NTNU
Kunnskap for en bedre verden

Department of Computer Science

IT3708 – Bio-Inspired Artificial Intelligence

Pseudo-Boolean Optimization with Bio-Inspired Algorithms

Authors:

Magnus Grini
Yasin Marouga

26.04.2026

Table of Contents

List of Figures

List of Tables

1 Introduction

This report explores the domain of pseudo-Boolean optimization using bio-inspired algorithms. The focus is on feature selection problems as well as other arbitrary benchmark functions.

For feature selection, the objective is to maximize the predictive accuracy of underlying models while applying regularization techniques to encourage solutions that use fewer features without significantly compromising performance.

To further analyze the behavior of the algorithms we also include additional functions where minimizing the number of activated bits is not explicitly encouraged. One such function is the Triangle function [1].

2 Datasets and Exploratory Data Analyses

The training datasets provided are a set of ablation studies of four different machine learning models. The studies evaluate a model's performance by testing each feature combination of each study's underlying model in multiple trials. In contrast to the three provided datasets, the Triangle dataset is the only synthetic dataset. The procedure for generating the values of the dataset can be found in Xavier's paper [1, p. 2].

Table 1: Summary of datasets. Mean and standard deviation are computed over all accuracies. The *Triangle* datasets are deterministic.

Dataset	Features	Search Space	Trials	Mean Acc.	Std. Dev.
Breast [2]	9	511	32	0.9521	0.0210
Letter [3]	16	65535	32	0.7639	0.1315
Credit [4]	15	32767	32	0.8009	0.0731
Triangle [1]	16	65536	—	—	—
Zoo [5]	16	65535	32	—	—
Hepatitis [6]	19	524287	32	—	—
Triangle-test [1]	31	2^{31}	—	—	—

To further explore the datasets, the fitness landscapes were visualized in several ways. Figure 7 summarizes the main landscape visualization for each training dataset discussed in the following subsections. In the tables, Pareto-optimal solutions are defined with respect to maximizing $h_a(T(x))$ while minimizing the number of selected features ($h_p(x)$).

2.1 Triangle Dataset EDA

Figure 7a shows the hinged bitstring map for the Triangle dataset.

2.2 Breast Dataset EDA

Figure 7b shows the Breast dataset's penalized fitness as a function of selected feature count. Table 9 lists the Pareto-optimal and locally optimal feature subsets identified for the Breast dataset. The global optimum selects three features and achieves a penalized fitness of 0.9496. An additional visualization of the local-optima structure can be found in Figure 5.

2.3 Letter Dataset EDA

Figure 7c shows the Letter dataset's penalized fitness as a function of selected feature count. The selected Pareto-optimal and locally optimal subsets as well as the local-optima network are summarized in Table 10 and Figure 6. The global optimum selects eleven features.

2.4 Credit Dataset EDA

Figure 7d shows the Credit dataset's penalized fitness as a function of selected feature count. The Credit dataset contains 347 local optima. The global optimum is $x^{\text{opt}} = \texttt{000100001100011}$.

A full list of selected feature subsets for the Credit dataset is given in Table 11, and an additional visualization of its local-optima network is shown in Figure 8.

3 Methodology

3.1 Representation and Fitness Landscapes

Each chromosome is represented as a binary vector, where each bit indicates whether the corresponding feature is active. The implementation stores each chromosome as a packed integer mask, which also acts as the lookup-table index. Empty chromosomes are not allowed for feature-selection landscapes and are repaired by activating one random feature when sampled, while they are allowed by default for the synthetic triangle landscape.

The feature-selection datasets contain multiple accuracy measurements for each non-empty feature subset [7], [8]. For single-objective feature-selection runs, the penalized fitness is defined as [7]

$$h(x) = a(x) - \epsilon \frac{|x|_1}{n}, \tag{1}$$

where $a(x)$ is the mean accuracy of chromosome x across the repeated measurements, $|x|_1$ denotes the number of active bits in the chromosome, n is the total number of features in the dataset and ϵ is the penalty coefficient.

The synthetic landscape uses the triangle function [1], whose fitness depends only on the Hamming weight and is defined as

$$f_{\text{tri}}(x) = m \cdot \min(|x|_1 \bmod 2s,\ 2s - (|x|_1 \bmod 2s)), \tag{2}$$

where m is a scaling factor and s defines the half-period of the triangle function.

3.2 Landscape Analysis

For each stored landscape, every valid point is analyzed using its one-bit Hamming neighborhood [9]. A point is marked as a local optimum if none of its valid one-bit neighbors has strictly higher fitness [9]. Strict local optima additionally have no equal-fitness neighbors. Pareto-optimal points are identified with respect to maximizing fitness while minimizing the number of selected features [10].

Local optima networks are constructed from lookup-table landscapes [7]. Each point is assigned to a basin by repeatedly moving to the best improving one-bit neighbor until a local optimum is reached, using a deterministic tie break when several improving neighbors have the same best fitness. Edges between basins are counted from neighboring points assigned to different basins. These summaries are exported to CSV for visualization.

3.3 Genetic Algorithm

The single-objective algorithm uses a simple genetic algorithm. The initial population is generated by independent random bit sampling, followed by non-empty repair when required. Parent selection uses probabilistic tournament selection where the best tournament member is selected with probability `selection-rate`, otherwise the runner-up is selected [11]. Offspring are created by uniform crossover, where each bit is exchanged with probability `crossover-toss-rate`. Mutation is standard bit-flip mutation, where each bit is independently flipped with probability `mutation-rate` [11].

The implemented GA run configuration uses elitism with generalized crowding [12]. In the generalized crowding, each selected parent is paired with the most genetically similar child according to Hamming distance. For each parent–child pair, the child survives with probability

$$p(c) = \begin{cases} \dfrac{f(c)}{f(c) + f(p)\varphi}, & f(c) > f(p), \\ \dfrac{\varphi f(c)}{\varphi f(c) + f(p)}, & f(c) < f(p), \\ 0.5, & f(c) = f(p), \end{cases} \tag{3}$$

where φ is the crowding scaling factor, $f(c)$ is the child fitness, and $f(p)$ is the parent fitness.

3.4 NSGA-II

The multi-objective algorithm uses NSGA-II [13]. For feature selection, the objectives are high unpenalized accuracy and low feature count. For comparison with the other algorithms, NSGA-II results are summarized using penalized fitness $h(x)$. For triangle landscapes, NSGA-II optimizes unpenalized fitness together with an implementation-specific niching objective inspired by diversity preservation in multi-objective optimization [10]. The niching objective rewards underrepresented Hamming-weight shells and is defined as

$$s(|x|_1) = \frac{1}{\sqrt{\sum_y \max(0, 1 - ||x|_1 - |y|_1|/r)}}, \tag{4}$$

where $|x|_1$ is the number of selected bits in the current solution, y ranges over the current population, $|y|_1$ is the number of selected bits in chromosome y, and r is the count sharing radius. This encourages the survivor population to cover multiple Hamming-weight shells instead of collapsing to one weight class.

The NSGA-II implementation uses standard crowded tournament selection, non-dominated sorting, and crowding distance [13], together with the same uniform crossover and bit-flip mutation operators as the genetic algorithm. Survivor selection is performed from the combined parent-offspring population. Duplicate genomes are removed when possible, and for triangle runs a small reserve mechanism helps preserve missing Hamming-weight niches.

3.5 Binary Particle Swarm Optimization

The swarm algorithm uses binary particle swarm optimization [14]. Each particle stores a binary position, a real-valued velocity for each bit, and its personal best. The velocities are updated using inertia and attraction toward good previous positions, and the update rule is defined by

$$v_{i,j}^{t+1} = \omega_t v_{i,j}^t + A_{i,j}^t, \tag{5}$$

where $v_{i,j}^t$ is the velocity of bit j in particle i at time t, ω_t is the linearly decreasing inertia weight, and $A_{i,j}^t$ is the attraction toward good previously found positions. The attraction term is based on the personal bests of particles in a ring neighborhood. Velocities are clamped to avoid extreme probabilities. Since the particle position must be binary and not real-valued, each velocity is converted to a probability using the logistic function [14].

$$P(x_{i,j}^{t+1} = 1) = \frac{1}{1 + e^{-v_{i,j}^{t+1}}}. \tag{6}$$

The new bit value is then sampled randomly according to this probability, producing either 0 or 1. Personal bests are updated whenever a particle finds a strictly better solution.

3.6 Parameter Control and Evaluation

All algorithms use random initialization. In the reported experiments, population sizes, evaluation budgets, and bit-flip mutation rates are scaled by the problem dimension n, while selection, crossover, crowding, and BPSO coefficients are fixed across landscapes. For the GA, the mutation rate and crowding scaling factor are adapted by population entropy using the following implementation-specific formula, based on online parameter control and diversity-adaptive generalized crowding [15], [16].

$$q(H) = q_{\min} + (q_{\max} - q_{\min}) \left(1 - \frac{H}{\max(H_0, H_{\text{ref}})}\right), \tag{7}$$

where H is the current population entropy, H_0 is the initial entropy, H_{ref} is a fixed normalization baseline, and $q_{\min}$ and $q_{\max}$ define the allowed range of the parameter. As entropy decreases, both the mutation rate and crowding scaling factor increase.

Each logged run records best fitness, mean fitness, entropy, mean Hamming distance to the best individual, and the final population. Experiments are repeated over multiple random seeds, and the comparative scripts summarize mean fitness, standard deviation, global/local optimum hit rates, Hamming distance to the optimum, and convergence behavior.

4 Experimental Setup

4.1 Training Set Parameter Setup

We used the same parameter policy across the training sets, scaling population size and mutation strength with the problem dimension n. For all feature-selection datasets, we used $\epsilon = 0.05$, 10 seeds, 200 generations, and no archive-based reporting in the final analysis.

Table 2: Training-set population sizes, mutation rates, and evaluation budgets.

Dataset	GA pop.	NSGA-II pop.	BPSO pop.	NSGA-II mut.	Max evals
Breast	72	72	90	0.1111	7200
Credit	120	120	150	0.0667	12000
Letter	128	128	160	0.0625	12800
Triangle	128	128	160	0.0625	12800

Shared crossover, mutation, and selection settings were applied consistently across training runs. Full hyperparameter specifications are provided in Table 12.

For GA, the mutation and crowding parameters were entropy-scaled with reference entropy 0.8. The mutation ranges were $[0.0556, 0.4444]$ for Breast, $[0.0333, 0.2667]$ for Credit, and $[0.03125, 0.25]$ for Letter and Triangle. The crowding range was $[0.5, 3.0]$ for all GA runs. For Triangle, NSGA-II additionally used a count-sharing radius of 3.

4.2 Test Set Hyperparameter Optimization

Because the hidden test instances fall into two different problem families, hyperparameter optimization was carried out separately for the feature-selection algorithms (GA and BPSO) and for NSGA-II. For the hidden feature-selection tasks, tuning was performed on the raw Breast, Credit, and Letter landscapes, since the final hidden instances also use the unpenalized objective ($\epsilon = 0$). A single shared parameterization was selected for each algorithm based on aggregate performance across these training landscapes and then transferred unchanged to both hidden feature-selection test sets. For these algorithms, the only size-dependent parameter was population size (or swarm size), which was scaled linearly with the number of features n.

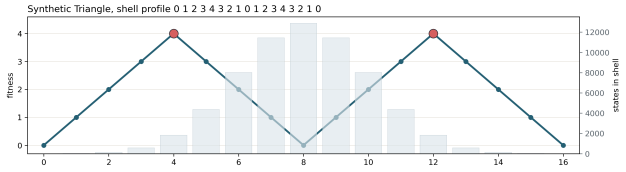
NSGA-II was tuned separately for the synthetic multimodal triangle task. Since its purpose was to maintain diverse local-optimum niches rather than optimize the feature-selection objective directly, it was not tuned on the feature-selection landscapes. Instead, it was tuned on the provided training triangle instance, with selection based on consistent recovery of the local-optimum shells across seeds.

Final shared configurations for the hidden test sets were selected based on aggregate performance across training landscapes. The full configuration details are provided in Table 13.

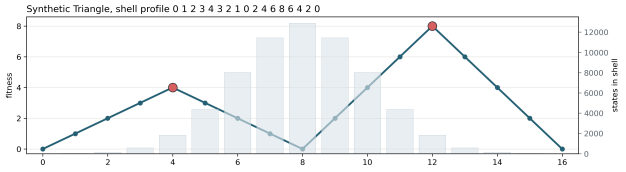
5 Results and Findings

5.1 Algorithm Behavior

This subsection examines how the algorithms move through the triangle landscapes, with emphasis on niche occupancy, diversity over time, and the effect of changing the landscape structure. Because the triangle fitness depends only on Hamming weight, shell occupancy is a useful representation of niche occupancy in these visualizations.



(a) Training triangle landscape.



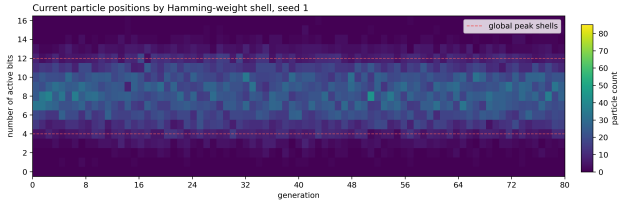
(b) Asymmetric triangle landscape.

Figure 1: Triangle landscapes used in the behavior analysis. The training triangle has equal peaks at shells 4 and 12, while the asymmetric triangle makes shell 12 dominant.

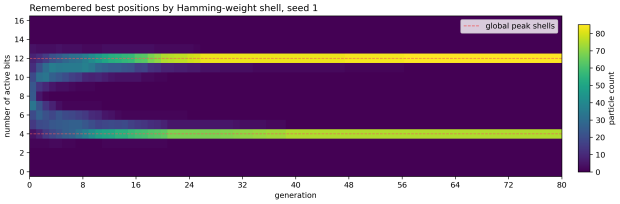
5.1.1 BPSO on the Training Triangle

The training triangle has two equally good peak shells at weights 4 and 12 (Figure 1a). On this landscape, BPSO shows a clear separation between exploration and memory. In Figure 2a, the current particle positions remain broadly distributed around the middle shells instead of collapsing onto either peak. This matches the implementation since particles sample new bitstrings stochastically from their velocities and are influenced by a fully informed local neighborhood. In contrast, Figure 2b shows that the remembered best positions quickly accumulate on both peak shells.

Overall, BPSO continues to explore with its current positions while preserving both good niches through personal-best memory.



(a) Current particle positions over generation.



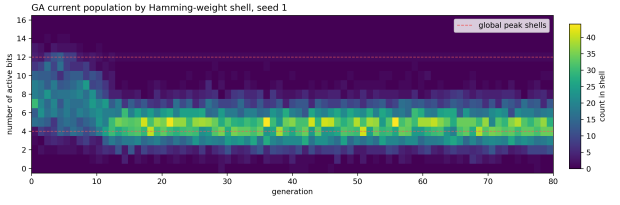
(b) Remembered best positions over generation.

Figure 2: BPSO on the training triangle. Current positions stay dispersed, while remembered personal-best positions concentrate on the two peak shells.

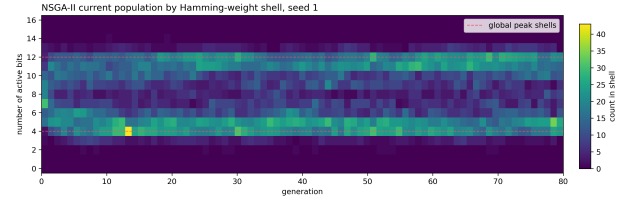
5.1.2 Comparison with GA and NSGA-II

The GA and NSGA-II results show different diversity dynamics on the same training landscape. The GA population in Figure 3a gradually concentrates around the lower peak region and largely abandons the upper one. Its mean bit entropy in Figure 3c drops quickly early in the run, which is to be expected from a single-objective GA using no explicit diversity-preserving mechanisms.

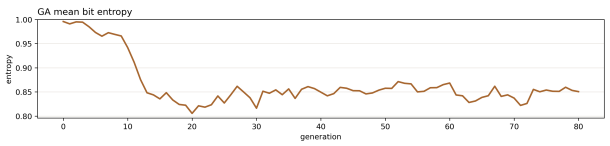
NSGA-II behaves differently. On the triangle landscape, it optimizes for both fitness and genotype novelty. This is reflected in Figures 3b and 3d, which show that it keeps population mass near both peak shells and maintains entropy at a consistently high level. Compared with GA, NSGA-II preserves diversity much more effectively. Compared with BPSO, it preserves both niches directly in the current population rather than mainly in remembered best positions.



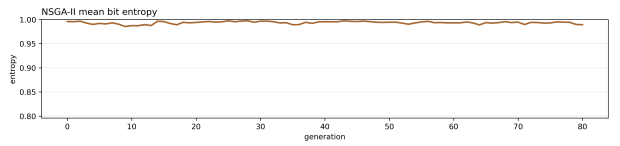
(a) GA current population over generation.



(b) NSGA-II current population over generation.



(c) GA mean bit entropy over generation.



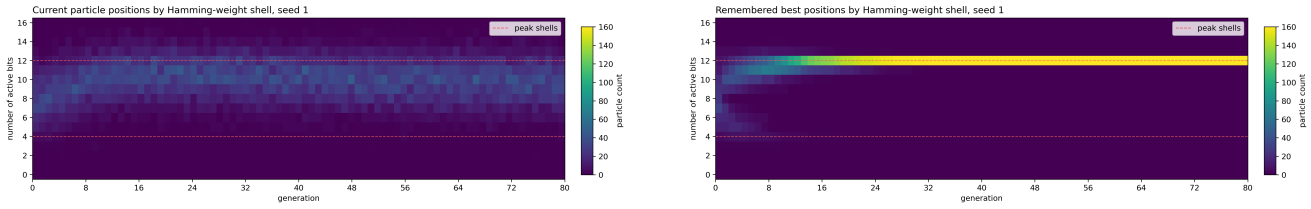
(d) NSGA-II mean bit entropy over generation.

Figure 3: GA and NSGA-II behavior on the symmetrical training triangle. The GA converges toward one region and loses diversity early, while NSGA-II keeps both peak regions populated and maintains high diversity.

5.1.3 Effect of Changing the Landscape

The asymmetric triangle shows that BPSO does not preserve both peaks by default. When shell 12 becomes strictly better than shell 4 (Figure 1b), the remembered-best distribution collapses almost entirely onto the dominant niche in Figure 4b, and the current swarm also shifts upward in Figure 4a.

This comparison shows how the same algorithm reacts to a different landscape structure. On the symmetrical training triangle, BPSO can retain both niches because they are equally good. Once the symmetry is broken, the personal-best memory and social attraction mechanisms bias the swarm toward the globally stronger peak.



(a) Current particle positions over generation. **(b)** Remembered best positions over generation.

Figure 4: BPSO on the asymmetric triangle. Both current and remembered positions shift toward the dominant upper peak.

5.1.4 Summary

Overall, the visualizations show three consistent patterns. GA converges quickly but tends to commit to one basin, NSGA-II maintains diversity across basins by design, and BPSO balances continued exploration in its current positions with exploitation through personal-best memory.

5.2 Comparative Performance

This section compares the performance of the three algorithms across the training datasets (see Table 1). Results are aggregated over 10 independent runs and reported as mean ± standard deviation where applicable. The experimental setup is detailed in Section 4.1.

Table 3: Breast Dataset: Algorithm performance over 10 seeds.

Algorithm	Best Subset (x)	Best $h(x)$	x^{opt} Hit Rate	Evals to x^{opt}	Local Optima per Run
BPSO	`100001010`	0.9496 ± 0.0000	10/10	360.0 ± 263.9	3.7 ± 1.1
GA	`100001010`	0.9496 ± 0.0000	10/10	348.9 ± 214.1	3.3 ± 1.0
NSGA-II	`100001010`	0.9496 ± 0.0000	10/10	223.2 ± 118.1	11.0 ± 0.0

Table 4: Letter Dataset: Algorithm performance over 10 seeds.

Algorithm	Best Subset (x)	Best $h(x)$	x^{opt} Hit Rate	Evals to x^{opt}	Local Optima per Run
BPSO	`0000011111111111`	0.9220 ± 0.0000	10/10	1664.0 ± 595.2	0.8 ± 0.4
GA	`0000011111111111`	0.9220 ± 0.0000	10/10	1463.6 ± 510.9	1.0 ± 0.0
NSGA-II	`0000011111111111`	0.9220 ± 0.0000	10/10	832.0 ± 263.9	1.0 ± 0.0

Table 5: Credit Dataset: Algorithm performance over 10 seeds.

Algorithm	Best Subset (x)	Best $h(x)$	x^{opt} Hit Rate	Evals to x^{opt}	Local Optima per Run
BPSO	`000100001100011`	0.8705 ± 0.0000	10/10	2595.0 ± 765.0	4.4 ± 1.7
GA	`000100001100011`	0.8702 ± 0.0008	9/10	2637.3 ± 2218.1	4.8 ± 2.4
NSGA-II	`000100001100011`	0.8705 ± 0.0000	10/10	1512.0 ± 951.3	32.4 ± 1.7

For performance comparisons on the Triangle dataset, see Section 5.1.

5.3 Test Instance Performance

Table 6: Triangle-test Dataset: NSGA-II summary over 10 runs (500 generations). Values are mean ± standard deviation.

Best Solution (x)	Best $h(x)$	Evals to x^{opt}	Local Optima (x^*)	Dominant Shell Share	Shell Discovery
31	6.0 ± 0.0	50435 ± 39746	586 ± 20	$34.8\% \pm 1.5\%$	$100.0\% \pm 0.0\%$

Table 7: Zoo Dataset: BPSO summary statistics over 10 runs (150 generations). Values are reported as mean ± standard deviation.

Best Solution (x)	Best $h(x)$	Evals to x^{opt}	Entropy	Diversity
21705	0.9792 ± 0.0000	2888.0 ± 1743.3	0.9420 ± 0.0082	6.6686 ± 0.3496

Table 8: Hepatitis Dataset: GA summary over 10 runs (2500 generations). Values are mean ± standard deviation.

Best Solution (x)	Best $h(x)$	Evals to x^{opt}	Entropy	Diversity
184322	0.9375 ± 0.0000	76082.2 ± 67828.5	0.8590 ± 0.0127	8.2046 ± 0.5610

6 Discussion

6.1 Differences Between Training and Test Landscapes

The results indicate that strong performance on the training landscapes did not automatically transfer to the hidden test instances. Although the training datasets were sufficient for tuning general search behavior, the hidden tests differed in ways that altered which mechanisms were most important for success. This is evident when comparing the training-set results in Tables 3–5 with the hidden-test summaries in Tables 6–8. In particular, the synthetic triangle test was not merely a larger version of the training triangle, but a structurally different multimodal landscape with an asymmetric global optimum.

This difference was also reflected in the role of search budget. Some configurations that initially appeared weak on the hidden test sets were not necessarily poorly tuned, but instead limited by insufficient generations or evaluations. This was especially clear for the hepatitis dataset (Table 8), where increasing the GA budget substantially improved the optimum hit rate. These results suggest that transferring a tuned configuration from the training set required not only reusing operator parameters, but also reconsidering whether the computational budget remained appropriate for the hidden instance.

6.2 Solving the Triangle Test Dataset

The synthetic triangle test instance required a more iterative refinement of the methodology than the feature-selection test sets. During the training stage, NSGA-II was selected as the main algorithm for highly multimodal landscapes because its multiobjective selection mechanism is well suited to preserving multiple niches. This tendency is already visible in the training-triangle behavior shown in Figure 3, where NSGA-II maintains solutions across multiple peak regions while the GA collapses more strongly.

However, the initial NSGA-II formulation did not reliably locate the global optimum on the hidden triangle test instance. The main issue was that genotype-level distance in Hamming space did not necessarily correspond to diversity across shell weights. The algorithm could therefore maintain many different bitstrings while still concentrating most of its search effort within the same shell. In addition, standard bit-flip mutation was not well suited to moving consistently between distant shell weights.

To address this, the second objective was replaced with a sharing-based mechanism over Hamming weight, so that the second objective explicitly rewarded underrepresented shell regions rather than merely genotypic difference. A Hamming-weight mutation mechanism was also introduced to improve movement between shell regions. The effect of these changes is reflected in Table 6, where NSGA-II consistently recovered all shell optima and reached the global optimum on the synthetic test landscape.

6.3 Reliability, Budget Sensitivity, and Algorithm Behavior

The experiments also show that the three algorithms responded differently to multimodality. On the training feature-selection datasets (Tables 3–5), all three methods performed competitively, but their behavior diverged more clearly on structured multimodal landscapes. The GA tended to converge more aggressively toward a single basin, NSGA-II was better able to preserve multiple niches, and BPSO behaved differently again because its personal-best memory allowed promising solutions to be retained even when the current swarm moved elsewhere.

A further distinction concerned eventual success versus convergence speed. A high final hit rate did not necessarily imply stable time-to-solution. This was visible on the triangle test (Table 6), where NSGA-II consistently reached the optimum but with substantial variation in the evaluations required to first do so. A similar effect appeared for the GA on hepatitis (Table 8): even when the search budget was increased enough to make optimum discovery reliable, the number of evaluations required to reach the optimum still varied considerably across seeds. This suggests that the GA became reliable in eventual outcome, but remained sensitive to stochastic effects in convergence speed.

7 Conclusion

This project investigated pseudo-Boolean optimization using GA, NSGA-II, and BPSO on both feature-selection and synthetic multimodal landscapes. The results show that no single algorithm was uniformly best across all problem types. While all three methods were competitive on the training feature-selection tasks, their differences became clearer on more multimodal landscapes.

NSGA-II proved most effective on the synthetic triangle test after its objectives and operators were aligned with the shell structure of the landscape. BPSO benefited from its personal-best memory, while the GA remained competitive but was more sensitive to multimodality and search budget. Overall, the project shows that performance depends not only on algorithm choice, but also on how well the operators and budget match the structure of the landscape.

Appendix

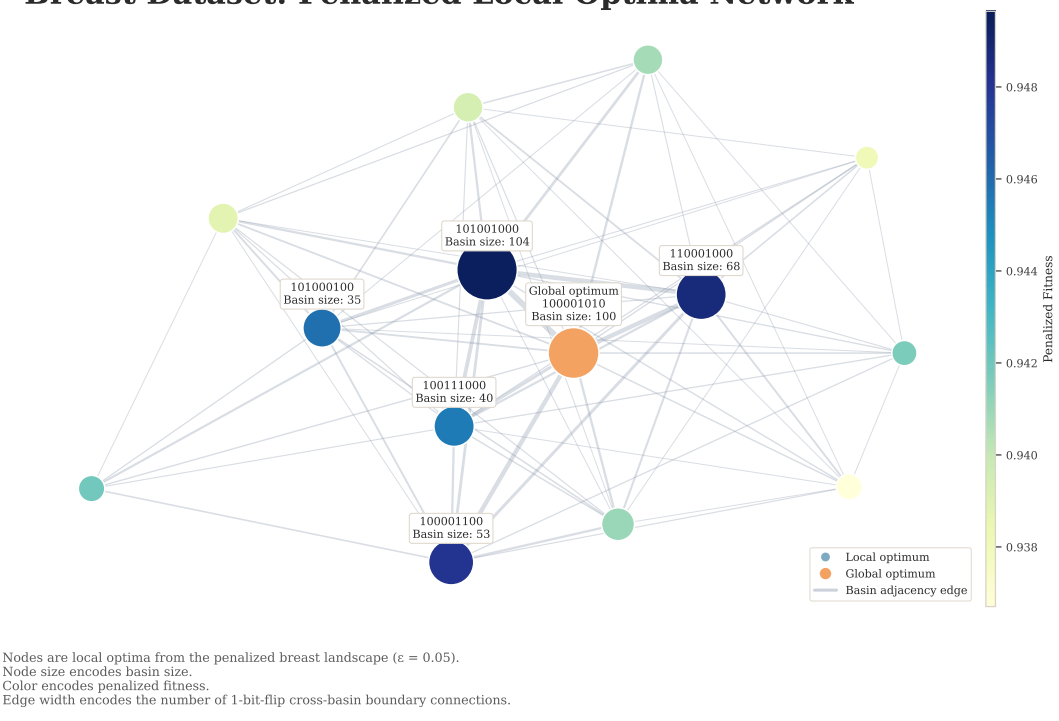
A Breast dataset feature subsets

Table 9: Breast dataset: Selected feature subsets, including Pareto-optimal solutions under $h_a(T(x^*))$ and locally optimal solutions under the penalized objective $h(x^*)$. Subsets satisfying both criteria are marked as Both. The global optimum under $h(x^*)$ is shown in bold.

Subset (x^*)	Features	Source	$h_a(T(x^*))$	$h(x^*)$
010000000	1	Pareto	0.9323	–
010001000	2	Pareto	0.9556	–
011000000	2	Local	–	0.9393
001000010	2	Local	–	0.9407
110010000	3	Local	–	0.9381
100100100	3	Local	–	0.9410
101000100	3	Local	–	0.9458
100001100	3	Local	–	0.9481
101001000	3	Local	–	0.9495
100001010	**3**	Both	0.9663	**0.9496**
110001010	4	Pareto	0.9683	–
101100010	4	Local	–	0.9388
100011001	4	Local	–	0.9416
100111000	4	Local	–	0.9455
100011110	5	Pareto	0.9697	–
101101100	5	Both	0.9697	0.9419
010011011	5	Local	–	0.9367
111001111	7	Pareto	0.9701	–

B Breast dataset local optima network

Figure 5: Local Optima Network of the Breast dataset



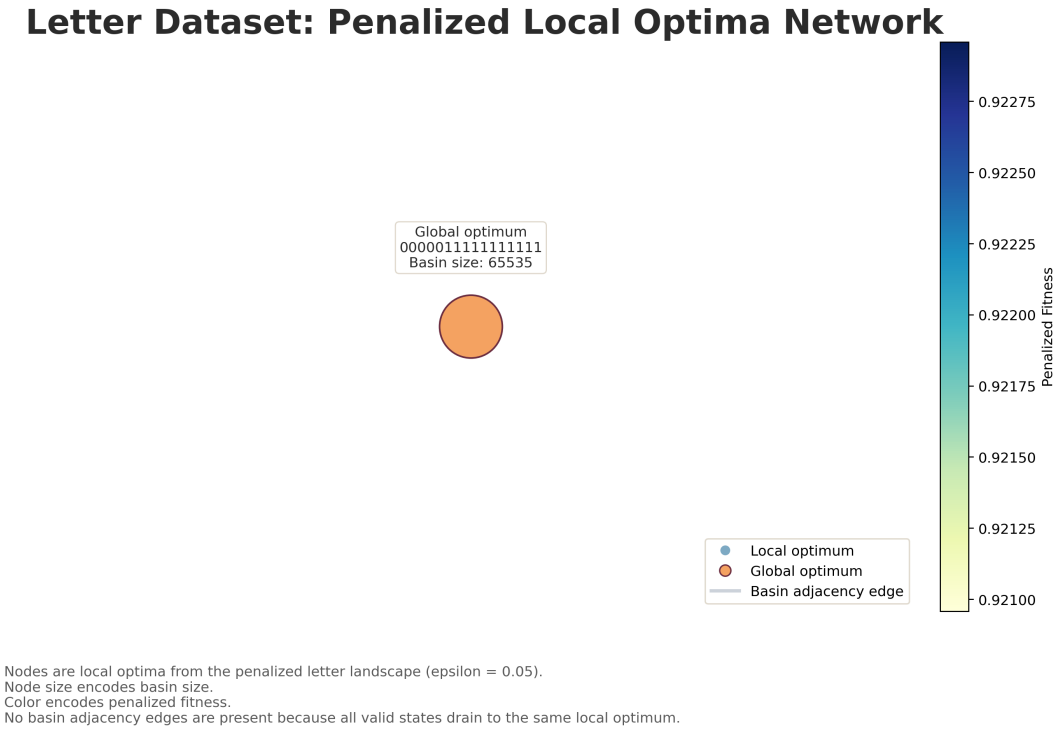
C Letter dataset feature subsets

Table 10: Letter dataset: Selected feature subsets, including Pareto-optimal solutions under $h_a(T(x^*))$ and locally optimal solutions under the penalized objective $h(x^*)$. The subset satisfying both criteria is marked as Both. The global optimum under $h(x^{\text{opt}})$ is shown in bold.

Subset (x^*)	Features	Source	$h_a(T(x^*))$	$h(x^*)$
0000000000100000	1	Pareto	0.1197	–
0000000000101000	2	Pareto	0.2967	–
0000000010001010	3	Pareto	0.5187	–
0000000110001010	4	Pareto	0.7047	–
0000000110101010	5	Pareto	0.8063	–
0000000110111010	6	Pareto	0.8773	–
0000000111111010	7	Pareto	0.9191	–
0000000111111011	8	Pareto	0.9389	–
0000000111111111	9	Pareto	0.9473	–
0000010111111111	10	Pareto	0.9525	–
0000011111111111	**11**	Both	0.9563	**0.9220**

D Letter dataset local optima network

Figure 6: Local Optima Network of the Letter dataset

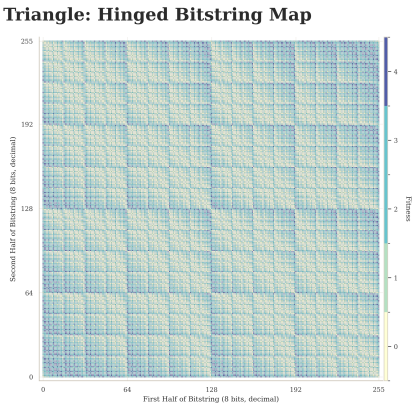


E Credit dataset feature subsets

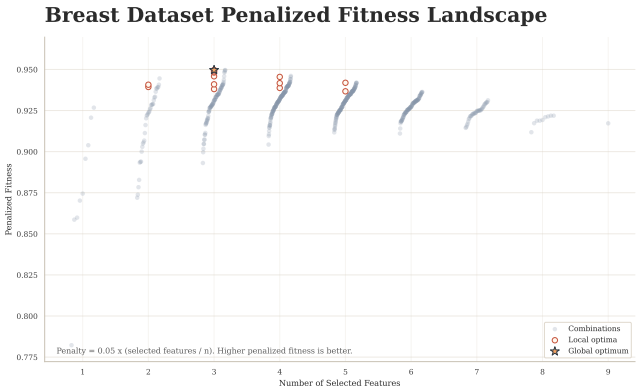
Table 11: Credit dataset: Selected feature subsets, including Pareto-optimal solutions under $h_a(T(x^*))$ and locally optimal solutions under the penalized objective $h(x^*)$.

Subset (x^*)	Features	Source	$h_a(T(x^*))$	$h(x^*)$
0000000001000000	1	Pareto	0.8627	–
0000000101000000	2	Pareto	0.8712	–
0010000001100000	3	Pareto	0.8729	–
0000000001010110	4	Pareto	0.8782	–
0001000001100011	5	Pareto	0.8871	–
0011100011110111	9	Pareto	0.8913	–

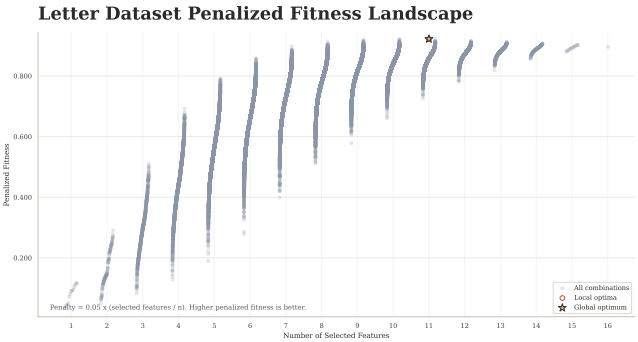
F Fitness Landscape Visualizations



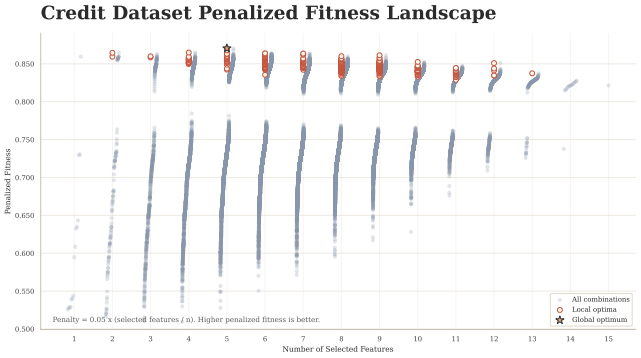
(a) Triangle dataset.



(b) Breast dataset.



(c) Letter dataset.

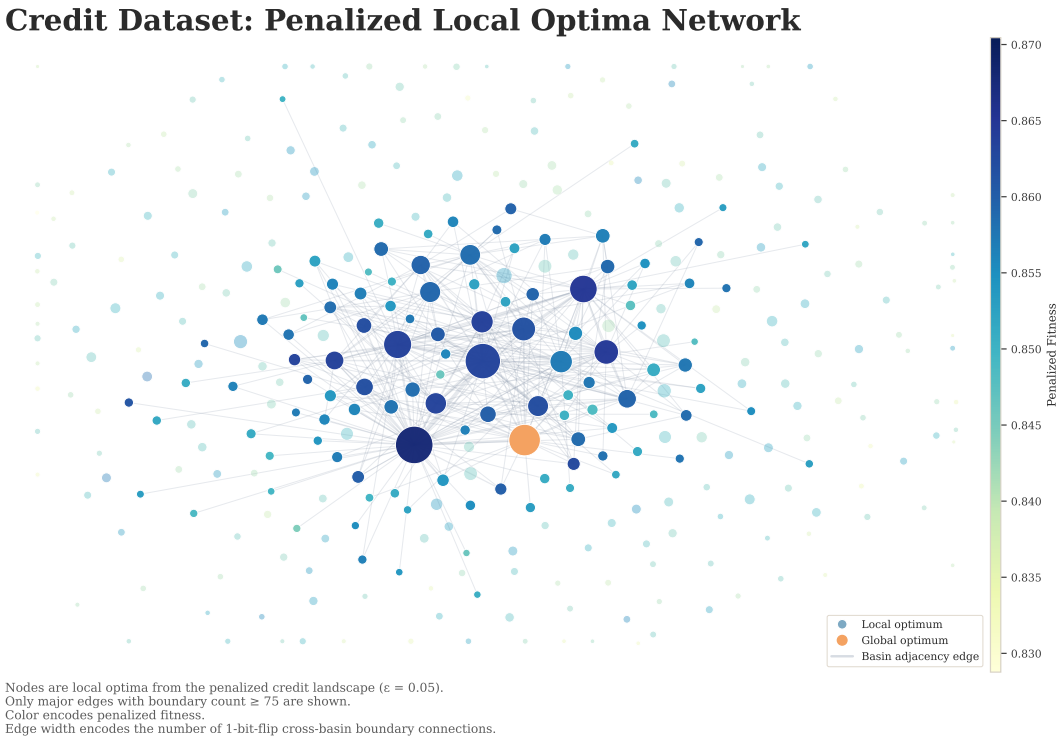


(d) Credit dataset.

Figure 7: Fitness landscape visualizations for the four training datasets discussed in Sections 2.1–2.4. The Triangle plot uses a hinged bitstring map, while the remaining plots show penalized fitness against the number of selected features.

G Credit dataset local optima network

Figure 8: Local Optima Network of the Credit dataset



H Training Hyperparameter Details

Table 12: Shared hyperparameter settings across training runs.

Algorithm	Shared settings
GA	crossover = 0.9, toss = 0.5, tournament = 3, selection = 0.8, generalized crowding
NSGA-II	crossover = 0.9, toss = 0.5, tournament = 2, selection = 1.0
BPSO	inertia $0.9 \rightarrow 0.4$, $c_1 = c_2 = 2.05$, velocity limit = 4.0, neighborhood radius = 2

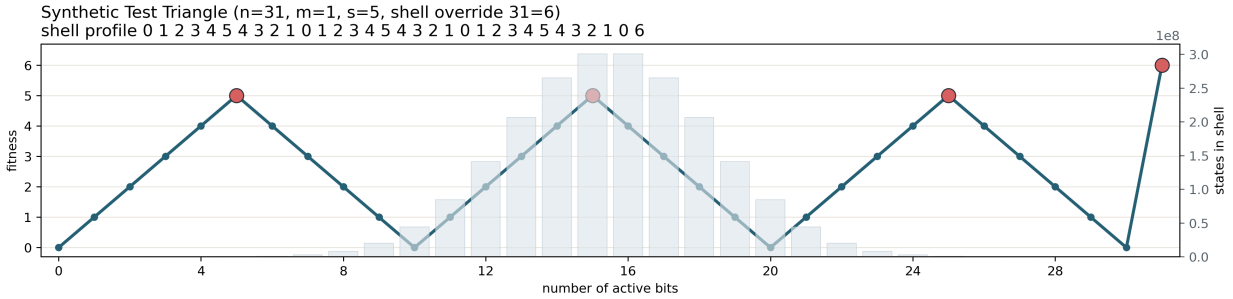
I Final Test Configurations

Table 13: Shared configurations selected for the hidden feature-selection test sets and the hidden synthetic triangle test instance.

Algorithm	Final shared test-set configuration
GA	population size = $6n$, crossover rate = 0.872, crossover toss rate = 0.312, elitism = 1, tournament size = 2, selection rate = 0.757, generational survivor selection, and entropy-scaled mutation with minimum ratio = 0.575, maximum ratio = 2.462, and entropy reference = 0.706.
BPSO	swarm size = $6n$, inertia weight linearly decayed from 0.772 to 0.200, cognitive coefficient $c_1 = 2.240$, social coefficient $c_2 = 2.185$, velocity limit = 2.370, and a global-best social topology.
NSGA-II	population size = 650, crossover rate = 0.882, crossover toss rate = 0.101, tournament size = 3, selection rate = 0.667, fixed mutation rate = 0.285, selected-count sharing radius = 1, and adaptive selected-count mutation enabled with base ratio = 0.300, maximum ratio = 1.000, and low-diversity threshold = 0.300. This configuration was tuned on the training triangle instance and transferred unchanged to the hidden synthetic triangle test set.

J Test Triangle Landscape Visualization

Figure 9: Test triangle landscape



Bibliography

[1] X. F. C. Sánchez-Díaz and O. J. Mengshoel, 'Estimating the number of local optima in multimodal pseudo-boolean functions: Validation via landscapes of triangles', in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '24 Companion, Melbourne, VIC, Australia: Association for Computing Machinery, 2024, pp. 211–214, ISBN: 9798400704956. DOI: 10.1145/3638530.3654156. [Online]. Available: https://doi.org/10.1145/3638530.3654156.

[2] W. Wolberg, *Breast Cancer Wisconsin (Original)*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5H 1990.

[3] D. Slate, *Letter Recognition*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5ZP40, 1991.

[4] J. R. Quinlan, *Credit Approval*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5FS30, 1987.

[5] R. Forsyth, *Zoo*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5R59V, 1990.

[6] G. Gong and P. Turney, *Hepatitis*, UCI Machine Learning Repository, Dataset, 1983. DOI: 10.24432/C5Q59J.

[7] X. F. C. Sánchez-Díaz, C. Masson and O. J. Mengshoel, 'Regularized feature selection landscapes: An empirical study of multimodality', in *Parallel Problem Solving from Nature – PPSN XVIII*, M. Affenzeller et al., Eds., Cham: Springer Nature Switzerland, 2024, pp. 409–426, ISBN: 978-3-031-70055-2.

[8] X. F. C. Sánchez Díaz, *Replication data for: Regularized feature selection landscapes: An empirical study of multimodality*, 2024. DOI: 10.18710/DQZKMX.

[9] O. J. Mengshoel, *IT3708 Bio-Inspired Artificial Intelligence: Week 12, fitness landscape analysis*, Lecture slides, Department of Computer Science, NTNU, Including slides by Xavier F. C. Sánchez-Díaz, Katherine Malan, and Gabriela Ochoa, 2026.

[10] O. J. Mengshoel, *IT3708 Bio-Inspired Artificial Intelligence: Week 9, multi-objective optimization*, Lecture slides, Department of Computer Science, NTNU, Contributions by Pauline Haddow, Håken Jevne, Kazi Ripon, and Xavier F. C. Sánchez-Díaz, 2026.

[11] X. F. C. Sánchez-Díaz, *IT3708 Bio-Inspired Artificial Intelligence: Week 3, variations on evolutionary, especially genetic, algorithms*, Lecture slides, Department of Computer Science, NTNU, Based on slides by Ole J. Mengshoel, 2026.

[12] S. F. Galán and O. J. Mengshoel, 'Generalized crowding for genetic algorithms', in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '10, New York, NY, USA: Association for Computing Machinery, 2010, pp. 775–782. DOI: 10.1145/1830483.1830625.

[13] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, 'A fast and elitist multiobjective genetic algorithm: Nsga-ii', *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002. DOI: 10.1109/4235.996017.

[14] J. Kennedy and R. C. Eberhart, 'A discrete binary version of the particle swarm algorithm', in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 4104–4108.

[15] J. Å. Bergquist and O. J. Mengshoel, *IT3708 Bio-Inspired Artificial Intelligence: Parameter setting in evolutionary algorithms*, Lecture slides, Department of Computer Science, NTNU, Week 7/8 lecture slides, 2025.

[16] O. J. Mengshoel, *IT3708 Bio-Inspired Artificial Intelligence: Week 4, preserving diversity in evolutionary algorithms*, Lecture slides, Department of Computer Science, NTNU, 2026.