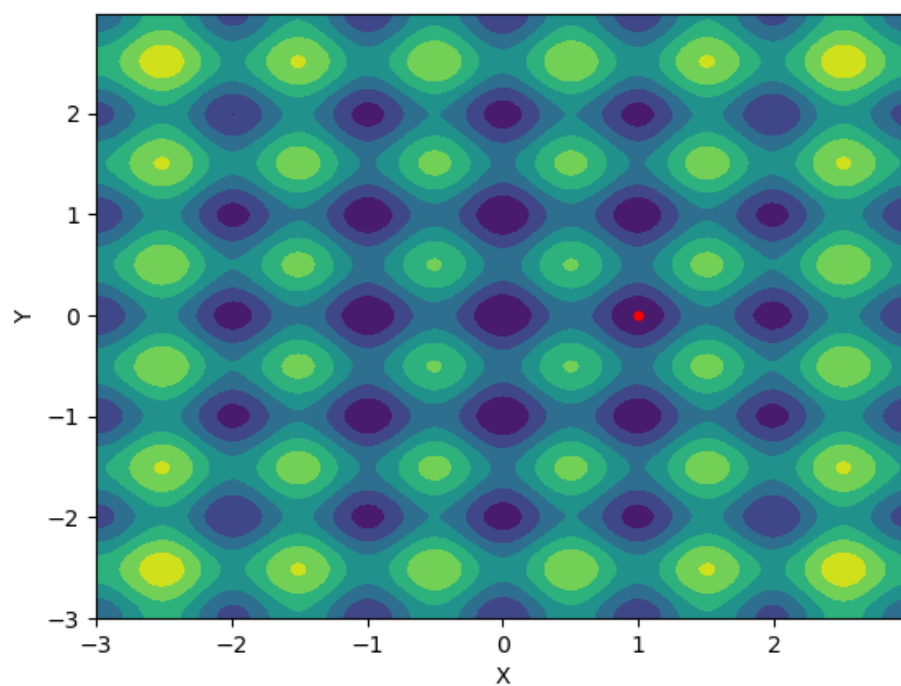


Figura 8: Gráfico da solução final encontrada pela CMA-ES para a função de Rastrigin.

Além de produzir essas figuras, foram rodadas várias animações mostrando a evolução da solução de cada algoritmo para cada função.

Para a esfera transladada, percebeu-se que ambas os algoritmos convergiam praticamente para o mesmo ponto, o mínimo global. No entanto, pode-se observar que a dispersão das amostras no caso do SES diminuía muito rapidamente próximo do centro da esfera, levando eventualmente a uma convergência estagnada, isto é, que estagnou num ponto subótimo pois o passo diminuiu rápido demais. Em contrapartida, o CMA-ES exibiu uma convergência mais dispersa próxima do centro, seguida de uma abrupta redução da dispersão, centrada no mínimo global.

Para a função de Ackley, ocorria em alguns casos do SES convergir para um mínimo local não global, demonstrando um certo vício no aspecto de *exploitation*, em detrimento do *exploration*. O CMA-ES não apresentou esse comportamento, convergindo sempre para o mínimo global.

Para a função de Schaffer 2D, observou-se uma curiosa inversão de desempenho. O CMA-ES por algumas vezes acabava por convergir para mínimos locais subótimos, pontos mínimos dos “vales” da superfície de nível, com coordenadas 0 em um dos eixos, e um pouco deslocado no outro. Houve inclusive um caso em que o CMA-ES convergiu para fora da região do plot. O SES por sua vez, apesar de também ter tido suas falhas ao convergir para pontos de mínimo localizados na diagonal, em geral convergiu mais vezes para o mínimo global. Vale observar o comportamento do CMA-ES ao encontrar mínimos subótimos, em que ele se concentrava numa região azulada, diminuía sua dispersão, e em seguida basicamente buscava o ótimo naquela região.

Por fim, para a função de Rastrigin, ambos os métodos tiveram dificuldade em encontrar o mínimo global. O SES, talvez um pouco viciado em *exploitation* rapidamente convergia para um mínimo local próximo de seu chute inicial. O CMA-ES apesar de exibir uma busca mais persistente, eventualmente acabava convergindo para um ponto subótimo também.

3. Testes benchmark

Para realizar uma análise comparativa mais rigorosa entre os dois algoritmos para cada função, foram realizados testes *benchmark* por meio de várias simulações de Monte Carlo, e coletou-se as estatísticas do desempenho de cada algoritmo. Foram comparados os algoritmos SES com $(\lambda, \mu) = (6, 3)$, $(\lambda, \mu) = (12, 6)$ e $(\lambda, \mu) = (24, 12)$, e o CMA-ES padrão, com $(\lambda, \mu) = (6, 3)$.

Após cada bateria de simulações, foram produzidos gráficos da evolução do valor da função de *fitness*, considerando o *fitness* médio das amostras a cada geração, e também o melhor *fitness* de cada geração. As figuras 9 a 16 mostram os gráficos com as estatísticas para cada uma das 4 funções de *fitness*.

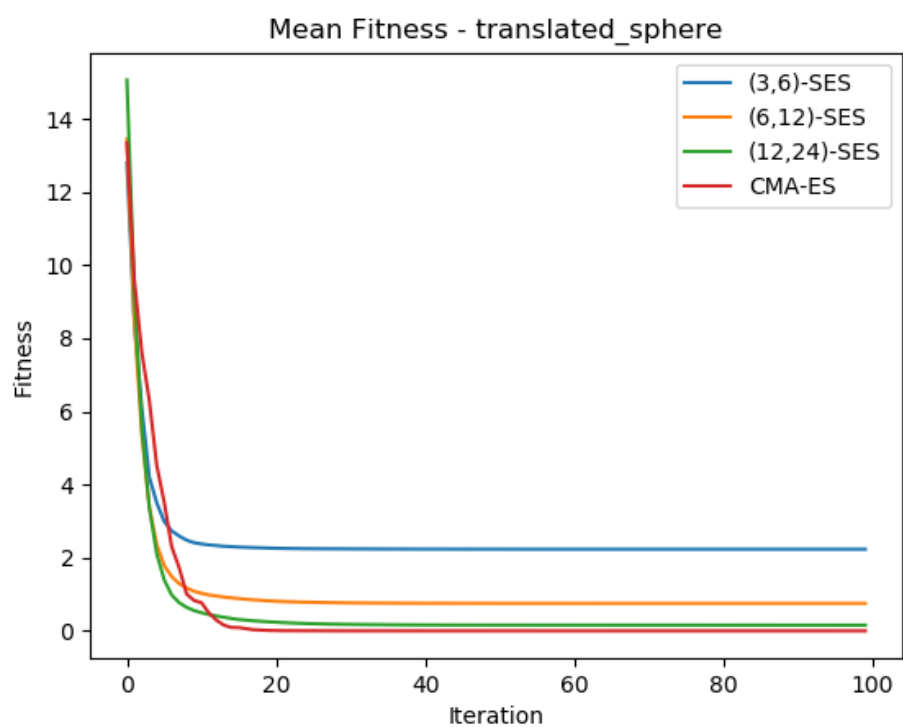


Figura 9: Valor de fitness médio por geração para a função Esfera Transladada.

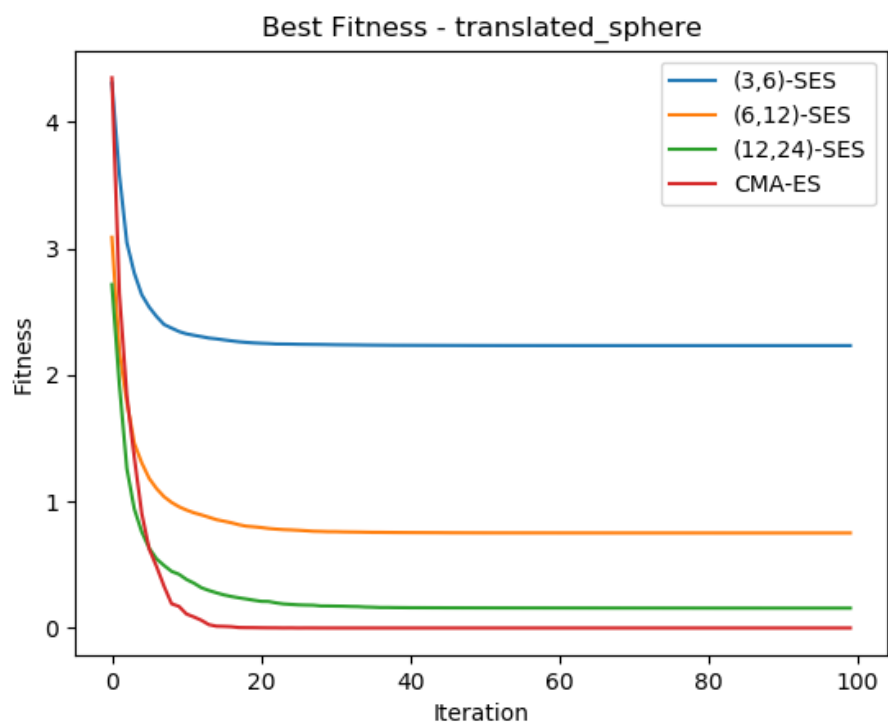


Figura 10: Valor do melhor fitness por geração para a função Esfera Transladada.

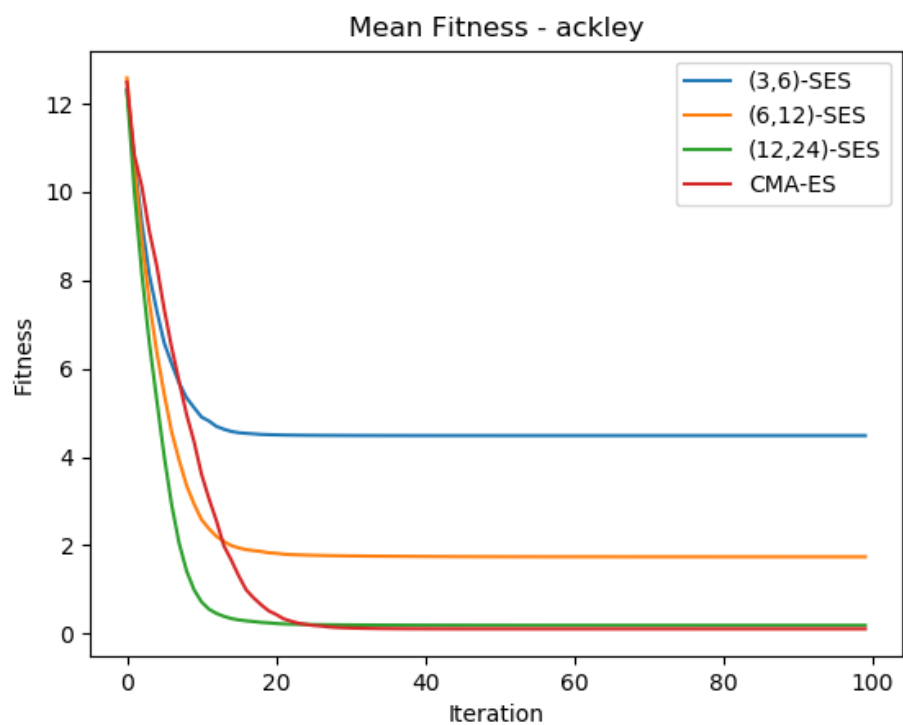


Figura 11: Valor de fitness médio por geração para a função de Ackley.

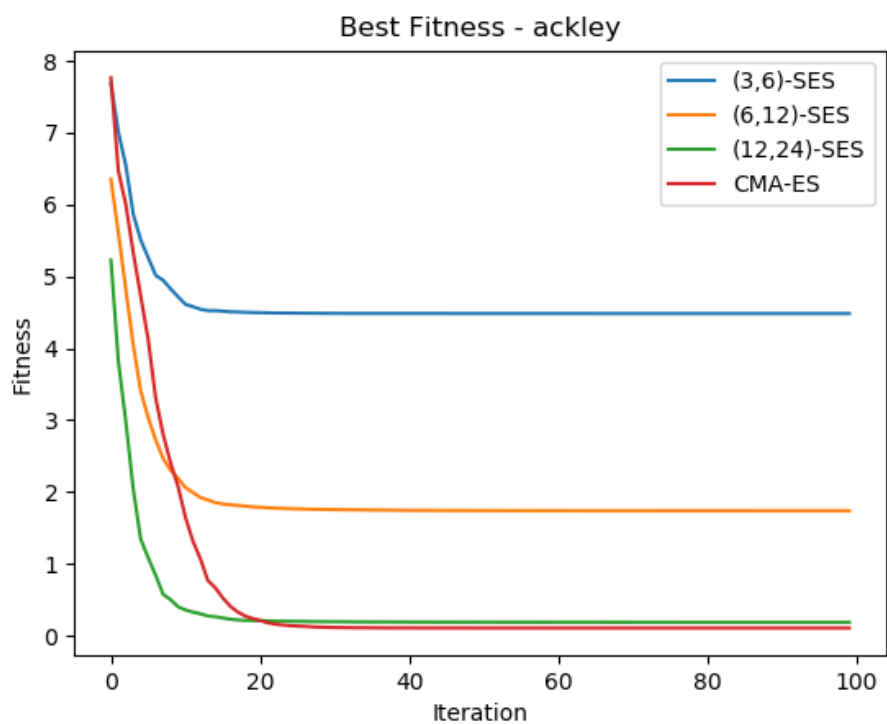


Figura 12: Valor do melhor fitness por geração para a função de Ackley.

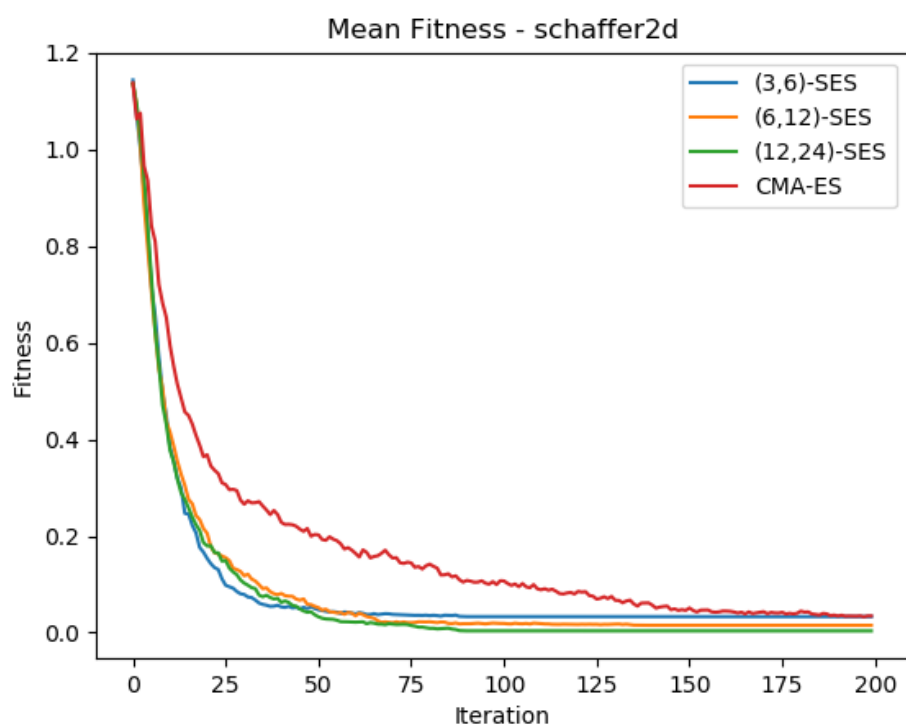


Figura 13: Valor de fitness médio por geração para a função de Schaffer 2D.

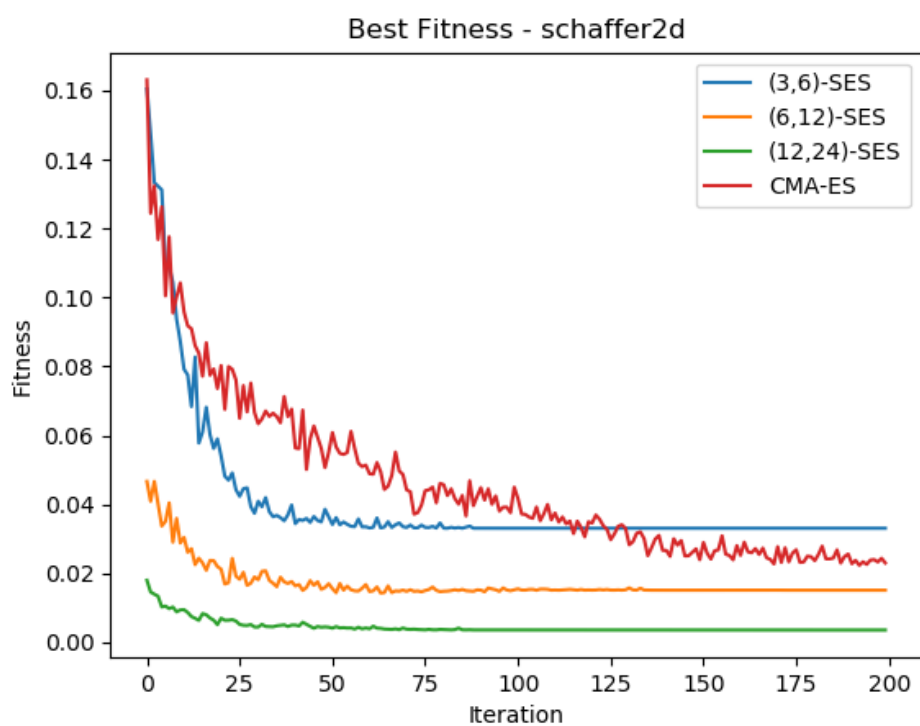


Figura 14: Valor do melhor fitness por geração para a função de Schaffer 2D.

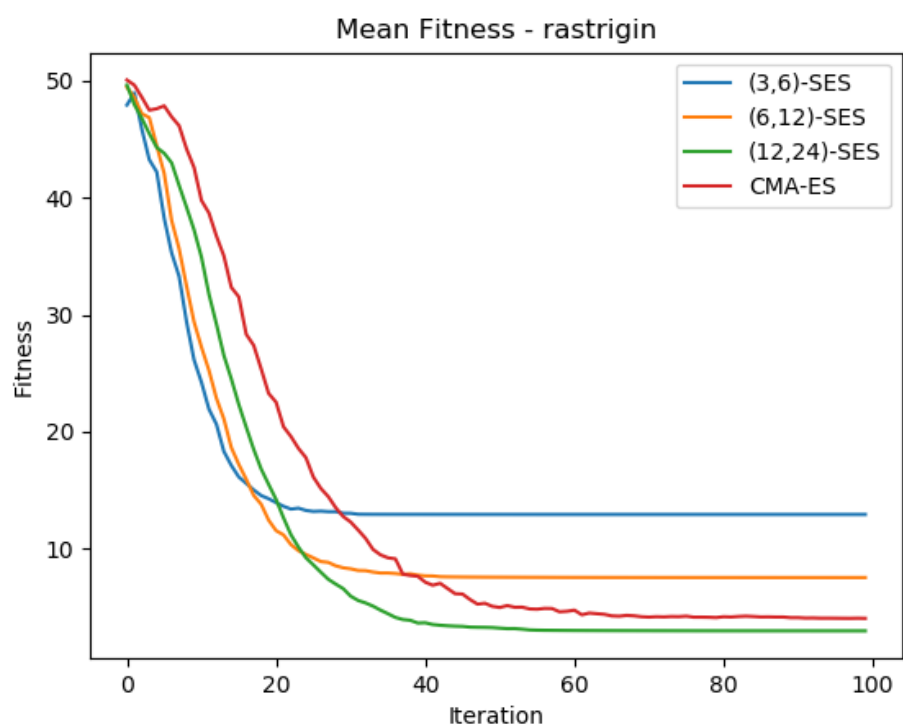


Figura 15: Valor de fitness médio por geração para a função de Rastrigin.

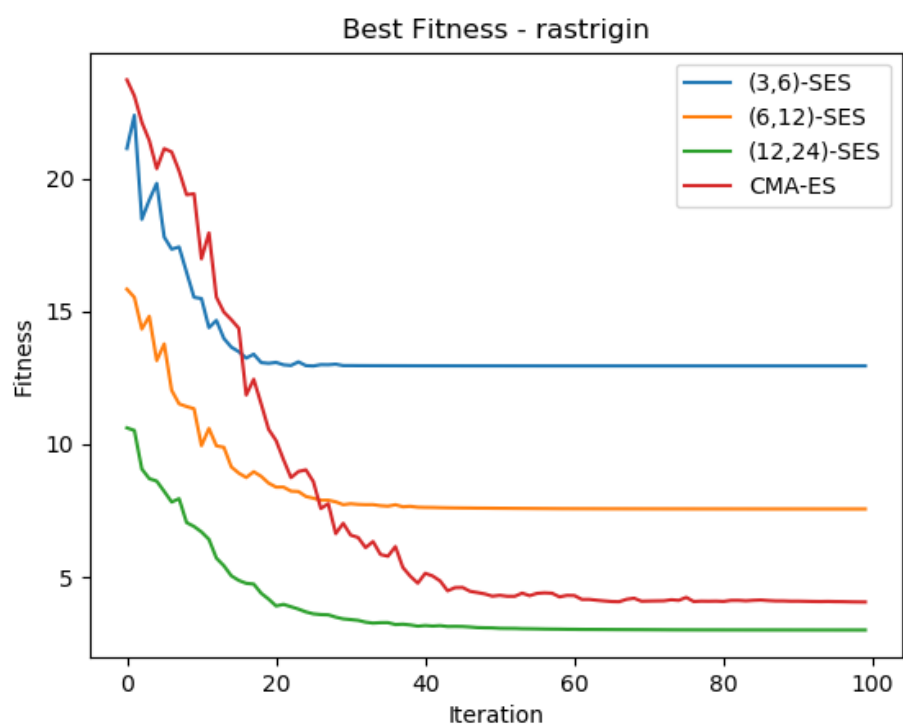


Figura 16: Valor do melhor fitness por geração para a função de Rastrigin.

Analizando os gráficos das figuras 9 e 10, vemos podemos comparar o comportamento do SES e do CMA-ES para a função esfera transladada. Vemos que conforme o tamanho da população usada pelo SES aumenta, mais baixo é o valor final obtido. Ou seja, a qualidade da solução final melhora com o aumento do tamanho da população. Isso demonstra a calibração do SES sob a ótica do *trade-off exploration x exploitation*. Quanto maior a população usada, mais se sobressai o aspecto de *exploration*, permitindo encontrar soluções melhores.

Em contrapartida, o CMA-ES exibe uma curva menos íngreme, e com um platô ainda mais baixo, ou seja, converge mais devagar no começo, e encontra uma solução ainda melhor (no caso, a solução ótima). Esse comportamento é resultado da sofisticação do CMA-ES, que implementa uma taxa de aprendizado variável.

Para a função de Ackley, o SES apresenta uma convergência mais rápida e melhor conforme aumenta o tamanho da população utilizado, novamente demonstrando uma dependência desse hiperparâmetro para ajustar o seu passo, ou sua taxa de aprendizado. O CMA-ES por sua vez, apresenta uma curva de convergência distinta, tendo uma velocidade de convergência mais lenta que o SES-(24, 12), mas eliminando o fenômeno de estagnação.

Para a função de Schaffer 2D, foram executadas 2 vezes mais iterações (ou seja, 2 vezes mais gerações de cada algoritmo) com 2.5 vezes mais simulações Monte Carlo. O intuito era reduzir o ruído das simulações para essa função, que apresenta um aspecto um pouco mais complicado que as demais.

Das figuras 13 e 14 podemos constatar esse aspecto ruidoso da função. No entanto, vemos que o SES foi em geral menos sujeito a todo esse ruído, e apresentou uma velocidade maior de convergência, e ainda platôs mais baixos (conforme aumenta-se o tamanho da população) que o CMA-ES. É possível ainda notar que o CMA-ES não chegou exatamente a exibir platôs muito visíveis nos gráficos, diferente de todas as versões do SES.

Uma possível explicação do porquê o SES foi melhor que o CMA-ES nesse caso reside na análise dos platôs. O CMA-ES possui uma taxa de aprendizagem adaptativa, enquanto essa funcionalidade no SES é regulada pelo tamanho da população. Isso faz com que o CMA-ES precise fazer muito mais *exploration* da função antes de convergir. Com uma função tão ruidosa, o CMA-ES provavelmente precisaria de muito mais iterações até chegar ao nível de *exploitation* que o SES exibe em poucas iterações. Ou seja, o CMA-ES demora muito mais para exibir um platô, devido a sua alta tendência de *exploration*. Como o SES possui esse viés de *exploitation* intrínseco, ele é menos sujeito a buscar mínimos novos da função e tende a concentrar sua convergência nos mínimos mais óbvios (os mínimos com maior probabilidade de serem encontrados dado um chute inicial aleatório).

Por fim, olhando as figuras 15 e 16, que exibem as estatísticas para a função de Rastrigin, temos que novamente o SES apresenta uma dependência da taxa de aprendizagem com o tamanho da população, convergindo mais rapidamente e para pontos piores quanto menor for o tamanho da população.

Vemos que o SES-(24, 12) foi capaz de superar o CMA-ES nesse caso. Naturalmente, é de se esperar que um algoritmo com uma população maior tenha mais chances de detectar mínimos subótimos e evita-los. No caso dessa função, marcada pela presença de vários mínimos locais próximos, isolados por regiões de máximos locais, a capacidade do CMA-ES de regular sua taxa de aprendizagem é superada pelo tamanho da população usada pelo SES, e seu *modus operandi* de convergir mais devagar (e para pontos melhores) quanto maior for a população.