(mu+lamda)–ES and other ES algorithms

10 benchmark function

CJ Chung

Fall 2025

ESalgorithms10funcs.ipynb

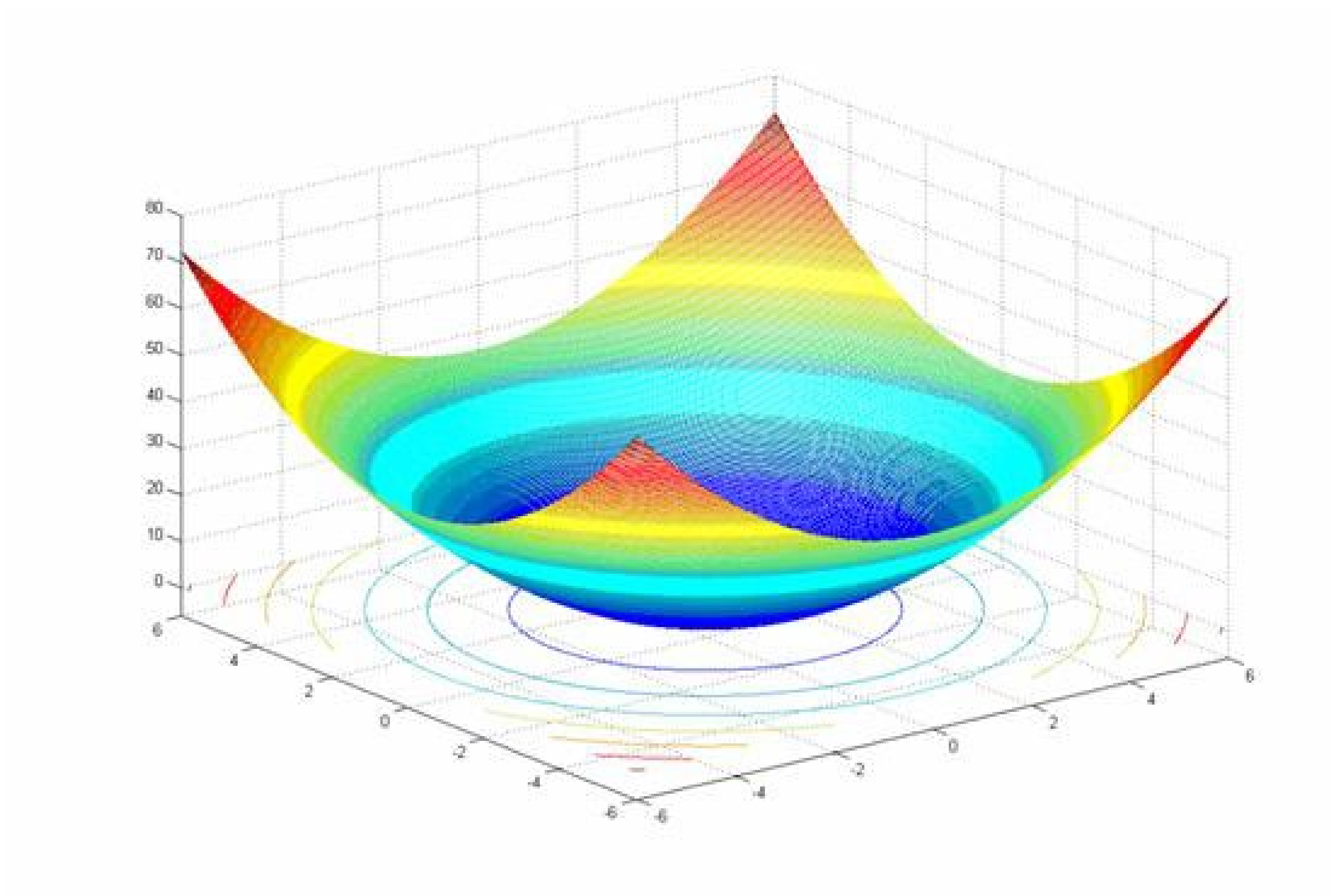
- (1,1) random search
- (1+1) with 1/5 success rule
- (mu+lamda) with 1/5 success rule
- (mu+lamda) with 1/5 success rule + variance adaptation

n-dimensional 10 Benchmark functions

Search range   acceptable global fitness, f*(**x**)

```
functions = [
  ('Sphere', sphere, [-5.12, 5.12], 1e-5),             # Easy - should find near-optimal
  ('Rosenbrock', rosenbrock, [-2.048, 2.048], 1e-1), # Medium - curved valley makes exact optimum hard
  ('Rastrigin', rastrigin, [-5.12, 5.12], 1e0),       # Hard - many local optima
  ('Ackley', ackley, [-32.768, 32.768], 1e-1),        # Medium - many local optima but well-structured
  ('Griewank', griewank, [-600, 600], 1e-1),          # Medium - regular local optima pattern
  ('Schwefel', schwefel, [-500, 500], 1e1),           # Very hard - deceptive landscape
  ('Lunacek BiRstrgn', lunacek_bi_rastrigin, [-5.12, 5.12], 1e0), # Hard - deceptive double funnel
  ('Levy', levy, [-10, 10], 1e-1),                    # Medium - rugged but manageable
  ('Zakharov', zakharov, [-5, 5], 1e-5),              # Easy - unimodal with interactions
  ('Hybrid Composition', hybrid_composition, [-5, 5], 1e0)       # Hard - mixed characteristics
]
```

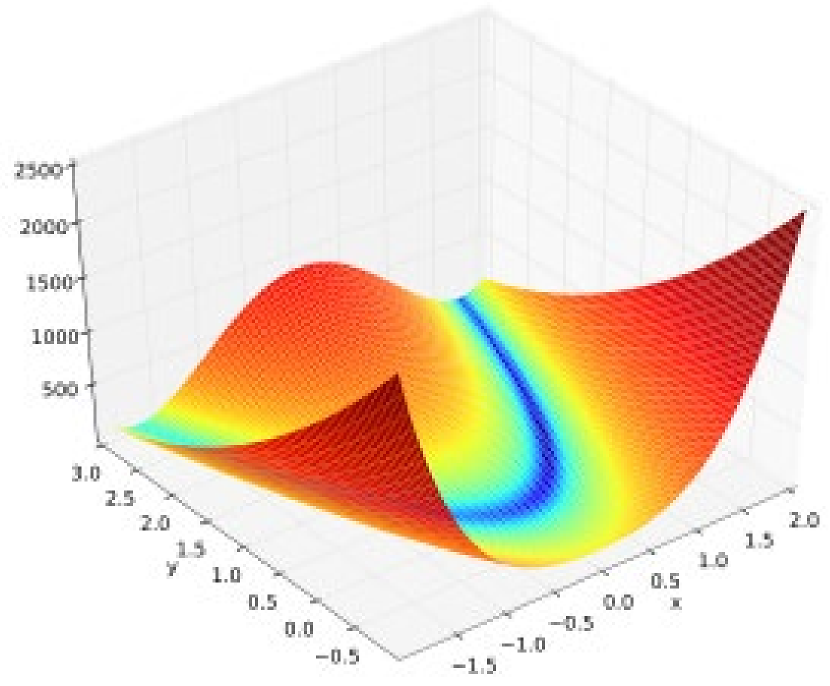
1. Sphere function



$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$$

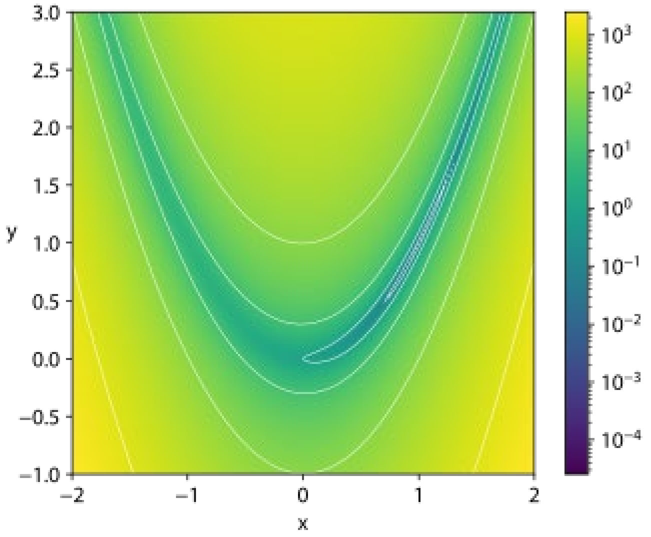
$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \dots, 0)$$

2. Rosenbrock

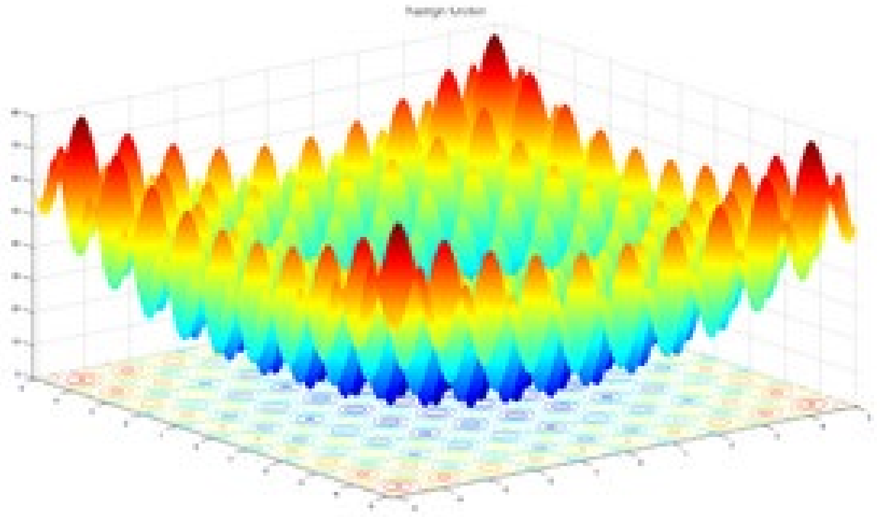


$$f(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (1, \dots, 1)$$

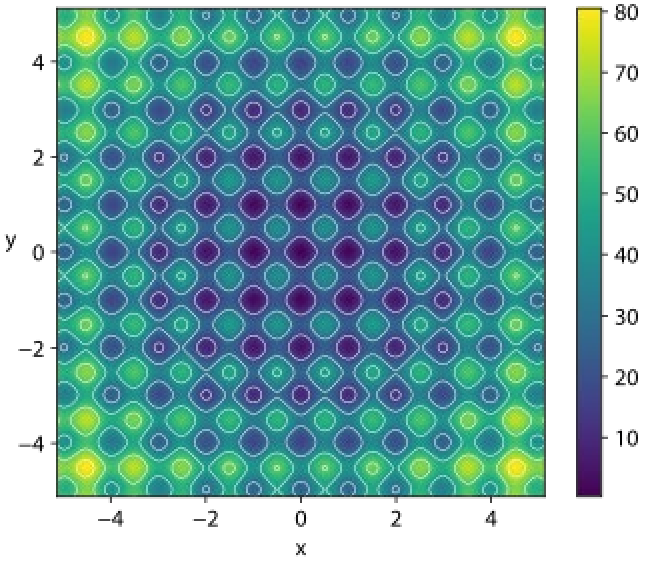


3. Rastrigin

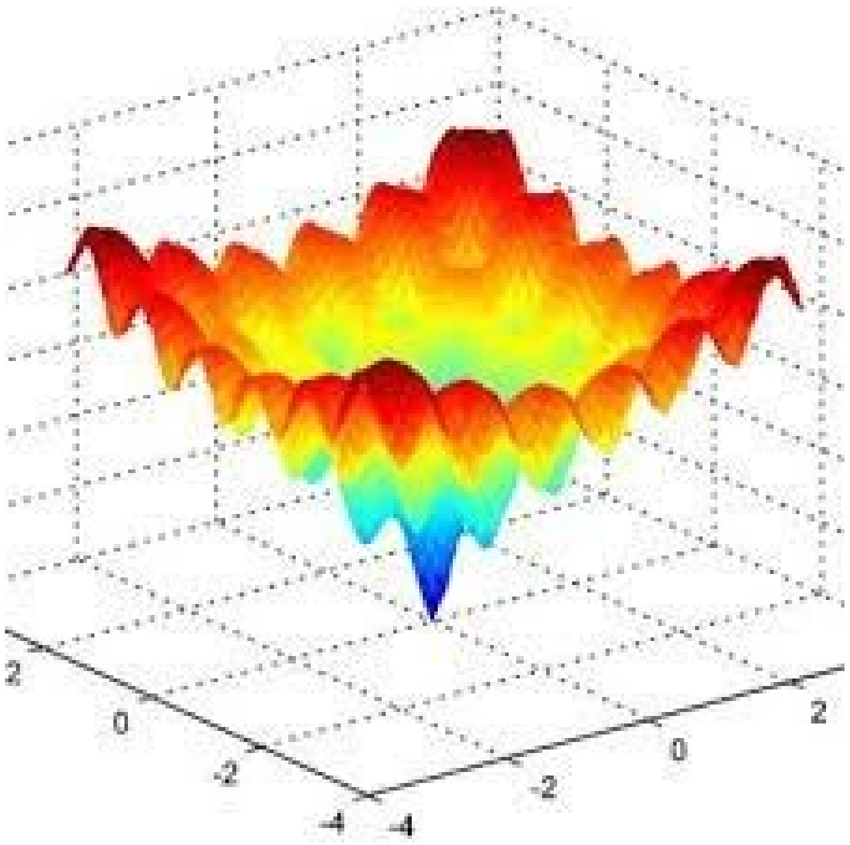


$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right]$$

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \dots, 0)$$

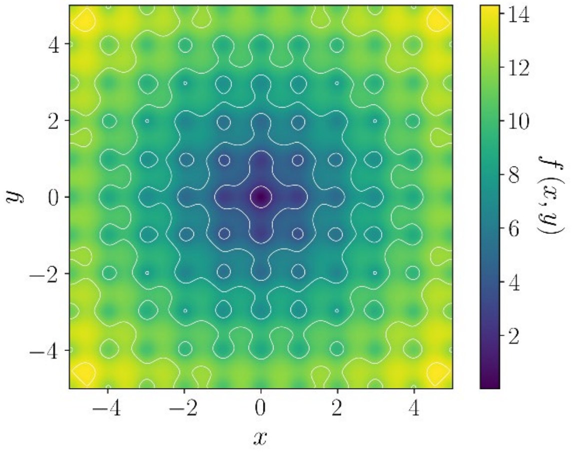


4. Ackley

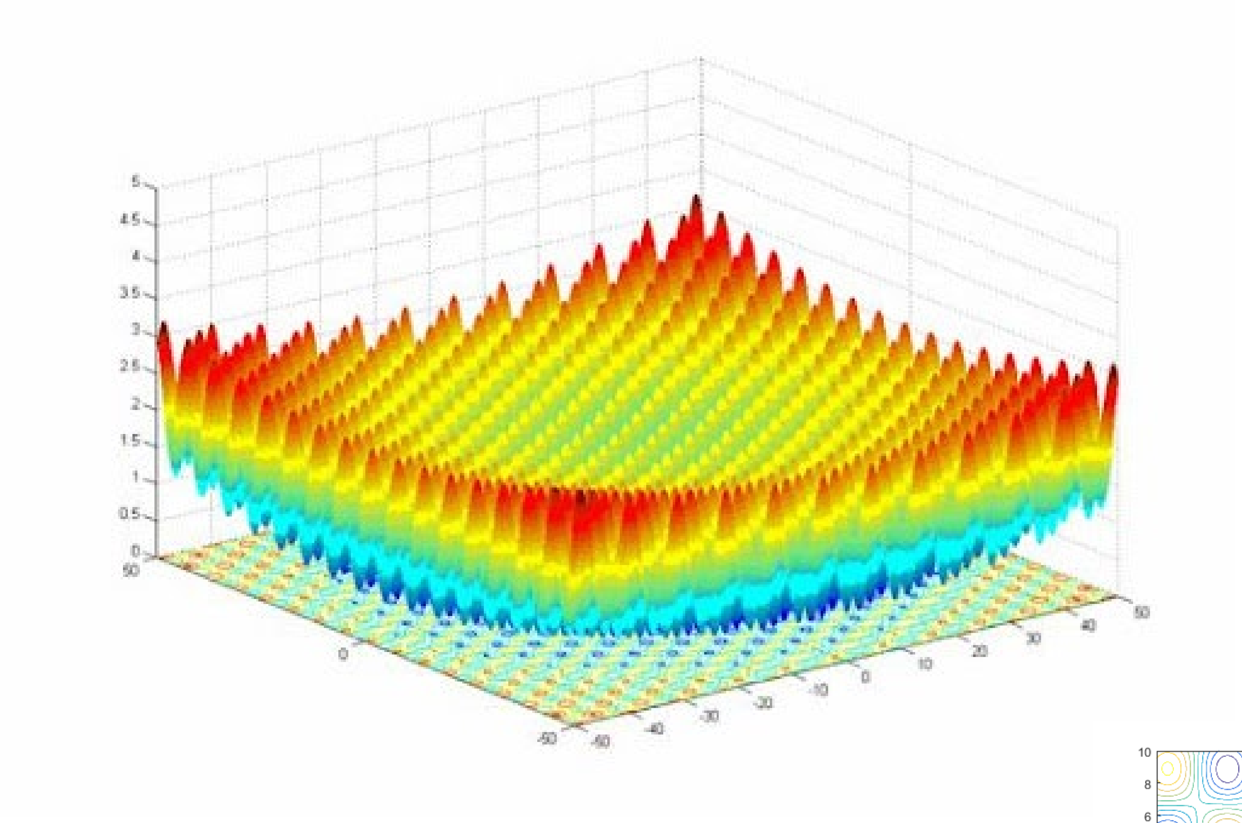


$$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$$

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \ldots, 0)$$

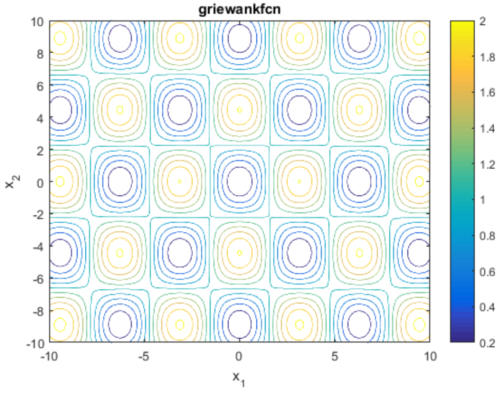


5. Griewank

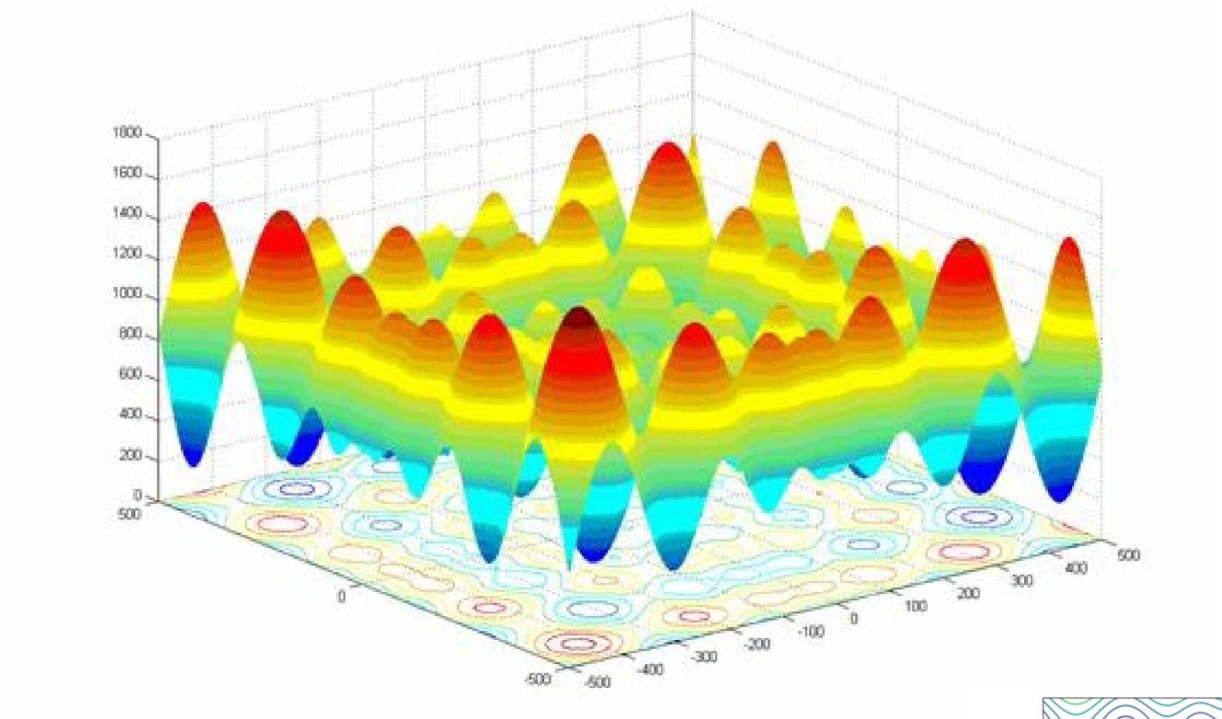


$$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \dots, 0)$$

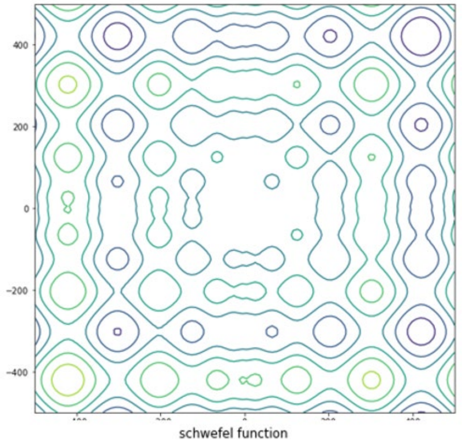


6. Schwefel (very hard)



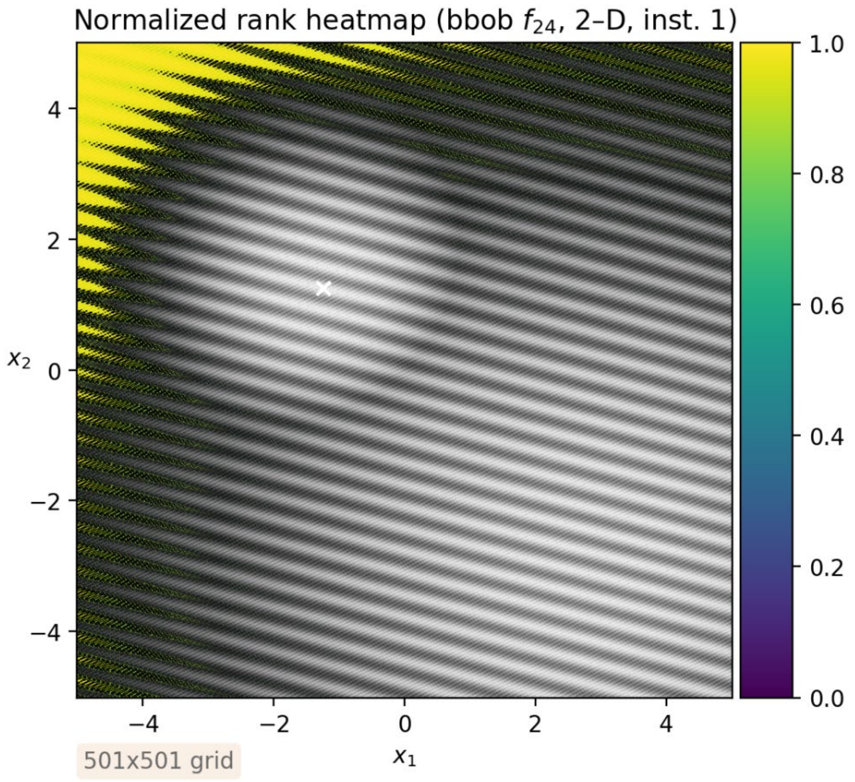
$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|})$$

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (420.9687, \dots, 420.9687)$$



7. Lunacek Bi Rastrigin

A particularly deceptive benchmark function with a **double-funnel structure** that makes finding the global optimum very challenging.



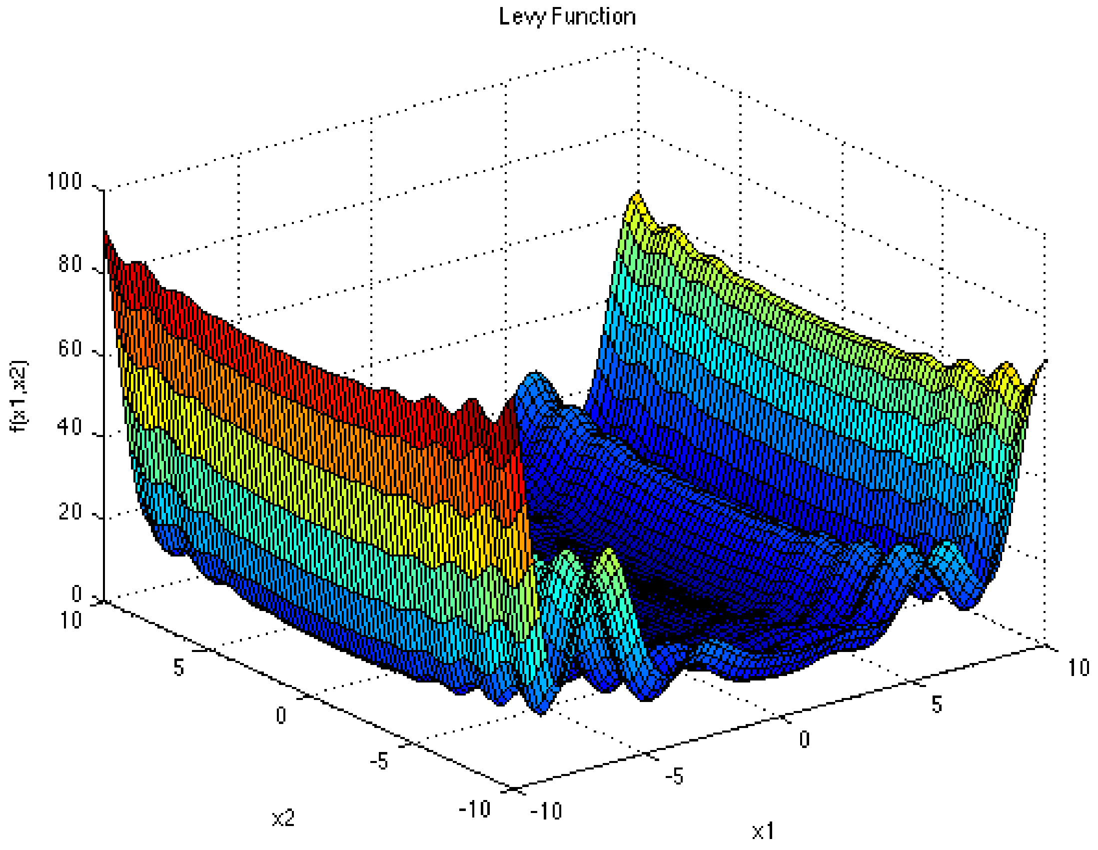
$$f(X) = \min\left[\sum_{i=1}^{n}(x_i - \mu_1)^2\ ,\ d \cdot n + s \cdot \sum_{i=1}^{n}(x_i - \mu_2)^2\right] + 10\sum_{i=1}^{n}\left\{1 - \cos[2\pi\,(x_i - \mu_1)]\right\}$$

$$f(\mathbf{x}^*) = 0$$

8. Levy

$$f(\mathbf{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2\left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_d - 1)^2\left[1 + \sin^2(2\pi w_d)\right], \text{ where}$$

$$w_i = 1 + \frac{x_i - 1}{4}, \text{ for all } i = 1, \dots, d$$

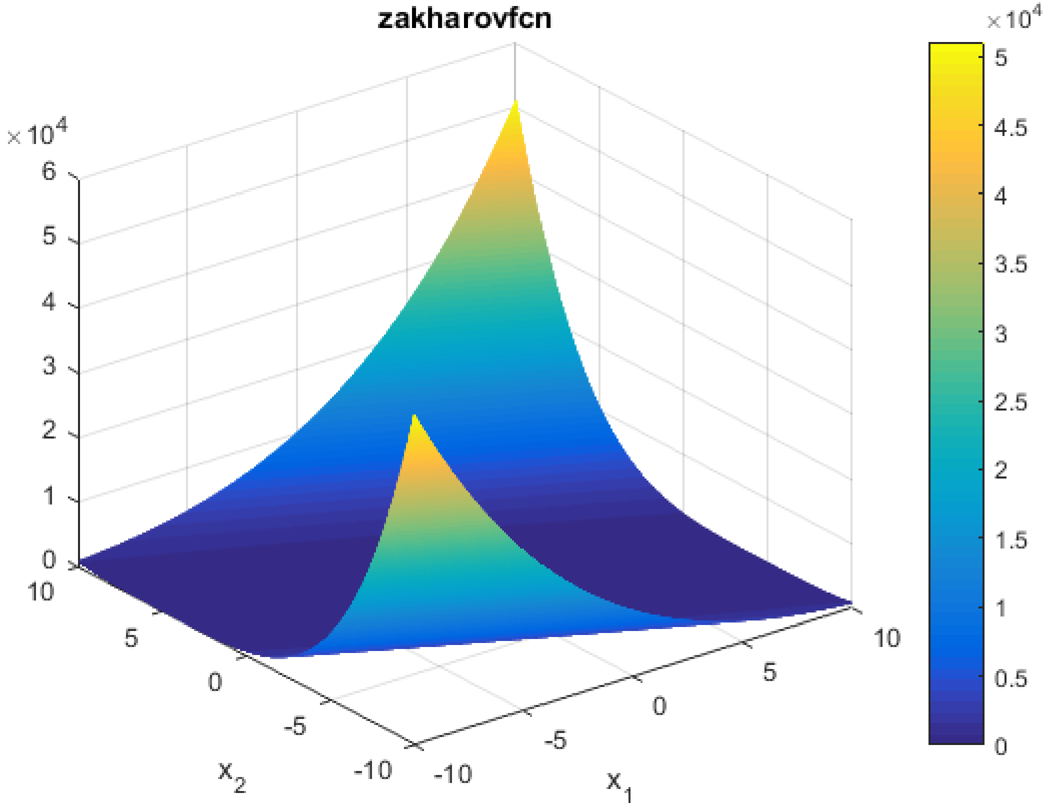


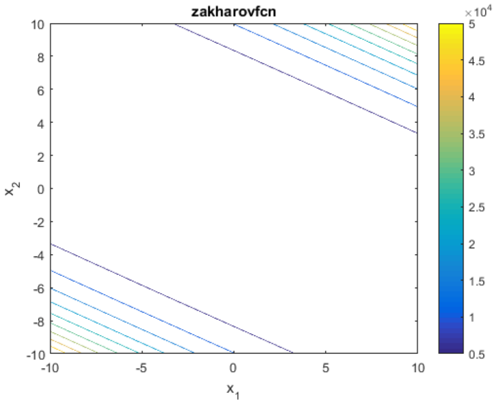
$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (1, \dots, 1)$$

9. Zakharov

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i)^2 + (\sum_{i=1}^{n} 0.5 i x_i)^4$$

$$f(\mathbf{x}^*) = 0 \text{ at } \mathbf{x}^* = (0, \ldots, 0)$$





10. Hybrid Composition

- Position: x* = [0, 0, ..., 0] (for all dimensions)
- Value: f(x*) = 0
- Challenge: The optimum requires optimizing ALL three different function types simultaneously

Region	Function Type	Characteristics	Difficulty
Part 1	Sphere	Unimodal, convex, smooth	Easy
Part 2	Rastrigin	Highly multimodal, many local optima	Hard
Part 3	Rosenbrock	Unimodal but ill-conditioned valley	Medium

(1,1) random search – simple **baseline** algorithm

- In each iteration until the maximum number of evaluations is reached., a new random solution (using uniform random numbers) is generated within the defined bounds.
- If the new solution has a better fitness, it replaces the current best solution and fitness.
- The best fitness found so far is recorded in a history list.
- Return: Finally, the function returns the best solution found, its fitness, and the history of the best fitness over the evaluations.

(1+1) with 1/5 success rule

- Rewritten using linear alebra style using Numpy arrays without using loops for each variable. → Simpler and shorter

(mu+lamda) with 1/5 success rule

- Mu=5, Lamda=15
- sigma, a variable
- success_count: any offspring that are better than their parent and made it into the new population
- Every WINDOW generations, calculate

  success_rate = success_count / adaptation_interval

Variance = spread (in squared units)

Variance

- Variance measures **how spread out** a set of numbers is from the mean.
- Formula (population variance):

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

where

- N = number of data points
- x_i = each data point
- μ = mean (average)

So variance is the **average of squared deviations** from the mean.

(mu+lamda) with 1/5 success rule + **variance**

- Adapts a sigma vector (*sigma for each variable*) based on overall success rate and offspring variance
- The following has rooms for improvement

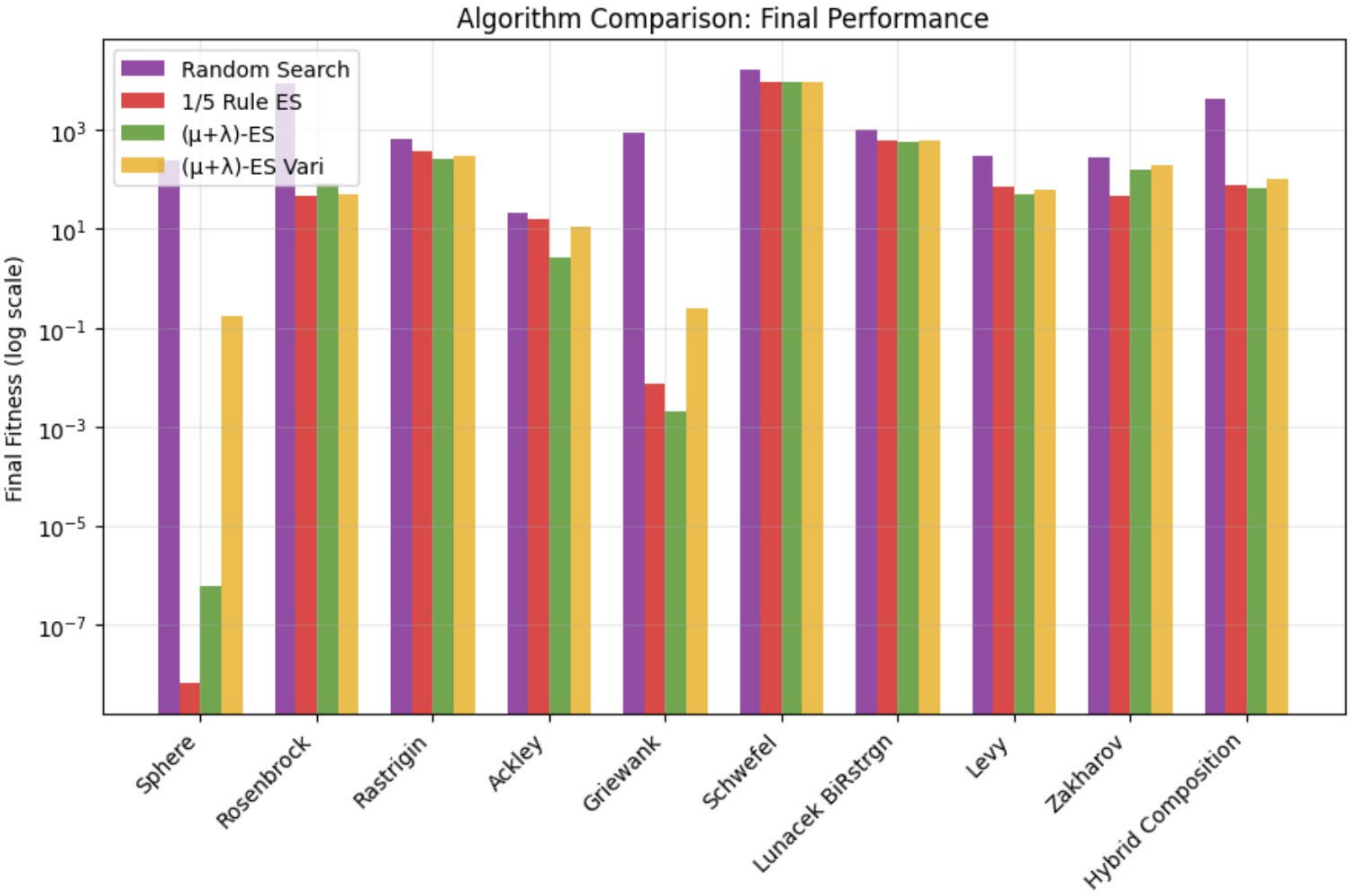
```
if success_rate > target_success_rate:
    sigma = sigma * (1.1 + 0.1 * (variance_per_dim / np.mean(variance_per_dim)))
    # Increase more for higher variance
else:
    sigma = sigma * (0.9 - 0.1 * (variance_per_dimension / np.mean(variance_per_dim)))
    # Decrease more for lower variance
```

Results

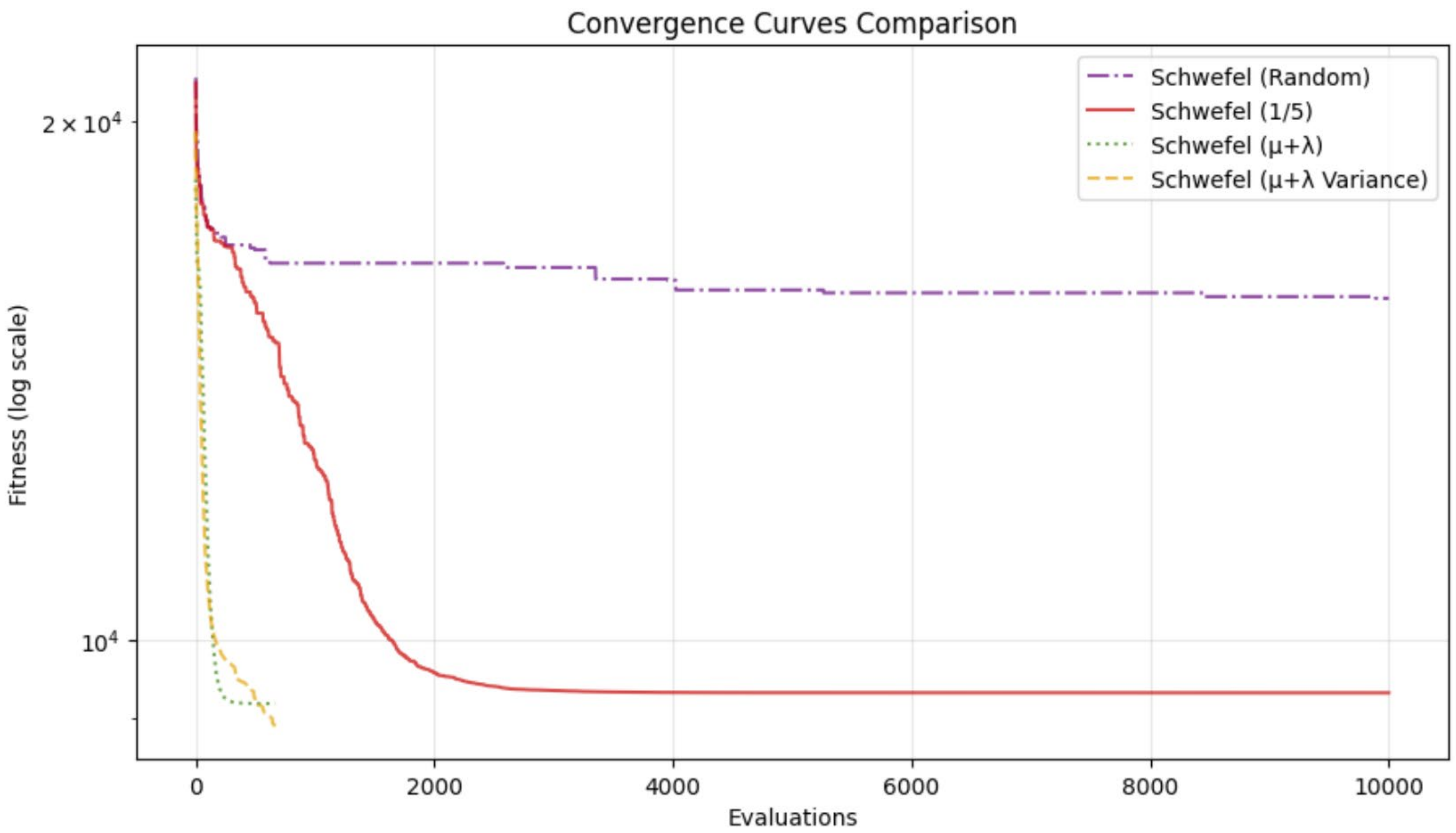
```python
if __name__ == "__main__":
    main(dim=2, max_evaluations=1000, runs=5)
    main(dim=5, max_evaluations=2000, runs=5)
    main(dim=10, max_evaluations=3000, runs=5)
    main(dim=20, max_evaluations=5000, runs=5)
    main(dim=50, max_evaluations=10000, runs=5)
```

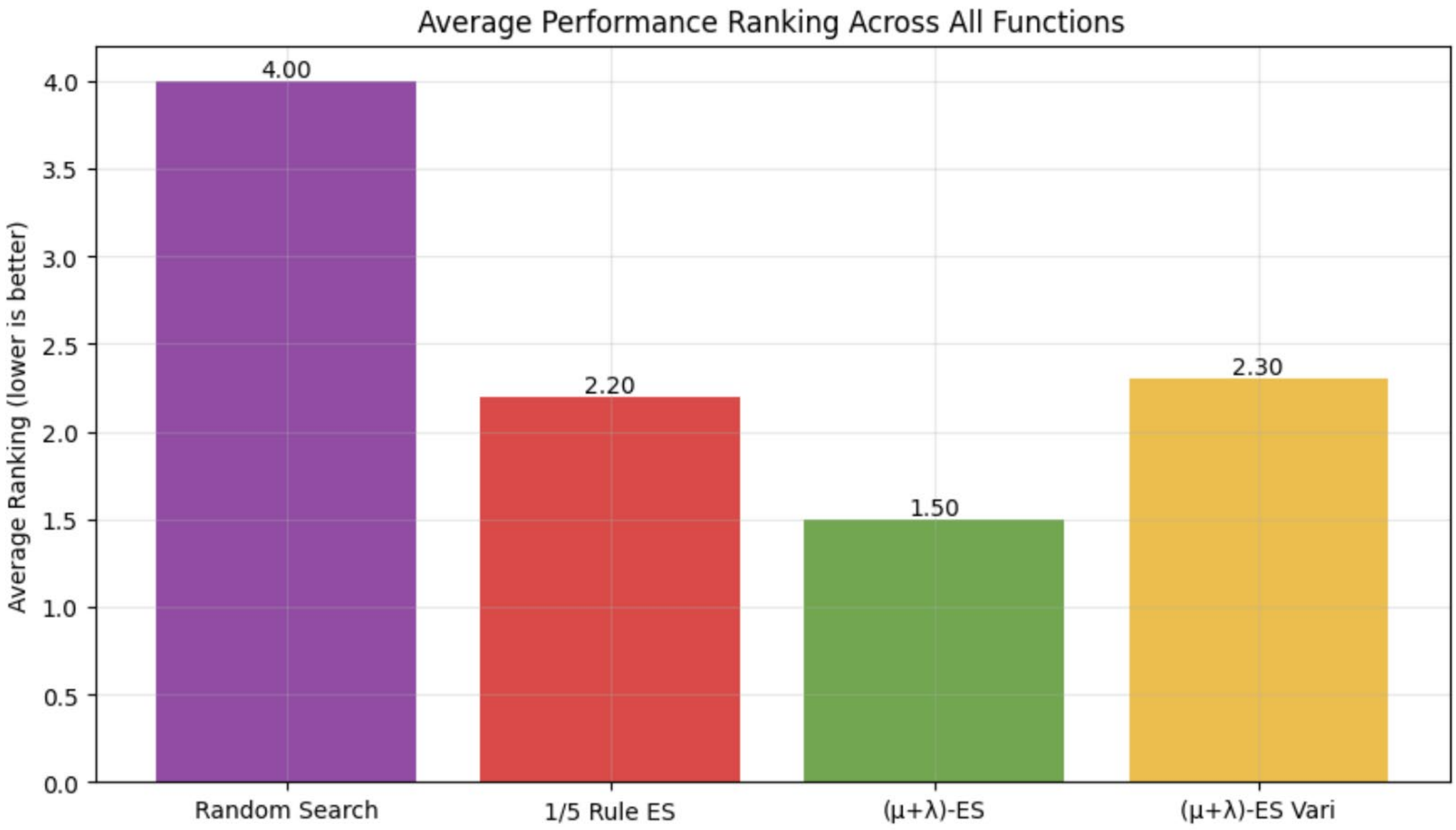
D=50 ALGORITHM COMPARISON SUMMARY

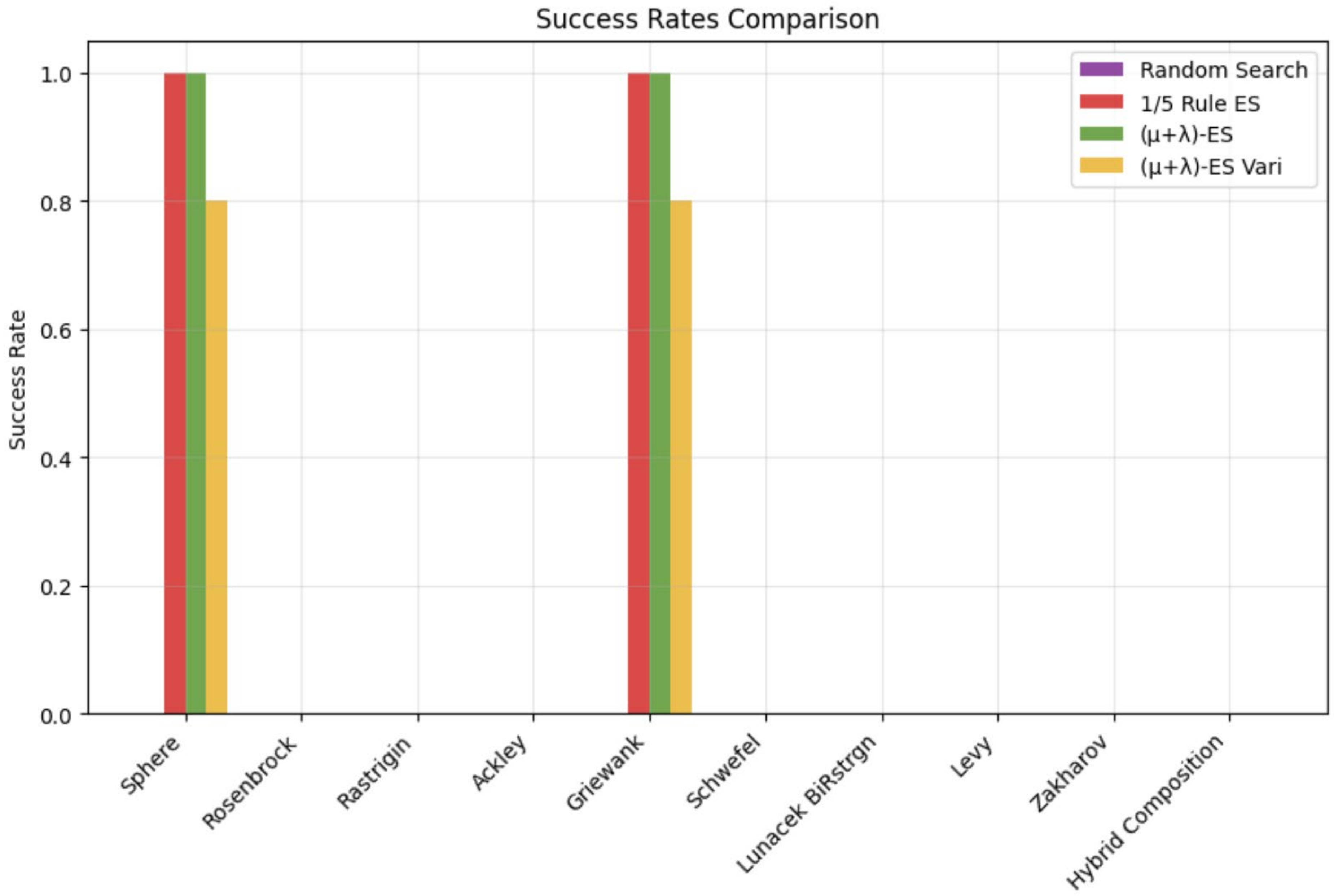
Function	Random Search	1/5 Rule ES	(μ+λ)-ES	(μ+λ)-ES Vari	Winner
Sphere	2.439536e+02	6.630599e-09	5.892409e-07	1.778690e-01	1/5 Rule ES
Rosenbrock	8.319194e+03	4.540386e+01	8.091125e+01	4.942365e+01	1/5 Rule ES
Rastrigin	6.612240e+02	3.613667e+02	2.578922e+02	2.949916e+02	(μ+λ)-ES
Ackley	2.040477e+01	1.607866e+01	2.717542e+00	1.063256e+01	(μ+λ)-ES
Griewank	8.349578e+02	7.392846e-03	2.137159e-03	2.545383e-01	(μ+λ)-ES
Schwefel	1.577321e+04	9.312351e+03	9.178924e+03	8.905477e+03	(μ+λ)-ES Vari
Lunacek BiRstrgn	9.838902e+02	6.110706e+02	5.518346e+02	5.881268e+02	(μ+λ)-ES
Levy	2.857138e+02	6.800644e+01	4.868285e+01	6.156526e+01	(μ+λ)-ES
Zakharov	2.769966e+02	4.634503e+01	1.517428e+02	1.848003e+02	1/5 Rule ES
Hybrid Composition	4.274938e+03	7.602719e+01	6.636009e+01	1.023113e+02	(μ+λ)-ES
TOTAL WINS	0	3	6	1	
OVERALL WINNER					(μ+λ)-ES



Algorithm Comparison: Final Performance
Random Search
1/5 Rule ES
(μ+λ)-ES
(μ+λ)-ES Vari
Final Fitness (log scale)
10^3
10^1
10^-1
10^-3
10^-5
10^-7
Sphere
Rosenbrock
Rastrigin
Ackley
Griewank
Schwefel
Lunacek BiRstrgn
Levy
Zakharov
Hybrid Composition


Convergence Curves Comparison
Schwefel (Random)
Schwefel (1/5)
Schwefel (μ+λ)
Schwefel (μ+λ Variance)
Fitness (log scale)
2 × 10^4
10^4
Evaluations
0
2000
4000
6000
8000
10000


Average Performance Ranking Across All Functions
Average Ranking (lower is better)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
4.00
2.20
1.50
2.30
Random Search
1/5 Rule ES
(μ+λ)-ES
(μ+λ)-ES Vari


Success Rates Comparison
Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
Random Search
1/5 Rule ES
(μ+λ)-ES
(μ+λ)-ES Vari
Sphere
Rosenbrock
Rastrigin
Ackley
Griewank
Schwefel
Lunacek BiRstrgn
Levy
Zakharov
Hybrid Composition