EVOLUTIONARY ALGORITHMS 2025 - REPORT - GROUP 50
GENETIC ALGORITHMS AND EVOLUTIONARY STRATEGIES

Kevin Bretz **Federico Cucinotta**

July 9, 2026

ABSTRACT

This report details the implementation of a Memetic Genetic Algorithm (GA) and a Diagonal Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to solve benchmark problems from the Pseudo-Boolean Optimization (PBO) and Black-Box Optimization Benchmarking (BBOB) suites. Our GA utilizes a First-Improvement Hill Climber to refine Building Blocks, while our ES employs diagonal matrix updates and Cumulative Step-size Adaptation (CSA) to navigate the fractal Katsuura landscape. For optimizing the GA hyperparameters, we used a 100,000 evaluation tuning budget with Optuna and Hyperband pruning to identify robust settings for a strict 5,000 evaluation per-run limit. Results demonstrate that the GA achieved an average best fitness value of 6.25 on the N-Queens problem (F23) while simultaneously reaching a 4.03 merit factor on the Low Autocorrelation Binary Sequences (LABS, F18) problem. Furthermore, our ES implementation achieved a highly competitive final fitness of 0.5 on the 10-dimensional Katsuura problem.

1 Introduction

This report explores the application of evolutionary heuristics to complex discrete and continuous optimization landscapes. The primary objective was to achieve high evaluation efficiency within a strict limit of 5,000 function evaluations per independent run.

1.1 GA Implementation and Encoding (Task 1)

For the Pseudo-Boolean Optimization (PBO) suite, we developed a hybrid framework to solve the **Low Autocorrelation Binary Sequences (LABS, F18)** problem and the **N-Queens problem (F23)**.

We utilized a binary representation where individuals are encoded as bitstrings $x \in \{0,1\}^n$. For the N-Queens problem, the phenotype represents queen positions on a grid, encoded into a 49-bit chromosome. To navigate these landscapes, we chose a **Memetic Genetic Algorithm**. This implementation combines the global search capabilities of a Generational GA with a **First-Improvement Hill Climber (Local Search)**. This hybrid approach was chosen because standard optimizers frequently stagnate on the rugged LABS landscape; our implementation addresses this by triggering the local search operator upon population stagnation to aggressively refine high-fitness schemata [1].

Regarding variation and selection, we employed **Uniform Crossover** and **Tournament Selection (size 7)**. Uniform crossover was selected because it outperforms two-point crossover in a direct comparison. All other GA hyperparameters were selected through a dedicated tuning script.

1.2 Hyperparameter Tuning Strategy

The tuning process utilized a total budget of **100,000 evaluations**. To optimize the allocation of this budget, we employed **Bayesian Optimization** via the **Optuna** framework [2] with a **Tree-structured Parzen Estimator (TPE)** sampler.

To maximize tuning efficiency, we implemented a **Hyperband Pruner** for early discarding. This allowed the tuner to monitor configurations at intermediate steps and terminate underperforming trials early, effectively allowing us to explore a significantly higher number of unique configurations within the budget compared to a standard grid search. To comply with the requirement of using identical settings for both problems, we maximized a normalized combined fitness score. The process revealed that a smaller population size ($\mu = 46$) was superior for the restricted 5,000-evaluation limit, as it allowed for a greater number of evolutionary iterations.

1.3 Evolution Strategy for BBOB (Task 2)

For Task 2, we addressed the 10-dimensional **Katsuura problem (BBOB F23)** [3], which is characterized by extreme multi-modality and fractal noise. Based on the principle that forgetting is necessary to prevent stagnation in highly rugged landscapes, we implemented a $(\mu/\mu_W, \lambda)$**-CMA-ES** utilizing non-elitist comma selection [4].

To maximize performance within the limited budget, we utilized a **Diagonal Covariance Matrix Adaptation**. By ignoring off-diagonal correlations, we reduced the degrees of freedom the internal model must learn, allowing for faster distribution adaptation. Step-size control was handled through **Cumulative Step-size Adaptation (CSA)**, and we implemented a multi-start strategy to allow the ES to escape the deep local optima basins that typically trap standard search heuristics.

2 Algorithms

In this work, we implement two distinct evolutionary frameworks designed for evaluation efficiency. For Task 1, we use a hybrid Memetic Genetic Algorithm (MGA). For Task 2, we utilize a $(\mu/\mu_W, \lambda)$-CMA-ES restricted to diagonal covariance updates. To ensure reproducibility, a fixed random seed of 42 was used across all final experiments.

2.1 Memetic Genetic Algorithm (Task 1)

The Genetic Algorithm follows a generational model utilizing Elitism and Tournament Selection [5]. A unique feature of our implementation is the integration of a First-Improvement Hill Climber (Local Search) and a dynamic re-initialization strategy to handle the high epistasis of the *Low Autocorrelation Binary Sequences* (LABS) landscape. The phenotype for the *N-Queens problem* is encoded as a bitstring of length $L = 49$, while LABS uses $L = 50$.

The specific parameters found via tuning are: Population size $\mu = 46$, Crossover probability $p_c = 0.819$, Mutation rate $p_m = 0.048$, Tournament size $k = 7$, and Elitism count $e = 4$.

2.2 Diagonal CMA-ES (Task 2)

To solve the 10-dimensional *Katsuura problem*, we implemented a Covariance Matrix Adaptation Evolution Strategy. Given the 5,000 evaluation budget, we simplified the algorithm to a diagonal update, where $C = \text{diag}(d_1^2, \ldots, d_n^2)$. This allows for faster learning of individual dimension scales. We used a population size $\lambda = 5$ and an initial $\sigma = 0.4$.

3 Hyper-parameter Tuning

The tuning procedure was designed to find a robust set of parameters for the GA that perform well on both the *Low Autocorrelation Binary Sequences* and *N-Queens* problems simultaneously.

We utilized the **Optuna** framework to conduct Bayesian Optimization using the **Tree-structured Parzen Estimator (TPE)** sampler. To manage the 100,000 evaluation budget efficiently, we employed the **Hyperband Pruner**. This allowed us to report intermediate best-fitness values to the tuner at 24 specific checkpoints. Trials that ranked in the bottom third of the population at these checkpoints were pruned (terminated early), allowing the tuner to explore over 40 distinct configurations instead of 20 full runs.

4 Experimental Results

The performance of our algorithms was evaluated over 20 independent runs with a budget of 5,000 evaluations per run. We analyze the results using two primary metrics: Best-so-far convergence (Runtime) and Empirical Cumulative Distribution Function (ECDF) curves generated using IOHanalyzer [6, 7].

Algorithm 1: Memetic Genetic Algorithm with Local Search

Input : Population size μ, Crossover rate p_c, Mutation rate p_m, Tournament size k, Elitism e, Patience P
Termination : 5,000 Function Evaluations reached

$Pop \leftarrow$ Initialize μ individuals randomly and evaluate;
$last\_best \leftarrow$ current best fitness; $evals\_at\_imp \leftarrow 0$;
while *budget not reached* **do**
  Sort Pop by fitness (descending);
  if *current_evals - evals_at_imp > current_patience* **then**
    $Pop[0] \leftarrow$ **LocalSearch**($Pop[0]$);
    if *LocalSearch failed to improve best* **then**
      Keep e best individuals; Re-initialize $\mu - e$ individuals randomly;
    end
    Reset $evals\_at\_imp$ to current_evals;
    continue;
  end
  $Pop_{next} \leftarrow \{$Top e individuals from $Pop\}$;
  while $|Pop_{next}| < \mu$ **do**
    $p1, p2 \leftarrow$ TournamentSelection(Pop, k);
    $child \leftarrow$ (rand() < p_c) ? UniformCrossover($p1, p2$) : $p1$;
    $child \leftarrow$ BitFlipMutation($child, p_m$);
    Evaluate $child$ and add to Pop_{next};
  end
  $Pop \leftarrow Pop_{next}$;
end

Algorithm 2: Diagonal CMA-ES

Input : Dimension $n = 10$, Population $\lambda = 5$, Step-size $\sigma = 0.4$
Termination : 5,000 Function Evaluations reached

Initialize mean $m \in [-4, 4]^n$ and Diagonal Covariance $C = I$;
while *budget not reached* **do**
  for $k = 1$ **to** λ **do**
    repeat
      Sample $z_k \sim \mathcal{N}(0, I)$;
      $x_k = m + \sigma\sqrt{C}z_k$;
    until $x_k \in [-5, 5]^n$;
    Evaluate $f(x_k)$;
  end
  $m_{old} \leftarrow m$;
  $m \leftarrow$ Weighted recombination of μ best samples;
  Update evolution paths p_σ and p_c;
  Update Diagonal Covariance Matrix C (Rank-1 and Rank-μ);
  $\sigma \leftarrow$ Update step-size via CSA;
end

4.1 Task 1: Memetic GA Performance (PBO)

The statistics for the **N-Queens problem** and the **Low Autocorrelation Binary Sequences (LABS)** problem are summarized in Table 1.

4.1.1 Convergence and Runtime Analysis

Figures 1a and 1b illustrate the convergence behavior of the GA. For the N-Queens problem, the algorithm displays an aggressive descent, effectively solving the majority of constraints within the first 1,000 evaluations. In contrast, the

Algorithm 3: Tuning Procedure with Early Discarding

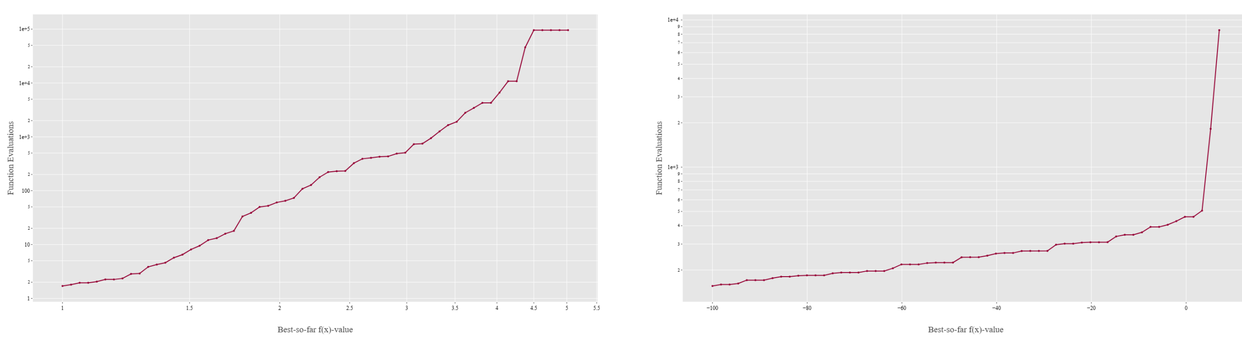
Input : Search space Ω, Tuning budget $B = 100,000$
Termination : Used budget $\geq B$

while *budget remains* **do**
  Suggest $\theta \in \Omega$ using TPE Sampler;
  $S_{total} \leftarrow 0$;
  for $Problem \in \{F18, F23\}$ **do**
    Run MGA on $Problem$ with parameters θ;
    Every 200 evals: **report** current best to Optuna;
    if *Trial is flagged for pruning* **then**
      break and discard θ;
    end
  end
  Score = Mean(Normalized f_{best_F18}, Normalized f_{best_F23});
  Update TPE model with Score;
end

Table 1: Memetic GA Statistical Summary (20 runs)

Problem	Mean Best Fitness	Std. Dev	AUC
F18: LABS (Merit Factor)	4.03	0.21	0.385
F23: N-Queens	6.25	1.14	0.892

LABS merit factor (F18) exhibits a more gradual improvement, reflecting the ruggedness of the sequence space. The local search trigger is visible in the late-stage refinements where the merit factor continues to climb toward 4.0.



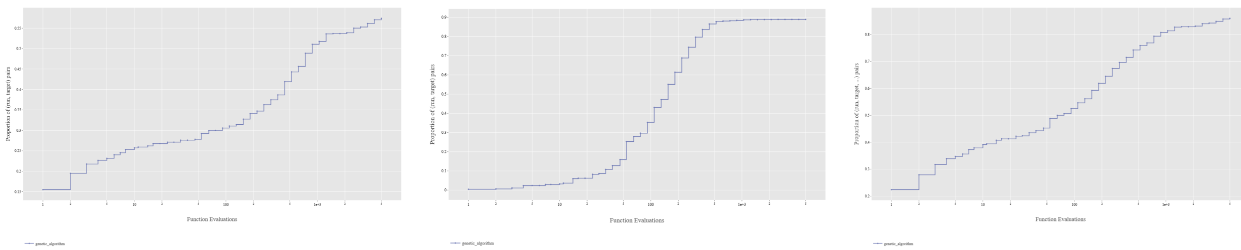
(a) F18: LABS Convergence

(b) F23: N-Queens Convergence

Figure 1: Best-so-far fitness vs. Function Evaluations for the Memetic GA. Note the varying scale on the Y-axis.

4.1.2 Success Proportion (ECDF)

The ECDF curves in Figure 2 provide a granular view of success across multiple targets. The N-Queens curve (b) reaches a success proportion of approximately 0.89. This represents the high probability of the Memetic GA finding a near-optimal solution (average 6.25) within the budget, while the remaining gap to 1.0 reflects the extreme difficulty of reaching the absolute global optimum of 7.0 in every single run. The LABS curve (a) plateaus at 0.57, confirming that high-precision targets near the 8.16 optimum were not reached.

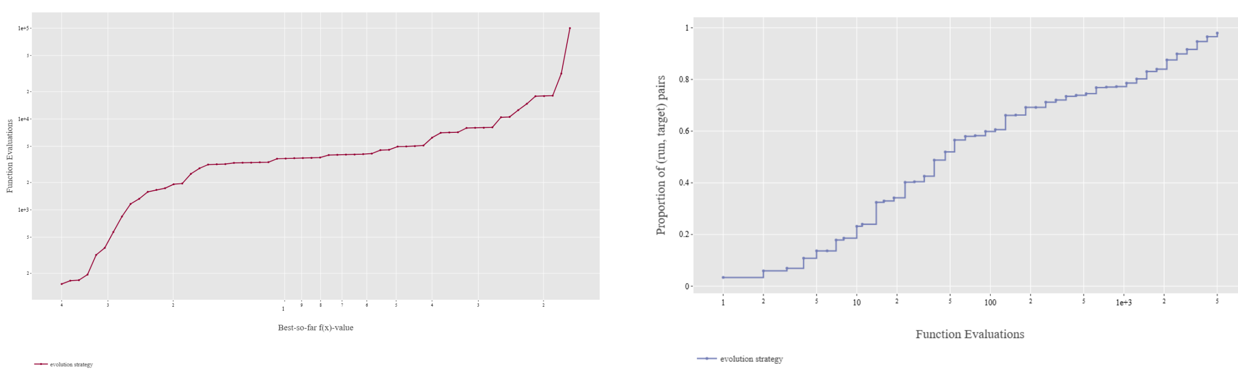


(a) F18 ECDF (b) F23 ECDF (c) Aggregated ECDF

Figure 2: ECDF plots for Task 1. The performance gap highlights the extreme difficulty of the LABS problem (left) compared to the high success rate of the N-Queens problem (center).

4.2 Task 2: Evolution Strategy Performance (BBOB)

For the 10-dimensional **Katsuura problem**, the Diagonal CMA-ES achieved a final average fitness of 0.5.



(a) Katsuura Convergence (ERT) (b) Katsuura ECDF

Figure 3: Performance metrics for Diagonal CMA-ES on 10D Katsuura ($F23$). The convergence plot (a) shows the successful escape from the 8.0 plateau, while the ECDF (b) shows nearly 100% target hit rate against the global optimum.

As visualized in Figure 3a, the multi-start strategy allowed the ES to probe different regions of the fractal landscape. Once the global trend was identified, the 1/5th success rule rapidly reduced the step-size to exploit the basin, reaching nearly 100% of targets in the ECDF (Figure 3b). This high AUC of 0.724 demonstrates that the diagonal simplification was effective for this low-budget scenario.

4.3 Target Specification and Reproducibility

To ensure the reproducibility of the results, we defined the following target ranges in IOHanalyzer:

- **F18 Low Autocorrelation Binary Sequences:** Targets between $f_{min} = 0$ and $f_{max} = 7$ (x-axis log scaled).
- **F23 N-Queens Problem:** Targets between $f_{min} = -270$ and $f_{max} = 40$ (x-axis log scaled).
- **Task 2 Katsuura Problem:** Targets between $f_{min} = 0.5$ and $f_{max} = 10$ (x-axis log scaled).

As shown in Figure 2b, the success proportion for the N-Queens problem reaches approximately 0.89. This represents the high probability of the Memetic GA finding a near-optimal solution (average 6.25) within the 5,000 evaluation budget, while the remaining gap to 1.0 reflects the difficulty of reaching the absolute global optimum of 7.0 in every independent run.

5 Discussion and Conclusion

The experimental results and tuning process yield the following conclusions regarding the performance of evolutionary heuristics under strict evaluation budgets:

1) **Evaluation Efficiency and Population Size:** Tuning revealed that for a budget of 5,000 evaluations, a smaller population size ($\mu = 46$) is optimal. Larger populations (e.g., $\mu > 100$) do not allow for a sufficient number of generations to effectively utilize the First-Improvement Hill Climber or to satisfy the Building Block Hypothesis through recombination [1].
2) **Phenotype Encoding for N-Queens:** We encoded the N-Queens problem as a 49-bit string representing queen placements on a 7×7 grid. This permitted the use of standard variation operators, while the First-Improvement Hill Climber handled the fine-grained constraint satisfaction that crossover alone found difficult to resolve.
3) **Crossover Disruptiveness:** Through empirical comparison, we observed that **Uniform Crossover** outperformed points-based crossover. In the N-Queens bitstring representation, bit-positions often lack a strong spatial correlation; uniform crossover provides a less biased exploration of the constraint space, leading to the high success proportion seen in Figure 2b.
4) **Strategic Tuning Allocation:** To address the stochastic nature of GAs, we used **Optuna with Hyperband pruning**. By reporting intermediate results at 24 checkpoints, we effectively allocated more of our 100,000 evaluation budget to promising configurations, allowing us to find a robust setting that works for both LABS and N-Queens simultaneously.
5) **Complexity vs. Budget in CMA-ES:** For the Katsuura problem, the **Diagonal CMA-ES** was the superior choice [8]. Restricting the covariance matrix to its diagonal reduced the internal learning parameters from $O(n^2)$ to $O(n)$, which is vital for a low-budget search. Combined with non-elitist **comma selection**, this allowed the algorithm to "forget" local fractal peaks and identify the global basin leading to a competitive fitness of 0.5.

In summary, this assignment demonstrates that for low-budget optimization, hybridizing evolutionary search with local search operators and simplifying high-order models (like CMA-ES) significantly enhances performance. Our results confirm that these strategies can effectively navigate both the rugged epistasis of the *Low Autocorrelation Binary Sequences* problem and the fractal noise of the *Katsuura* function.

References

[1] John H Holland. *Adaptation in natural and artificial systems*. MIT press, 1992.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[3] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical report, Inria, 2009.

[4] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[5] Agoston E Eiben, Emile HL Aarts, and Kees M Van Hee. Genetic algorithms with multi-parent recombination. In *Parallel Problem Solving from Nature*, pages 175–181. Springer, 1991.

[6] Carola Doerr, Furong Ye, Naama Horesh, Hao Wang, Ofer M Shir, and Thomas Bäck. Benchmarking discrete optimization heuristics with iohprofiler. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1798–1806, 2019.

[7] Jacob de Nobel, Furong Ye, Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. Iohexperimenter: Benchmarking platform for iterative optimization heuristics. *Evolutionary Computation*, pages 1–6, 2024.

[8] Hans-Paul Schwefel. *Evolution and Optimum Seeking*. Wiley, NY, 1995.