2º Relatório referente a disciplina de Métodos Computacionais Inspirados na Natureza (MCIN)

Autor: Larangeira, Vicente d'Orema Albuquerque

Propostas de melhorias ao algoritmo de Evolução Diferencial: um estudo comparativo

Resumo

Esse estudo tem como objetivo propor uma série de modificações ao algoritmo de otimização conhecido como Evolução Diferencial de modo a melhorar sua performance em um conjunto de sete problemas do CEC2014. Os resultados obtidos pelo algoritmo modificado são comparados com os obtidos por dois modelos clássicos de Evolução Diferencial e também pelo algoritmo conhecido como *Particle Swarm Optimization* (PSO).

1 Introdução

Ao longo das últimas décadas observou-se um crescimento significativo no ramo da computação evolutiva. Esse crescimento pode ser comprovado pela diversidade de métodos de otimização presentes na literatura. Como resumido em [1], o objetivo desses métodos é encontrar o máximo ou o mínimo que uma função ou um conjunto de funções podem assumir considerando um conjunto de variáveis $\boldsymbol{X} = (x_1, x_2, \cdots, x_D)$ num espaço de busca com D dimensões. Sendo assim, a qualidade de um algoritmo de otimização é demonstrada pela sua eficácia e eficiência em resolver a maior diversidade possível de problemas.

Grande parte dos estudos recentes envolvendo computação evolutiva consiste em propor modificações aos algoritmos de otimização clássicos de modo a aperfeiçoá-los. Seguindo essa linha, esse trabalho propõe uma série de modificações ao algoritmo conhecido como Evolução Diferencial considerando algumas das principais limitações desse algoritmo. Para avaliar o efeito das modificações na performance do algoritmo, em relação à dois modelos clássicos de Evolução Diferencial, foram selecionadas sete funções do desafio do *IEEE Congress on Evolutionary Computation (CEC) 2014*. As funções também são submetidas ao algoritmo conhecido como *Particle Swarm Optimization* (PSO).

Esse relatório está organizado da seguinte forma: a seção 2 apresenta os algoritmos de otimização avaliados, incluindo o algoritmo modificado, a seção 3 apresenta a metodologia de análise e as funções objetivo usadas, a seção 4 apresenta e discute os resultados obtidos e a seção 5 apresenta as conclusões.

2 Algoritmos avaliados

Nessa seção serão apresentados os modelos de *Particle Swarm Optimization* e Evolução Diferencial avaliados e os parâmetros adotados em cada um. Esses parâmetros foram escolhidos buscando gerar os melhores resultados tendo como base trabalhos anteriores e testes preliminares. Por último, são apresentadas e justificadas as modificações propostas a Evolução Diferencial.

2.1 Particle Swarm Optimization (PSO)

O modelo adotado para o *Particle Swarm Optimization* foi o mesmo usado em [1]. Os parâmetros necessários para esse algoritmo são o tamanho da população, a velocidade máxima permitida e os componentes que comandam os três componentes de velocidade de cada indivíduo.

Apesar de não ser o foco desse estudo, o algoritmo usado apresenta algumas modificações em relação ao originalmente proposto por Kennedy e Eberhart [2]. Foi adotado o coeficiente de constrição de Clerc [3] que multiplica todos os componentes de velocidade por 0,73 e limitou-se a velocidade máxima a 20% do espaço de busca como feito em [4]. Por último, foi adotado um decaimento linear para a componente que controla a velocidade de inércia como feito em [5].

Os valores adotados para cada um dos parâmetros mencionados estão apresentados na Tabela 1. O código utilizado é de implementação própria.

Tabela 1: Parâmetros adotados para o Partcile Swarm Optimization (PSO)

Descrição	PSO
Tamanho da População (NP)	100
Inércia ($w = f(\text{num}_{\text{iter}})$)	(0,9 – 0,4)
C1 (termo cognitivo)	2,0
C2 (termo social)	2,0
Velocidade máxima	0.2*($lim_{mín} - lim_{máx}$)
Fator de Clerc (χ = 0,73)	Sim

É importante ressaltar que o algoritmo de *Particle Swarm Optimization* (PSO) também apresenta um grande potencial em resolver problemas de otimização. Essa capacidade pode ser comprovada pela variedade de estudos relacionados ao algoritmo [6]. Além disso, este foi um dos algoritmos pioneiros em utilizar o conceito de inteligência de enxame presente em algoritmos desenvolvidos recentemente como o *Social Spider Algoritm* [7] e o *Artificial Bee Colony* [8].

2.2 Evolução Diferencial clássico (ED)

Os parâmetros necessários para esse algoritmo são o tamanho da população, o tamanho do passo e a taxa de *crossover*. O número reduzido de parâmetros, sua fácil implementação e sua eficiência justificam o elevado interesse que esse algoritmo desperta.

Essas características também estimulam diversos estudos voltados não só a encontrar os melhores parâmetros para o algoritmo, mas também propostas de modificações como as que são apresentadas nesse trabalho.

Os modelos adotados para representar o "estado da arte" da Evolução Diferencial são: o modelo clássico ED/rand/1 [9] e a variação conhecida como ED/best/2 [10]. Esses modelos foram selecionados pois apresentam características diferentes em termos de velocidade de convergência e capacidades de exploração e explotação. Enquanto o primeiro é mais lento o segundo tem mais chance de ficar preso em mínimos locais (convergência prematura).

Os valores dos parâmetros adotados nesse estudo estão apresentados na Tabela 2 e foram selecionados por apresentar os melhores resultados durante análises preliminares. Os códigos utilizados são de implementação própria e foram desenvolvido baseados em [10].

Tabela 2: Parâmetros adotados para Evolução Diferencial clássica (ED)

Descrição	ED/rand/1	ED/best/2
Tamanho da População NP	100	100
Taxa de Crossover (%)	90%	90%
Tamanho do passo (F)	0,50	0,55

2.3 Evolução Diferencial modificada

Como qualquer algoritmo de otimização, a performance da Evolução Diferencial clássica varia de acordo com o problema, o que faz com que ela obtenha melhores resultados em algumas funções e piores em outras. O objetivo das modificações sugeridas nesse trabalho é melhorar o desempenho desse algoritmo no maior número de funções de otimização possível. Essa seção apresenta o conjunto de modificações propostas justificando o porquê de suas aplicações.

2.3.1 População Inicial

A população inicial em todas as análises é a mesma utilizada pela Evolução Diferencial clássica apresentada na Tabela 2 (NP = 100). Uma pequena modificação foi o posicionamento de dois indivíduos nos extremos no espaço de busca e um no centro. Essa modificação agilizaria a busca caso o ótimo esteja próximo a uma dessas regiões. Os demais 97 indivíduos são gerados aleatoriamente distribuídos pelo espaço de busca.

2.3.2 Mutação e *Crossover*

A principal modificação implementada nesse trabalho foi inspirada no algoritmo SaDE proposto por Qin e Suganthan [10]. Essa modificação consiste em aplicar à um quarto da população (25 indivíduos) a estratégia ED/rand/1 seguido do mesmo um quarto usando ED/best/2. Para a metade restante da população é aplicado a estratégia de mutação que tiver o maior número de sucessos no processo de seleção. No início desse processo a população é "embaralhada", de modo que os indivíduos da população tenham a mesma chance de ser comparados com indivíduos gerados pelas duas estratégias.

Em seguida, duas modificações adicionais foram incluídas a estratégia do ED/rand/1. A primeira é que os vetores sorteados são organizados em função dos seus valores de aptidão. Assim, o melhor vetor (aquele com maior aptidão) é o vetor com maior importância na criação do vetor tentativa. Considerando os vetores apresentados na Figura 1 em um problema de minimização, isso significa que $f(\mathbf{v}_3) < f(\mathbf{v}_2) < f(\mathbf{v}_1)$.

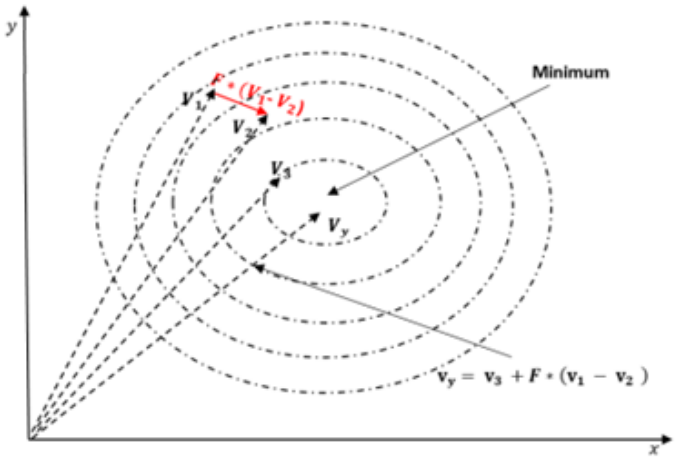

Figura 1: Representação gráfica da estratégia ED/rand/1

A aplicação dessa modificação por si só já gerou uma melhora significativa nos testes preliminares.

A segunda modificação foi o ajuste do valor do passo (F) em função da diferença dos valores de aptidão entre os dois vetores para qual o passo é aplicado. Esse ajuste é realizado pela seguinte relação:

$$F = 1{,}0 - \frac{\min\left(f(v_2),\ f(v_1)\right)}{f(v_2) + f(v_1)} \tag{1}$$

Esse ajuste faz com que o valor do passo diminua ao longo das iterações favorecendo uma busca global no início e uma busca local no final [11]. Além disso o valor de F é mantido entre [0,5; 1) estando dentro do intervalo normalmente adotado. Para a estratégia de mutação ED/best/2 foi adotado o mesmo valor de F apresentado na Tabela 2.

A taxa de *crossrate* foi mantido em 0,9 tanto para o ED/rand/1 quanto para ED/best/2. Em todo caso, é possível encontrar na literatura uma grande diversidade de trabalhos que propõem ajustes dinâmicos ou estratégias auto adaptativas para definir esse parâmetro. Eltaieib e Mahmodd [12] apresentam uma valiosa pesquisa sobre alguns desses trabalhos.

2.3.3 Convergência prematura

A convergência prematura ocorre quando o algoritmo fica preso em um mínimo local. A modificação implementada para tentar contornar esse problema foi reiniciar a população caso o algoritmo não encontre um resultado melhor em 50 gerações seguidas.

Entretanto o processo de inicialização adotado nesse trabalho faz uma consideração extra: a população só é reiniciada caso a relação apresentada na Eq. (2) for menor do que um erro pré-estabelecido. Nessa equação, f_{max} e f_{min} correspondem ao melhor e pior valores de aptidão e m_t corresponde ao valor médio. Para as análises realizadas nesse trabalho foi adotado o valor do critério de parada (10^{-8}).

$$Dif = \sqrt{\frac{(f_{max} - m_t)^2 + (f_{min} - m_t)^2}{2}} \quad (2)$$

Somente após atingir as duas condições mencionadas acima a população é reiniciada. No algoritmo implementado essa reinicialização pode ser realizada das duas seguintes maneiras:

- A primeira é quando os indivíduos estão muito próximos uns dos outros (maior distância < 1,0). Nesse caso a população inteira está presa num mínimo local. Para essas situações, guarda-se somente o melhor indivíduo enquanto o resto da nova população é gerada aleatoriamente num espaço de busca reduzido. Esse espaço de busca é definido pela média entre os limites iniciais (extremos do espaço de busca) e os extremos ocupados pelos indivíduos. Essa modificação é condizente com problemas conhecidos de otimização como Ackley e Griewank.
- A segunda é quando a maior distância entre os indivíduos é superior a 1,0. Nesse caso foi assumido que exista mais de um mínimo local e que a população está presa em regiões diferentes. Para essas situações, metade da população é mantida e a outra metade é substituída por novos indivíduos gerados aleatoriamente pelo espaço de busca. Essa modificação é condizente com problemas conhecidos de otimização como Rastrigin e Shaffer (F6).

3 Metodologia de análise e funções de avaliação

Como mencionado anteriormente, os aspectos mais importantes para determinar a qualidade de um algoritmo de otimização são a sua eficácia e eficiência em encontrar a melhor solução em um problema ou conjunto de problemas.

Nesse contexto, esse trabalho selecionou sete problemas de minimização para comparar o desempenho dos algoritmos avaliados. Cada um desses problemas corresponde a uma das funções propostas no desafio do CEC2014. A relação completa de funções pode ser encontrada em https://github.com/P-N-Suganthan/CEC2014. As Tabela 3 e 4 apresentam, respectivamente, as funções objetivo selecionadas e os critérios adotados nas análises.

Tabela 3: Lista de Funções objetivo avaliadas

ID	*Nome da função*	*Mínimo global*
F1	*Rotated High Conditioned Elliptic Function*	*100,0*
F2	*Rotated Bent Cigar Function*	*200,0*
F4	*Shifted and Rotated Rosenbrock's Function*	*400,0*
F6	*Shifted and Rotated Weierstrass Function*	*600,0*
F7	*Shifted and Rotated Griewank's Function*	*700,0*
F9	*Shifted and Rotated Rastrigin's Function*	*900,0*
F14	*Shifted and Rotated HGBat Function*	*1400,0*

Tabela 4: Parâmetros das Análises

Dimensões (número de variáveis)	10 e 30
Execuções (Runs)	25
Número máximo de avaliações (MaxFES)	MaxFES = 10000*D = 100000 e 300000
Região de busca (Intervalos)	$[-100,100]^D$
Geração da população	Uniforme na região de busca
Ótimo global	Igual a origem adotada
Critério de parada	MaxFES (sucesso ou não) ou Erro < 10^{-8}
Taxa de sucesso	Número de Sucesso/25

Nesse estudo, os algoritmos são rodados 25 vezes para cada uma das funções objetivo tendo, em cada rodada, um limite de 100000 (10D) e 300000 (30D) avaliações da função objetivo (MaxFES) para encontrar o ótimo global (dentro do erro de 10^{-8}). À cada rodada, uma série de resultados parciais (0*MaxFES, 0,001*MaxFES, 0,01*MaxFES, 0,1*MaxFES, 0.2*MaxFES, ..., 1*FES) é salva para avaliar o comportamento de convergência de cada algoritmo.

Ao final de cada rodada são retornados os valores dos melhores resultados encontrados, os resultados parciais e se o algoritmo teve sucesso em encontrar o ótimo global ou não. Ao final das 25 rodadas são determinados, o melhor, o pior, a média, a mediana e o desvio padrão desses 25 melhores resultados e a taxa de sucesso.

Por último, obtêm-se a média de cada um dos resultados parciais considerando os valores parciais obtidos nas 25 rodadas. São a partir dessas médias que os gráficos apresentados na seção de resultados são construídos.

4 Resultados

Nessa seção são apresentados os resultados obtidos nesse estudo. Esses resultados foram separados para cada uma das sete funções objetivo avaliadas permitindo a comparação das performances dos algoritmos caso a caso.

A Tabela 5 e a Tabela 6 apresentam, respectivamente, os principais resultados obtidos por cada algoritmo em 10 e 30 dimensões. Os valores de melhor, pior, mediana, média, desvio padrão e taxa de sucesso apresentados nessas tabelas são obtidos considerando as 25 rodadas que cada algoritmo executou conforme descrito na seção 3.

Os valores em negrito indicam o algoritmo que teve o melhor desempenho sendo o primeiro critério avaliado a taxa de sucesso. Caso dois ou mais algoritmos obtenham a mesma taxa de sucesso a velocidade de convergência pode ser usada como critério de desempate. Uma maneira simples de observar essa velocidade de convergência é analisando os gráficos para a F1 e F2 em 10 dimensões (Figura 2 e Figura 3).

Nos casos em que nenhum dos algoritmos obteve sucesso, o critério de desempate foi o melhor valor encontrado. Esse valor corresponde ao ponto com maior aptidão indicado nas tabelas como o menor valor de erro (Melhor).

Tabela 5: Resultados obtidos pelos algoritmos para as funções objetivos avaliadas (10D)

Função	Método	Melhor	Pior	Mediana	Média	Dev.	Taxa de Sucesso
F1	PSO	2.147E-02	1.486E+03	1.459E+02	2.725E+02	3.498E+02	0.00%
	ED/rand1	4.986E-09	9.908E-09	7.996E-09	8.028E-09	1.404E-09	**100.00%**
	ED/best2	5.535E-09	9.986E-09	8.275E-09	8.120E-09	1.326E-09	**100.00%**
	ED/mod	4.792E-09	9.999E-09	8.647E-09	8.426E-09	1.286E-09	**100.00%**
F2	PSO	3.706E-01	3.055E+02	5.832E+01	7.987E+01	8.230E+01	0.00%
	ED/rand1	3.348E-09	9.875E-09	7.336E-09	7.254E-09	1.984E-09	**100.00%**
	ED/best2	2.562E-09	9.849E-09	7.474E-09	7.358E-09	1.962E-09	**100.00%**
	ED/mod	4.212E-09	9.951E-09	8.437E-09	8.008E-09	1.659E-09	**100.00%**
F4	PSO	2.922E-06	3.478E+01	3.478E+01	2.139E+01	1.679E+01	0.00%
	ED/rand1	5.497E-09	3.478E+01	3.478E+01	2.522E+01	1.568E+01	**24.00%**
	ED/best2	9.071E-09	3.478E+01	3.478E+01	2.574E+01	1.485E+01	12.00%
	ED/mod	8.138E-09	3.478E+01	3.478E+01	2.713E+01	1.394E+01	8.00%
F6	PSO	9.656E-09	2.626E+00	1.735E-01	7.939E-01	8.541E-01	4.00%
	ED/rand1	5.836E-09	2.774E+00	1.210E-08	6.023E-01	7.584E-01	48.00%
	ED/best2	4.432E-09	3.899E+00	8.945E-01	1.269E+00	1.389E+00	36.00%
	ED/mod	3.807E-09	3.671E+00	8.308E-09	3.510E-01	8.490E-01	**80.00%**
F7	PSO	1.724E-02	1.451E-01	6.889E-02	7.354E-02	3.511E-02	0.00%
	ED/rand1	2.097E-02	4.499E-01	2.821E-01	2.561E-01	9.696E-02	0.00%
	ED/best2	2.704E-02	6.399E-01	3.372E-01	3.172E-01	1.795E-01	0.00%
	ED/mod	6.859E-09	3.201E-02	1.232E-02	1.448E-02	8.143E-03	**4.00%**
F9	PSO	1.990E+00	2.089E+01	1.293E+01	1.126E+01	4.778E+00	0.00%
	ED/rand1	1.506E+01	3.138E+01	2.468E+01	2.466E+01	4.474E+00	0.00%
	ED/best2	1.249E+01	3.917E+01	2.763E+01	2.734E+01	6.881E+00	0.00%
	ED/mod	**9.950E-01**	6.965E+00	2.985E+00	3.741E+00	1.500E+00	0.00%
F14	PSO	7.862E-02	3.291E-01	2.306E-01	2.073E-01	7.910E-02	0.00%
	ED/rand1	1.175E-01	2.687E-01	1.794E-01	1.780E-01	4.145E-02	0.00%
	ED/best2	6.864E-02	2.316E-01	1.252E-01	1.387E-01	4.442E-02	0.00%
	ED/mod	**6.168E-02**	4.061E-01	1.432E-01	1.594E-01	7.256E-02	0.00%

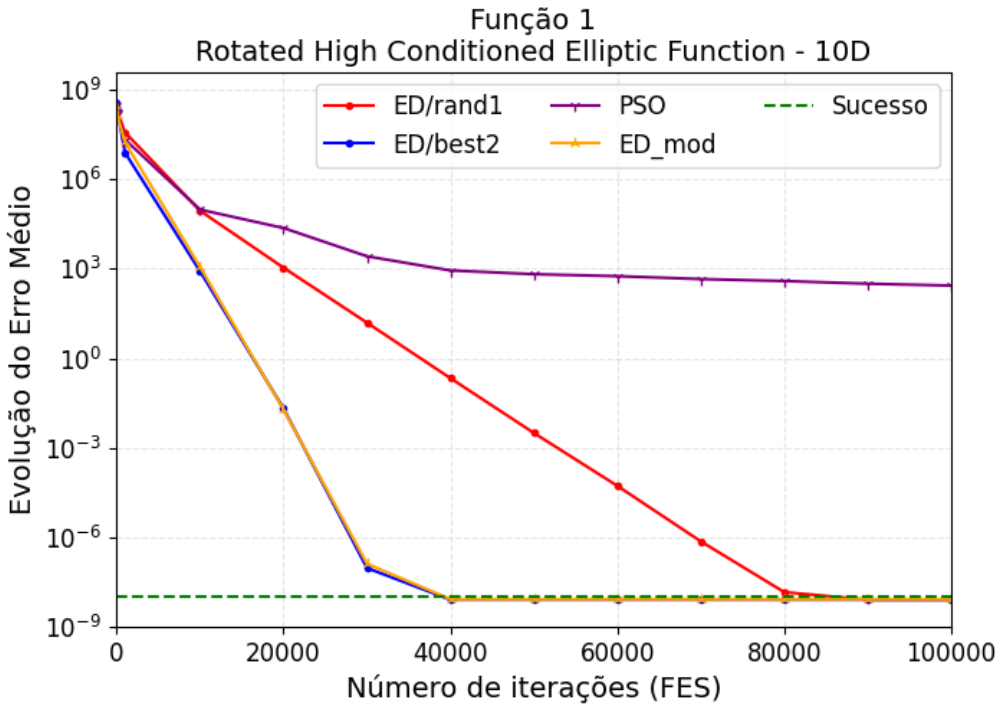

Figura 2: Convergência dos algoritmos para a função objetivo F1 (10D)

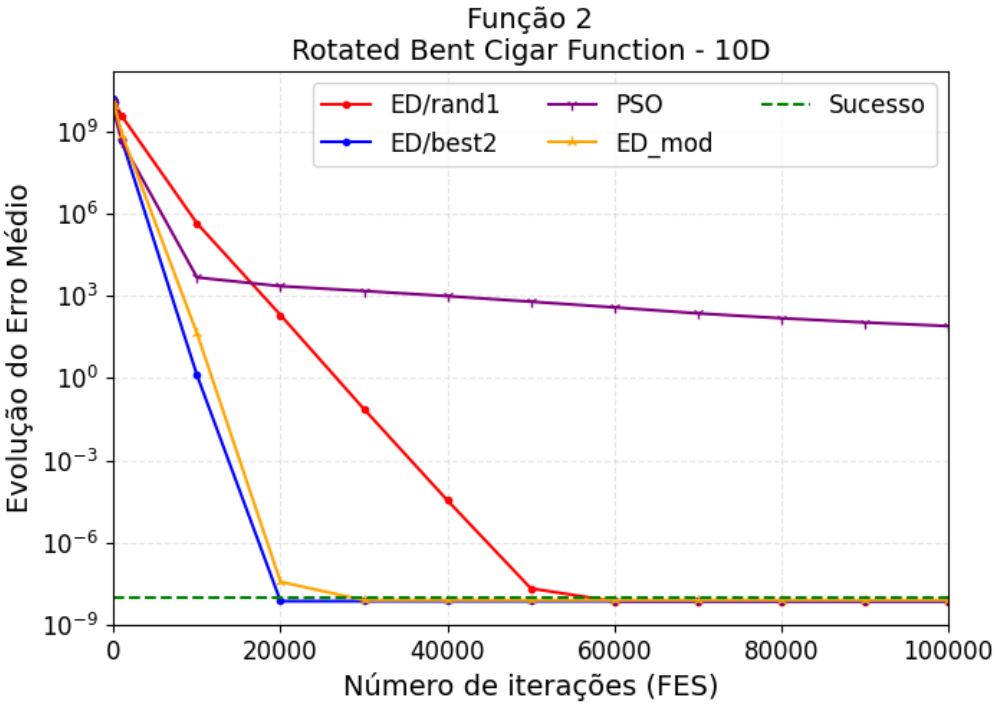

Figura 3: Convergência dos algoritmos para a função objetivo F2 (10D)

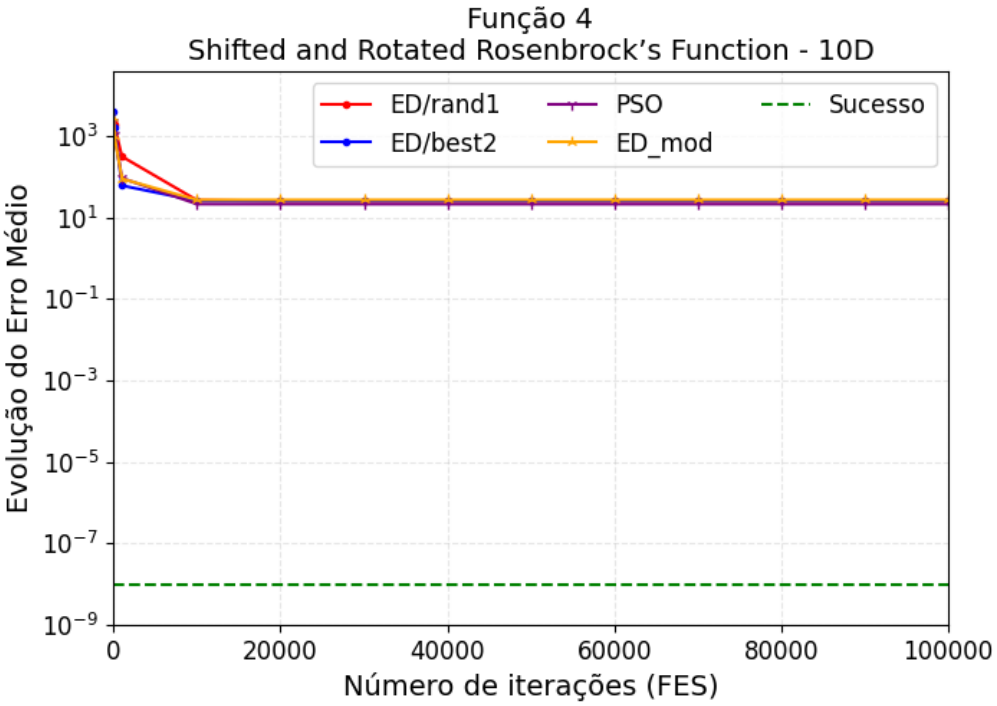

Figura 4: Convergência dos algoritmos para a função objetivo F4 (10D)

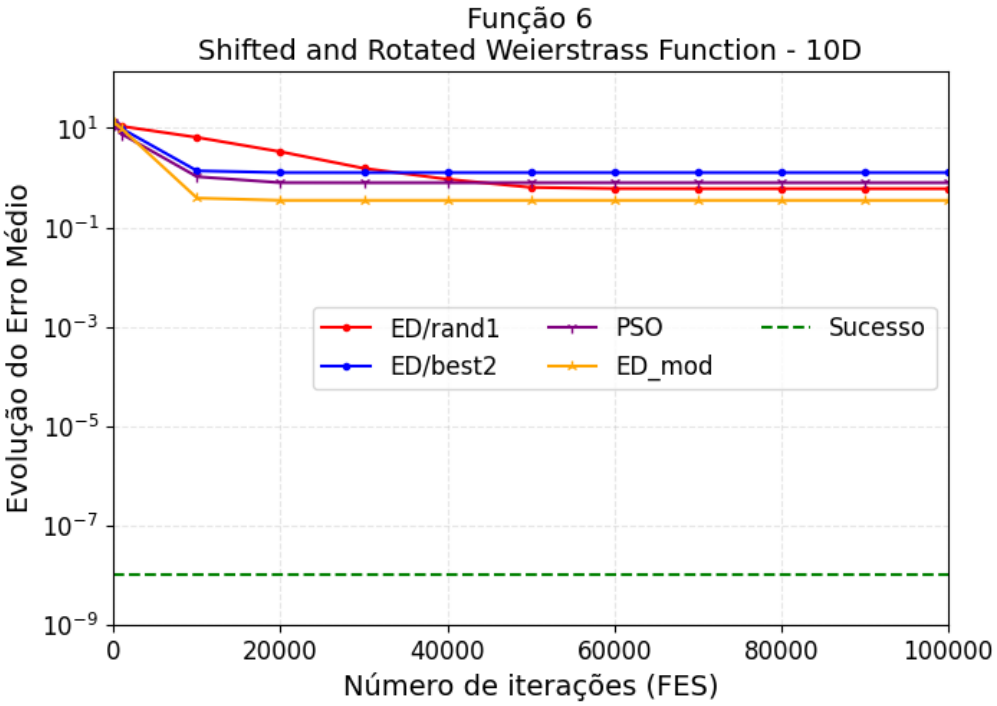

Figura 5: Convergência dos algoritmos para a função objetivo F6 (10D)

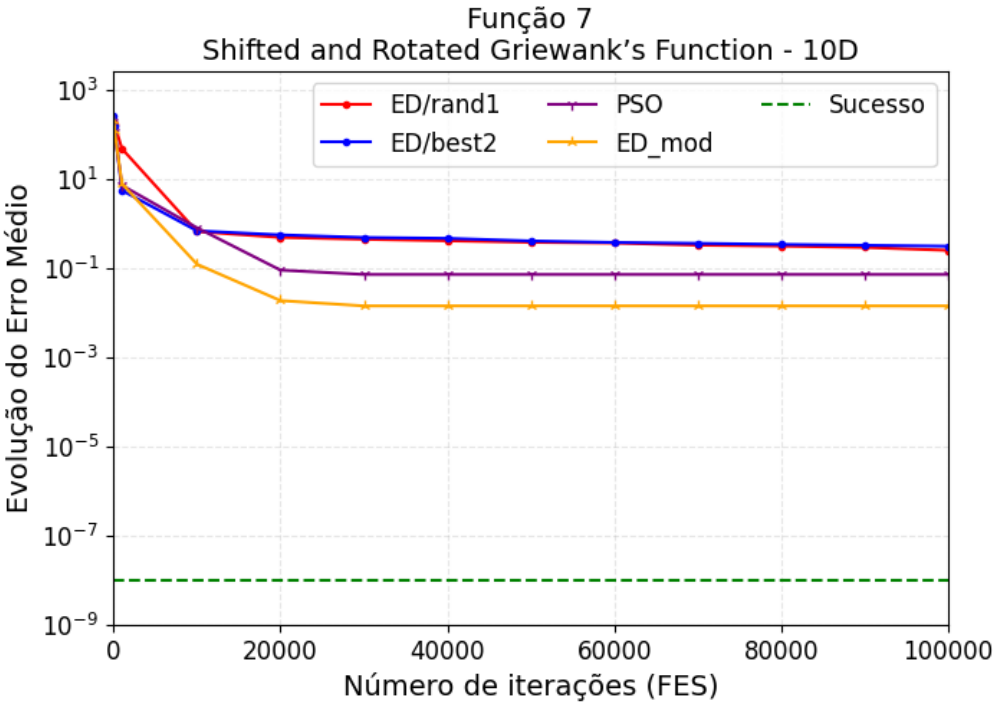

Figura 6: Convergência dos algoritmos para a função objetivo F7 (10D)

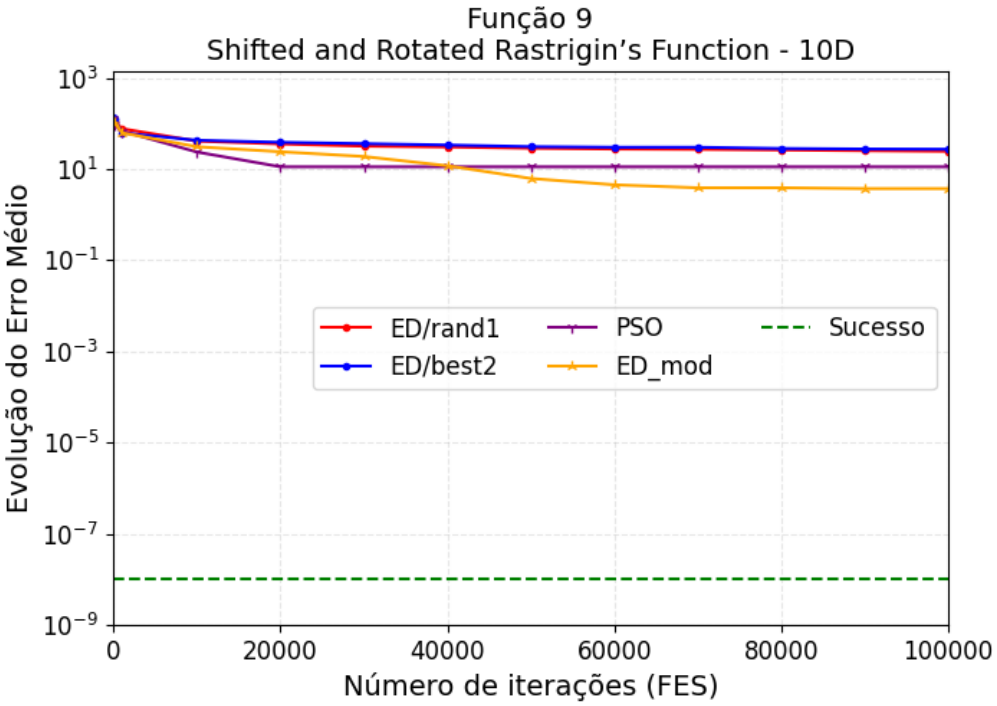

Figura 7: Convergência dos algoritmos para a função objetivo F9 (10D)

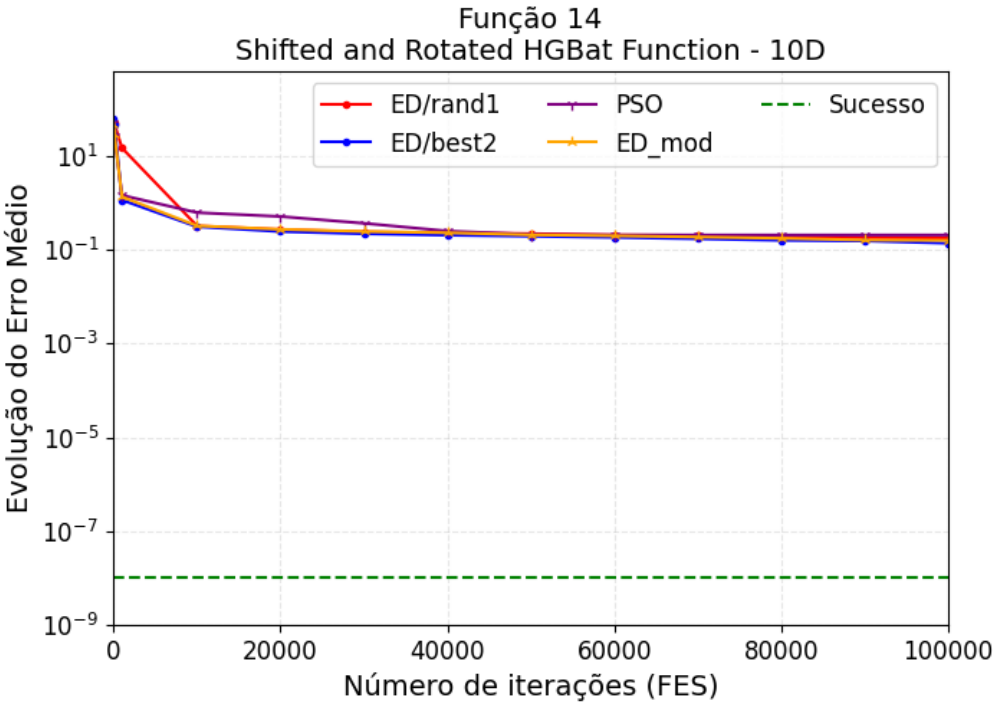

Figura 8: Convergência dos algoritmos para a função objetivo F14 (10D)

Tabela 6: Resultados obtidos pelos algoritmos para as funções objetivos avaliadas (30D)

Função	Método	Melhor	Pior	Mediana	Média	Dev.	Taxa de Sucesso
F1	PSO	1.281E+04	4.487E+05	6.599E+04	9.690E+04	9.569E+04	0.00%
	ED/rand1	4.918E+03	1.146E+05	3.954E+04	3.835E+04	2.360E+04	0.00%
	ED/best2	8.967E+03	1.861E+05	6.078E+04	7.597E+04	4.947E+04	0.00%
	ED/mod	**4.740E+03**	4.344E+04	1.440E+04	2.049E+04	1.282E+04	0.00%
F2	PSO	1.718E-06	3.855E+01	1.871E-01	4.380E+00	8.882E+00	0.00%
	ED/rand1	3.840E-09	9.977E-09	8.517E-09	8.472E-09	1.320E-09	**100.00%**
	ED/best2	7.618E-09	9.955E-09	8.991E-09	9.023E-09	6.755E-10	**100.00%**
	ED/mod	7.678E-09	9.909E-09	9.239E-09	9.104E-09	6.695E-10	**100.00%**
F4	PSO	2.139E-05	7.443E+01	3.101E-02	3.170E+00	1.487E+01	0.00%
	ED/rand1	2.617E-02	6.340E+01	1.453E-01	5.205E+00	1.751E+01	0.00%
	ED/best2	9.519E-09	6.340E+01	6.399E-05	1.046E+01	2.360E+01	8.00%
	ED/mod	8.503E-09	4.424E+00	9.757E-09	1.239E+00	2.027E+00	**72.00%**
F6	PSO	3.369E+00	1.422E+01	6.619E+00	7.064E+00	3.113E+00	0.00%
	ED/rand1	7.690E-09	9.017E+00	4.007E+00	4.180E+00	2.202E+00	**4.00%**
	ED/best2	3.498E+00	1.714E+01	9.113E+00	8.956E+00	3.567E+00	0.00%
	ED/mod	4.962E-01	1.257E+01	5.338E+00	5.479E+00	2.339E+00	0.00%
F7	PSO	2.762E-09	1.620E-01	1.477E-02	2.355E-02	3.724E-02	32.00%
	ED/rand1	6.506E-09	9.986E-09	9.139E-09	8.762E-09	1.015E-09	**100.00%**
	ED/best2	8.507E-09	3.936E-02	9.857E-03	9.555E-03	1.020E-02	40.00%
	ED/mod	8.350E-09	2.461E-02	9.771E-09	5.910E-03	7.946E-03	60.00%
F9	PSO	4.378E+01	1.264E+02	7.661E+01	7.868E+01	2.010E+01	0.00%
	ED/rand1	1.473E+02	1.944E+02	1.737E+02	1.744E+02	1.246E+01	0.00%
	ED/best2	1.791E+02	2.267E+02	2.106E+02	2.078E+02	1.430E+01	0.00%
	ED/mod	**9.950E+00**	5.074E+01	2.487E+01	2.551E+01	8.847E+00	0.00%
F14	**PSO**	**1.651E-01**	9.514E-01	2.738E-01	2.824E-01	1.481E-01	0.00%
	ED/rand1	2.177E-01	3.068E-01	2.705E-01	2.716E-01	2.456E-02	0.00%
	ED/best2	1.795E-01	9.391E-01	6.610E-01	5.507E-01	2.641E-01	0.00%
	ED/mod	1.821E-01	6.122E-01	2.796E-01	3.049E-01	9.294E-02	0.00%

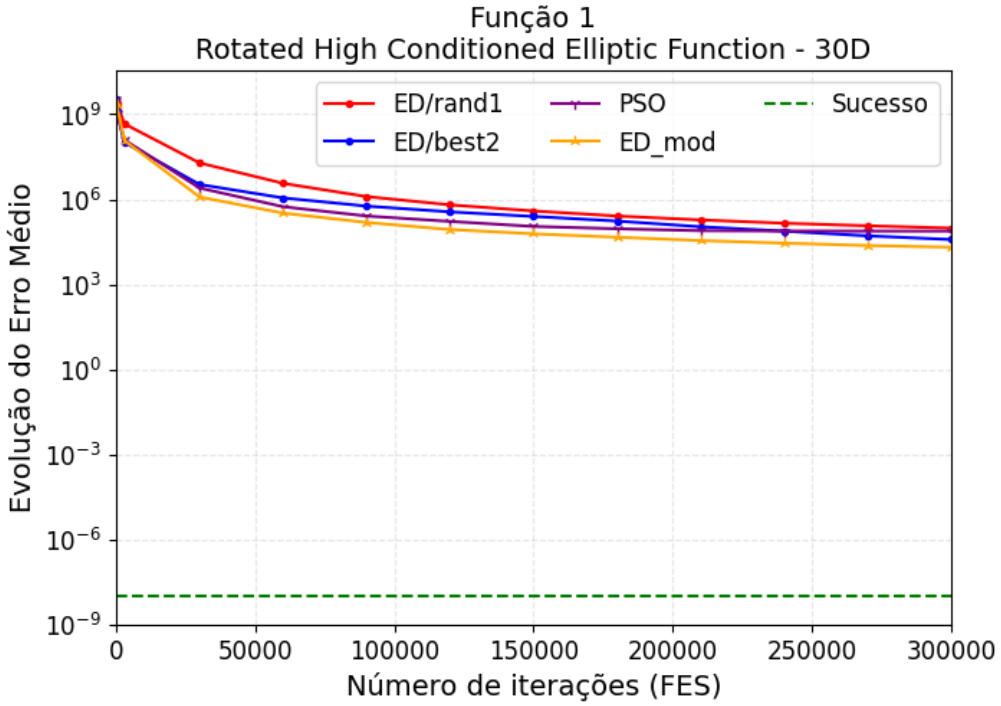

Figura 9: Convergência dos algoritmos para a função objetivo F1 (30D)

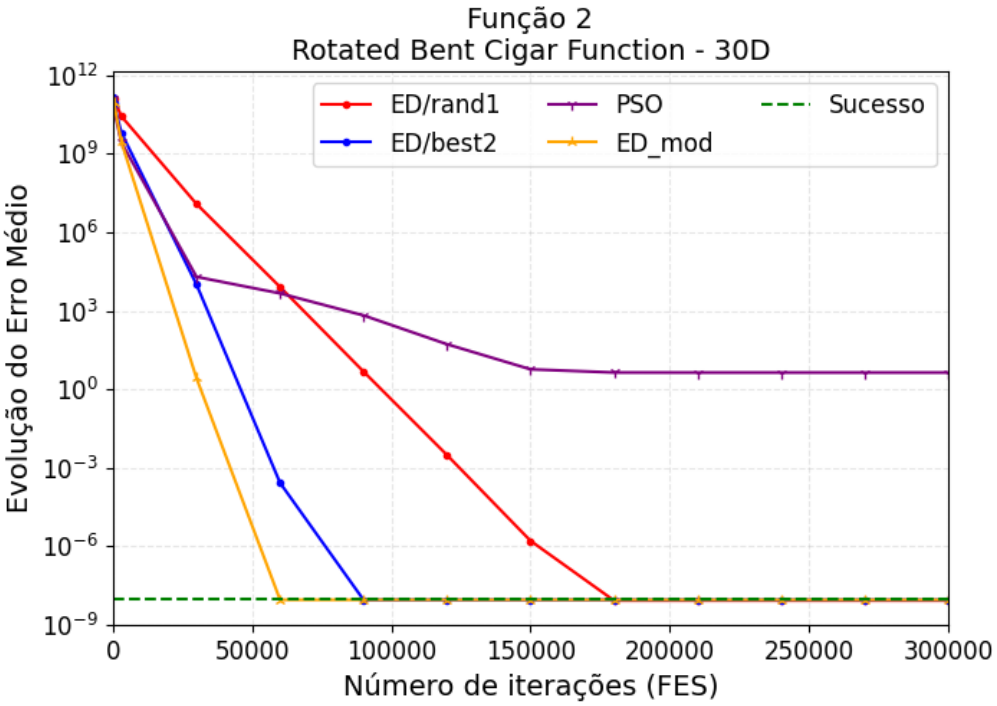

Figura 10: Convergência dos algoritmos para a função objetivo F2 (30D)

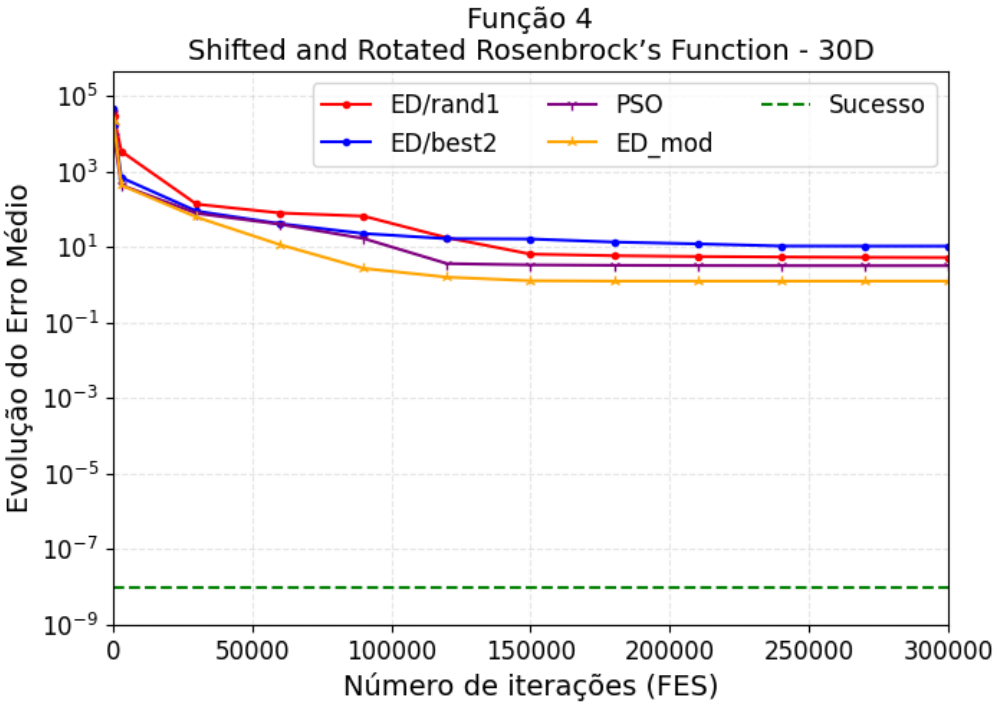

Figura 11: Convergência dos algoritmos para a função objetivo F4 (30D)

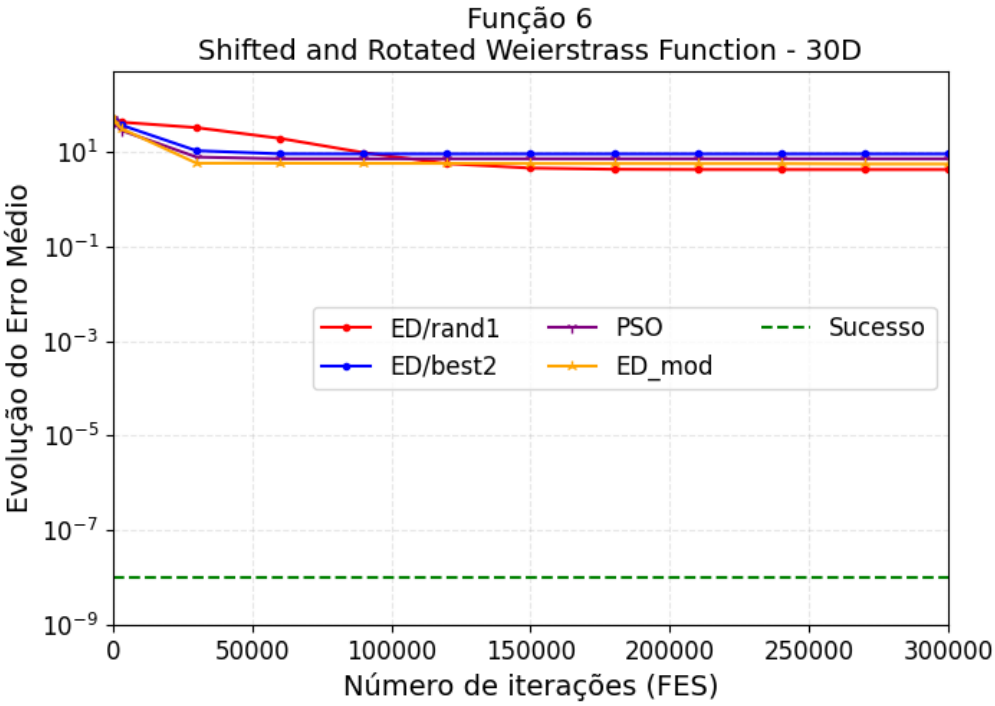

Figura 12: Convergência dos algoritmos para a função objetivo F6 (30D)

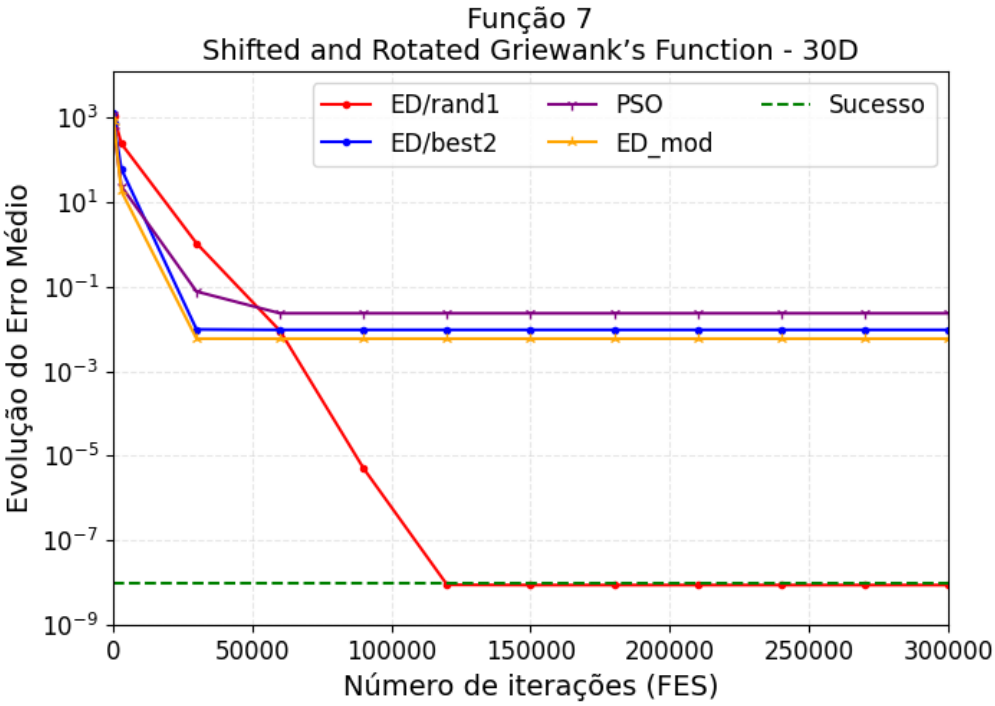

Figura 13: Convergência dos algoritmos para a função objetivo F7 (30D)

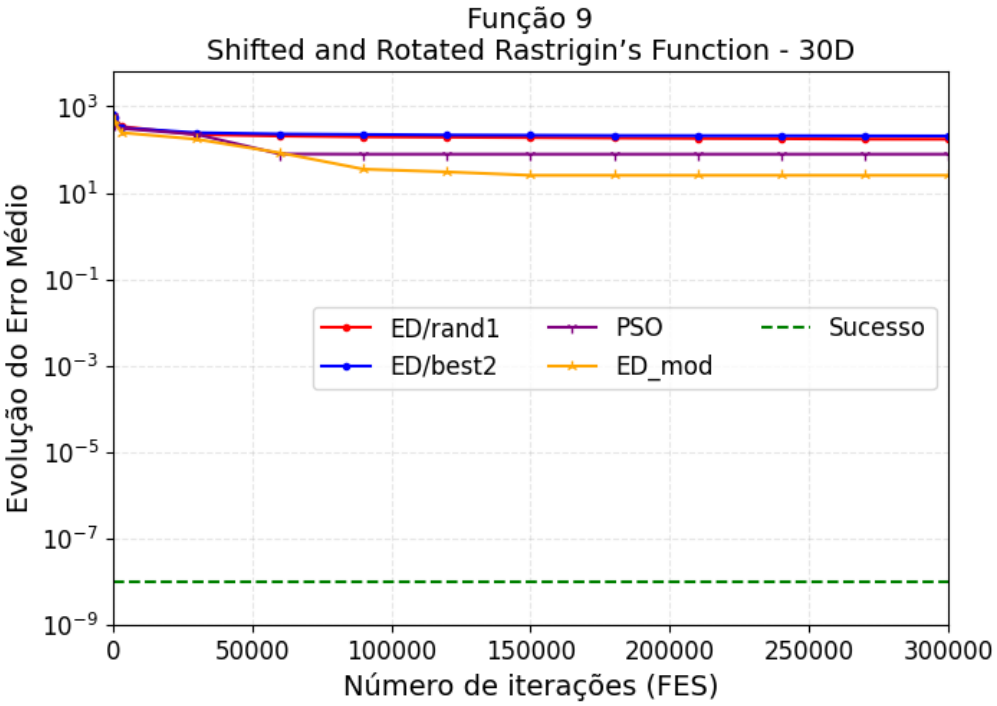

Figura 14: Convergência dos algoritmos para a função objetivo F9 (30D)

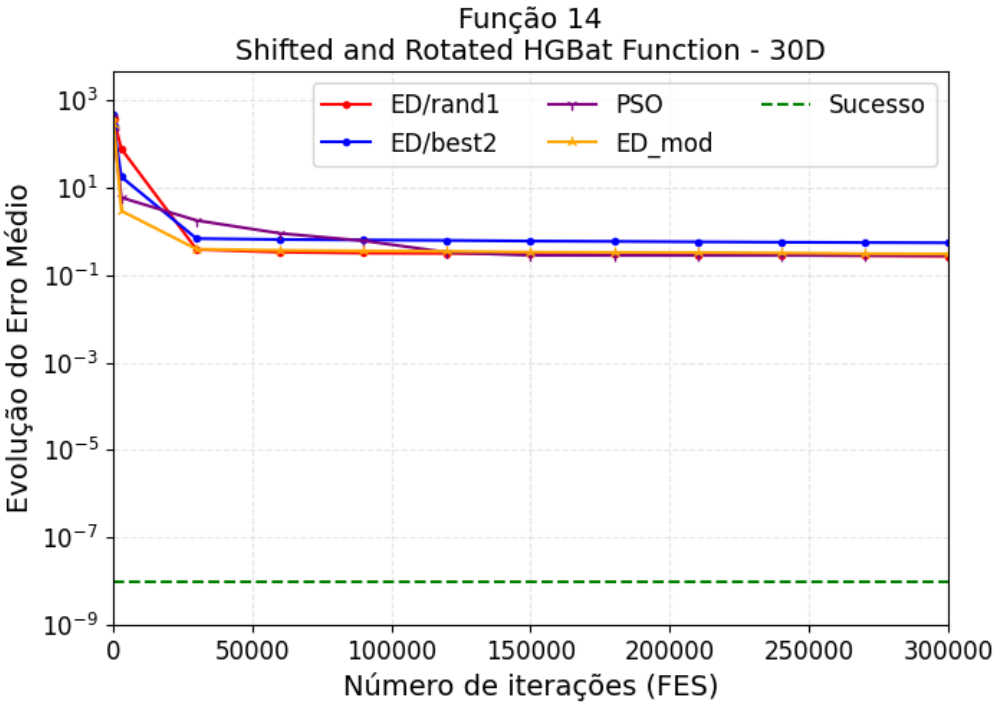

Figura 15: Convergência dos algoritmos para a função objetivo F14 (30D)

Avaliando os resultados apresentados por cada algoritmo, pode-se destacar que:

- O *Particle Swarm Optimization* (PSO) obteve sucessos em 10D (F6) e em 30D (F7). Foi o melhor algoritmo somente na F14 (30D). Considerando as demais funções, seus resultados sãos, em geral, piores comparados aos dos algoritmos clássicos ED/rand/1 e ED/best/2, com exceção para F7 (10D) e F9 (10D e 30D). Quando comparado ao algoritmo ED modificado, seus resultados são piores em todos as demais funções.
- A Evolução Diferencial clássica ED/rand/1 obteve sucessos em 10D (F1, F2, F4 e F6) e 30D (F2, F6 e F7). Foi o melhor algoritmo em F4 (10D), F6 (30D) e F7 (30D). Destaque para o caso da F7 (30D) onde foi o único dos algoritmos a obter 100% de sucesso enquanto o segundo colocado (ED modificado) obteve 60%.
- A Evolução Diferencial clássica ED/best/2 obteve sucessos em 10D (F1, F2, F4 e F6) e 30D (F2, F4 e F7). Foi o melhor algoritmo em F1 (10D) e F2 (10D). Como esperado, sua convergência foi muito mais rápida do que o DE/rand/1 nesses dois casos.
- A Evolução Diferencial modificada obteve sucessos em 10D (F1, F2, F4, F6 e F7) e em 30D (F2, F4 e F7). Foi o melhor dos algoritmos para F1 (30D), F2 (30D), F4 (30D), F6 (10D), F7 (10D), F9 (10D e 30D) e F14 (10D). Destaque na F6 (10D) e F4 (30D) pelo aumento significativo na taxa de sucesso e na F7 (10D) e F9 (10D e 30D) por superar o *Particle Swarm Optimization* onde os algoritmos clássicos ED/rand/1 e ED/best/2 são inferiores. Importante ressaltar que na F1 e F2 (10D) o algoritmo modificado apresentou resultados praticamente idênticos ao melhor algoritmo para essas funções (ED/best/2).

5 Conclusão

Esse relatório apresenta uma série de modificações propostas ao algoritmo conhecido como Evolução Diferencial. O efeito dessas modificações foi avaliado comparando a performance do algoritmo desenvolvido em relação ao modelo clássico ED/rand/1, a variação ED/best/2 e a outro tipo de algoritmo de otimização conhecido como *Particle Swarm Optimization* (PSO). A capacidade desses algoritmos em encontrar o mínimo global foi analisado em sete problemas de otimização conhecidos na literatura (*benchmark functions*).

Considerando os resultados obtidos, o algoritmo modificado foi o que obteve a melhor performance dentre os avaliados. Além de ser o algoritmo que mais superou os concorrentes, em apenas 1 dos 14 casos avaliados foi, ligeiramente, inferior aos dois modelos clássicos de Evolução Diferencial. Tendo isso em mente, conclui-se que a ideia de aproveitar a eficiência do ED/best/2 e a capacidade de exploração do ED/rand/1 em um mesmo algoritmo, assim como a proposta de reinicializar a população em casos de convergência prematura, foram bem sucedidas.

É importante ressaltar que os resultados apresentados aqui abrangem uma pequena gama de funções objetivo sendo recomendável que novos testes sejam realizados com uma maior diversidade de funções.

Referências

[1] Larangeira, V. d. A, 2020, “Um estudo sobre a performance de diferentes algoritmos de computação evolutiva”, 1º Relatório referente a disciplina de Métodos Computacionais Inspirados na Natureza (MCIN)

[2] Kennedy, J., Eberhart, R. C., 1995, “Particle swarm optimization”, Proceedings of the EEE international Conference on neural networks IV (pp. 1942–1948)

[3] Clerc, M., 1999, “The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization”, Congress on Evolutionary Computation, Washington DC (pp. 1951–1955)

[4] Liang, J. J., Suganthan, P. N., 2005, “Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search”, Proceedings for CEC2005 (web page: https://github.com/P-N-Suganthan /CEC2005 /blob/master/papers-with-results.zip)

[5] Tasgetiren, M. F., Liang, Y., Gencyilmaz, G., Eker, I., 2005, “Global Optimization of Continuous Functions using Particle Swarm Optimization”, Proceedings for CEC2005 (web page: https://github.com/P-N-Suganthan /CEC2005 /blob/master/papers-with-results.zip)

[6] Wang, D, Tan, D., Liu, L., 2018, “Particle swarm optimization algorithm: an overview”, Soft Comput (2018) 22 (pp: 87–408)

[7] Cuevas, E., Cienfuegos, M., Zaldívar, D., Pérez-Cisneros, M., 2013, “A swarm optimization algorithm inspired in the behavior of the social-spider”, Expert Systems with Applications 40 (pp. 6374–6384).

[8] Karaboga, D., Basturk, B., 2007, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, J Glob Optim 39 (pp: 459–471)

[9] Storn, R. e Price, K., 1995, “Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces”, Technical Report TR-95-012, ICSI

[10] Qin, A. K., Suganthan, P. N., 2005, “Self-adaptive Differential Evolution Algorithm for Numerical Optimization”, Proceedings for CEC2005 (web page: https://github.com/P-N-Suganthan/CEC2005 /blob/master/papers-with-results.zip)

[11] Pan, Q., Suganthan, P. N., Wang, L., Gao, L., Mallipeddi, R., 2011, “A differential evolution algorithm with self-adapting strategy and control parameters”, Computers & Operations Research 38 (pp: 394–408)

[12] Eltaieb, T, Mahmood, A, 2018, “Differential Evolution: A Survey and Analysis”, Applied Sciences