Cellular Automata, $(\mu + \lambda)$ Evolutionary Strategies and Morphogenesis: A CBR Approach

Sergio Martínez Olivera[1 [0009−0007−2845−8565]], Belen Díaz Agudo[1 [0000−0003−2818−027X]], and Juan A. Recio García[1 [0000−0001−8731−6195]]✉

Facultad de Informática, Universidad Complutense de Madrid, Spain
sergim21@ucm.es, belend@ucm.es, jareciog@ucm.es
✉ jareciog@ucm.es (Corresponding author)

Abstract. Modeling complex biological systems such as morphogenetic processes requires finding local interaction rules that produce coherent global patterns from strictly local information. In this paper we present a framework that couples *cellular automata* (CA) with a $(\mu + \lambda)$ *evolutionary strategy* ($(\mu + \lambda)$-ES) to automatically discover transition rules that reproduce target morphogenetic patterns, ranging from classical 1D benchmarks to 2D flag-generation problems. Our key contribution is the integration of this optimisation engine into a *Case-Based Reasoning* (CBR) system: solved cases are stored and reused to warm-start the GA for unseen problems, dramatically reducing convergence time. We evaluate the fitness function across three complementary similarity metrics (SSIM, Jaccard index, Mutual Information) and report their individual and combined impact on rule quality. Experimental results show that the CBR engine reduces the number of generations required by up to 80% while maintaining competitive fitness scores.

Keywords: Cellular automata · Genetic algorithms · Case-based reasoning · Morphogenesis · $\mu + \lambda$ strategy · Emergent computation · Similarity metrics

1 Introduction

Morphogenesis, the process by which biological organisms develop their spatial structure from a uniform mass of cells, has fascinated scientists since Turing's seminal work on reaction-diffusion systems [9,16]. At its core, morphogenesis is governed by *local* biochemical interactions that produce striking *global* order — a phenomenon that computational models struggle to replicate efficiently.

Cellular Automata (CA) are discrete dynamical systems that serve as a natural modeling tool for these processes. In a CA, each cell updates its state according to a fixed local rule depending only on its neighbors [10]. If these underlying rules are successfully found, a deterministic and explainable behavior can be assigned to complex dynamic phenomena whose evolution is, a priori, highly unpredictable. Consequently, this framework does not simply aim to transform a random input pattern into a target output, but rather to explicitly discover the specific logic and local interactions that drive a pattern's transformation.

The central challenge lies in identifying the optimal CA transition rule within an exponentially large search space. For instance, a 2D binary CA with a Moore neighborhood yields 2^{512} candidate rules, making manual or brute-force search impossible. Genetic Algorithms (GAs) are well-suited for this task due to their ability to evaluate enormous search spaces and find high-quality solutions [11,5].

However, a practical limitation of GA-based search is the "utility problem": each new morphogenetic target requires a full optimisation run from scratch, discarding all previously acquired expert knowledge. Following biological parallelism, where evolution reuses successful genetic structures [9], this work proposes integrating CA and an evolutionary strategy optimisation within a Case-Based Reasoning (CBR) engine [1]. This architecture provides the system with a memory to leverage prior experience, evaluating problem similarity to automate configuration and perform transfer learning by injecting successful rules as initial seeds [3].

The main contributions of this paper are:

1. **Implementation of a $\mu + \lambda$ Evolutionary Strategy**: We propose an adaptive mutation strategy to optimise local transition rules, successfully overcoming epistasis and premature convergence commonly found in classical GAs [14].
2. **Design of a 2D Evaluation Framework**: We introduce an advanced framework for artificial morphogenesis by combining structural (multichannel RGB SSIM), spatial (weighted Jaccard Index), and statistical dependence (Mutual Information) metrics [13].
3. **Development of a CBR Engine**: We present a CBR architecture that automates system parameterization featuring topological retrieval and transfer learning of CA rules [3,18], demonstrating a reduction in computational cost of up to 80%.
4. **Systematic Evaluation**: A comprehensive validation on 1D benchmarks (Density Classification, French Flag Problem, *Drosophila* embryogenesis) and 2D flag-generation problems.

To ensure reproducibility, the full implementation is publicly available at https://github.com/sergiio8/TFG_CS, alongside an interactive web interface to properly visualize the 2D spatiotemporal evolutions https://gaia.fdi.ucm.es/research/cellautomata/.

The rest of the paper is organised as follows. Section 2 surveys related work. Section 3 describes the optimisation framework and 2D metrics. Section 4 details the CBR architecture. Section 5 reports experimental results, and Section 6 concludes the paper.

2 Related Work

2.1 Evolving Cellular Automata with Genetic Algorithms

Identifying the optimal CA transition rule to approximate a desired target pattern represents a massive combinatorial challenge. Typically, the transition rule is

encoded as a vector where each position represents a possible configuration of a cell and its neighborhood, and the value dictates the new state of the central cell.

For a 2D binary CA with a von Neumann neighborhood ($N = 4$), the rule table has $2^5 = 32$ entries, yielding 2^{32} candidate rules. This vast search space renders exhaustive search methods infeasible, making GAs particularly well-suited to handle the inherent epistasis of rule tables robustly.

Mitchell et al. [11] provided the first systematic study of GA-evolved CA rules on the Density Classification Problem (DCP), achieving around 82% accuracy on random 149-cell automata. The Gacs–Kurdyumov–Levin (GKL) rule [6], designed by hand, has long served as a reference with approximately 96–98% accuracy at larger sizes.

For 2D shape generation, Mouret et al. [13] demonstrated GA-evolved CAs capable of generating complex geometric shapes, reporting promising results when using fitness functions based on mutual information and marginal entropies.

2.2 Case-Based Reasoning in Optimisation

CBR is a problem-solving paradigm in which new problems are solved by adapting solutions to similar past cases [1]. In the context of evolutionary computation, recent works have successfully applied CBR for the automatic parameter tuning of GAs. For instance, E. Yeguas et al. proposed a Bayesian CBR system that utilises probabilistic models to determine the optimal configuration of evolutionary algorithms [18]. Furthermore, transfer learning within evolutionary frameworks has been explored through selective fine-tuning, where knowledge from previously solved optimisation tasks is reused to bias the initial search towards promising regions [3]. These works provide a foundational justification for our approach of warm-starting the $(\mu + \lambda)$-ES by injecting rules from similar morphogenetic cases.

2.3 Neural Cellular Automata (NCA)

As a prominent alternative, Neural Cellular Automata [12] replace the discrete look-up table with a small convolutional network, enabling continuous-state automata that handle complex textures and regenerative patterns. However, NCA optimize convolutional weights rather than explicit transition rules, making them ill-suited for studying emergent computation: the goal of this work is not to map an input pattern to an output, but to discover the specific local interaction rules that give rise to a target morphogenetic process. This requirement for interpretability and exact rule extraction motivates our choice of deterministic CA over NCA, despite the latter's greater expressive power.

3 Design of the Optimisation Framework

3.1 Cellular Automaton Modeling

To implement the (CA), we utilised the Python library CellPyLib [2]. The transition rules are encoded as a vector where each element dictates the central cell's

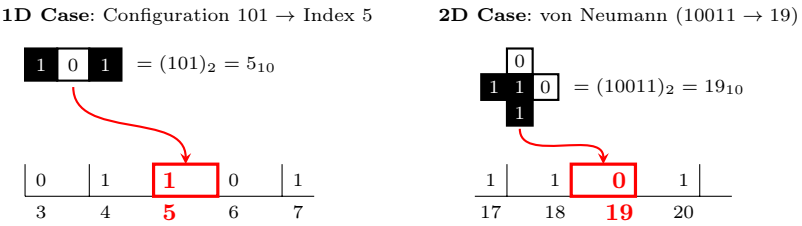


Fig. 1: Mapping logic to retrieve next state from the chromosome

new state. Assuming M states and a neighborhood of N cells, local configurations are mapped to a base-10 index, interpreting the cell states as a base-M number. Consequently, the search space consists of $M^{M^{N+1}}$ possible transition rules, which renders manual search impossible.

Key decisions regarding CA parameterization depend on the problem's topology:

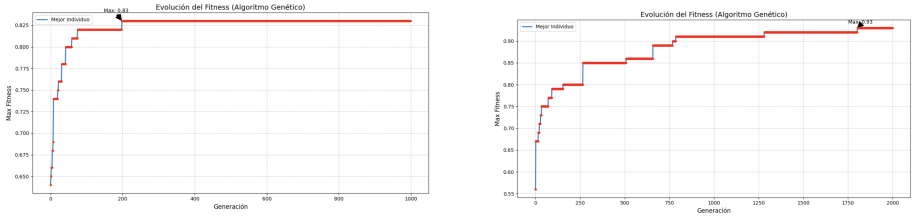
- **Neighborhood selection (2D topologies)**: For binary states, we utilise a Moore neighborhood ($N = 8$). However, for problems with three or more states, a von Neumann neighborhood ($N = 4$) is preferred to prevent exponential growth of the chromosome (e.g., $3^5 = 243$ genes vs. $3^9 = 19683$).
- **Boundary handling**: By default, CellPyLib uses periodic boundaries, suitable when opposite edges share the same state. However, we also implemented fixed boundaries for specific rows or columns, providing the minimal constraints necessary for the automaton to stabilize asymmetric patterns.

3.2 Evolutionary Strategy and Adaptive Mutation

We adopted the $(\mu + \lambda)$-ES strategy [14] —implemented via the DEAP library— to mitigate epistasis and premature convergence. In this approach, λ offspring are generated exclusively through mutation from the current population μ. The next generation is then formed by selecting the best μ individuals from the combined union of parents and offspring. Finally, a 2% diversity mechanism replaces the least fit individuals with random genomes to prevent stagnation and escape local optima.

We further implemented an *adaptive mutation schedule* where the number of mutated genes per offspring, $m(f)$, decreases as a step function of the best individual's fitness f to balance early exploration with fine-grained exploitation near convergence. For complex 2D scenarios —specifically, 2-state automata with a Moore neighborhood yielding a chromosome length of $L = 2^9 = 512$ genes— $m(f)$ is defined as:

$$m(f) = \begin{cases} 25, & \text{if } 0 \leq f < 0.25 \\ 18, & \text{if } 0.25 \leq f < 0.45 \\ 12, & \text{if } 0.45 \leq f < 0.60 \\ 8, & \text{if } 0.60 \leq f < 0.75 \\ 4, & \text{if } 0.75 \leq f \leq 1 \end{cases} \tag{1}$$



(a) Classical GA (b) $(\mu + \lambda)$-ES

Fig. 2: Convergence of maximum fitness over generations in the French Flag Problem. (a) Classical GA stalls at a local optimum due to premature convergence. (b) $(\mu + \lambda)$-ES strategy achieves progressive, staircase-shaped improvement.

where these empirically determined thresholds correspond to approximately 5%, 3.5%, 2.3%, 1.6%, and 0.8% of L, respectively. This ensures the schedule scales proportionally to the chromosome length and generalizes across problem sizes.

3.3 Evaluation Framework for 2D Morphogenesis

While a simple accuracy metric suffices for one-dimensional cases, evaluating two-dimensional morphogenesis requires capturing spatial, structural, and statistical nuances. Drawing inspiration from standard perceptual and statistical measures, we designed a composite framework integrating:

*1. Multichannel RGB Structural Similarity Index (*SSIM*).* Given windows x and y, SSIM [17] evaluates similarity based on luminance, contrast, and structure. Since CA states are nominal, we map them into an RGB color space ($[0, 255]^3$) before computation to accurately penalize visual noise:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

2. State-wise Jaccard Index. To rigorously evaluate spatial exactness, we average the Jaccard index [8] for each state $s \in \varepsilon$ independently, which equitably penalizes both false negatives and false positives:

$$\text{Jaccard}(X, Y) = \frac{1}{|\varepsilon|} \sum_{s \in \varepsilon} \frac{|X_s \cap Y_s|}{|X_s \cup Y_s|} \tag{3}$$

*3. Mutual Information (*MI*).* To handle cases where states might be inverted or translated, we incorporate MI [4] to measure the statistical dependence and reduction of uncertainty:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \tag{4}$$

Finally, the global fitness function is defined as a convex combination of these three metrics, ensuring the final score remains bounded within $[0, 1]$:

$$f_{\text{fit}} = \alpha \cdot \text{SSIM} + \beta \cdot \text{Jaccard} + \gamma \cdot \text{MI} \tag{5}$$

where $\alpha + \beta + \gamma = 1$, and $\text{SSIM}, \text{Jaccard}, \text{MI} \in [0, 1]$.

4 CBR Engine

Configuring the CA and the evolutionary strategy from scratch for every new target pattern requires extensive manual testing and immense computational effort. While standard Machine Learning techniques could theoretically predict these parameters, they depend on massive training datasets and struggle to map a 2D image matrix to a complex multi-variable algorithmic configuration.

Each fully optimised CA rule encodes expert knowledge about how to generate a specific morphogenetic pattern. Rather than discarding this knowledge after solving each problem, the CBR engine stores it as a *case* and retrieves structurally similar cases when a new problem is presented. This follows the classic 4R CBR cycle: Retrieve, Reuse, Revise, Retain.

Each case $\mathcal{C}$ is formalized as a tuple:

$$\mathcal{C} = \langle P,\ \varphi^*,\ \Theta_{CA},\ \Theta_{GA},\ f^* \rangle, \tag{6}$$

where P is the problem descriptor (target pattern matrix, dimensionality, number of states, presence of horizontal/vertical stripes, number of connected components, uniform borders, and detected geometric patterns), φ^* is the best rule found, Θ_{CA} and Θ_{GA} are the CA and GA hyperparameter configurations, and f^* is the final fitness achieved.

4.1 Similarity Metric for Case Retrieval

When a new target pattern P_{new} is introduced, its topological features are extracted. The engine evaluates its similarity against the existing case base using a composite similarity function. The weights assigned to each similarity component were determined empirically through iterative testing on the initial case base. The dominant weight of Sim_{struct} (0.5) reflects its role as the primary predictor of rule transferability, as structural topology directly determines neighborhood activation patterns. The lower weight of Sim_{states} (0.1) is justified by the binary-to-binary compatibility check already embedded in the transfer learning phase.

$$Sim = 0.2 \cdot Sim_{topo} + 0.5 \cdot Sim_{struct} + 0.1 \cdot Sim_{states} + 0.2 \cdot Sim_{dims} \tag{7}$$

To ensure reproducibility, each component is strictly bounded within $[0, 1]$:

i) Sim_{topo} computes the weighted average of boolean matches for topological features (e.g., stripes, borders).

ii) Sim_{struct} averages three sub-metrics: Mutual Information (statistical dependence), the inverse Euclidean distance of state distribution histograms, and connected components similarity calculated as $1/(1 + |C_1 - C_2|)$.

iii) Sim_{states} yields 1 if $|N_1 - N_2| = 0$, 0.5 if the difference is 1, and 0 otherwise.

iv) Sim_{dims} returns 1 if both problems operate in the same dimensionality.

An innovation in our retrieval phase is *rotational invariance detection*. The engine compares the new problem not only with the original base cases but also with their 90° rotated versions. If a rotated image yields a higher similarity, the system retrieves it with a 0.8 penalty multiplier, successfully identifying problems that are structurally identical but differently oriented.

4.2 Reuse, Revision and Retention

Reuse (Transfer Learning). Based on similarity scores, implicit parameters are automated, qualitative settings are inherited from the best case, and quantitative ones (e.g., mutation, population size) are calculated as a weighted average. Crucially, if the retrieved case shares the same state space and neighborhood type, *Transfer Learning* is applied by injecting its best transition rule into the new GA's initial population: 5% as exact clones and 5% as slightly mutated copies. This specific 10% ratio seeds high-quality genetic material while deliberately maintaining diversity; a larger proportion would risk flooding the population with overly similar individuals, reducing diversity and driving the search toward local optima. For rotated matches, a topological permutation recalculates base-10 indices to correctly map the rotated Moore or von Neumann neighborhoods before injection, providing an optimized seed that drastically improves convergence speeds.

Revision and Retention. The evolutionary framework runs using this automated parameterization and the injected seeds. If the best fitness f^*_{new} exceeds a dynamically adapting threshold θ_{retain} —which grows with the case base size (0.65, 0.75, and 0.85 for bases up to 10, 15, and beyond cases) to ensure only increasingly competitive solutions are stored— the new case is retained. Potentially contradictory cases are handled implicitly during retrieval: the engine blends the top-3 most similar cases via a weighted average of their configurations, successfully diluting the influence of any anomalous individual case.

5 Experiments and Results

To populate the initial CBR case base and validate our evolutionary framework, we first evaluated classical 1D and 2D morphogenesis problems. Subsequently, the CBR engine was tested against a novel dataset of 12 patterns to measure its efficiency compared to random and heuristic approaches.

Table 1: Hyperparameter configurations for the generic baseline heuristic and the specialized CBR base cases. Fitness weights are represented as a percentage tuple: (Accuracy, SSIM, Jaccard, MI).

Configuration	Neigh.	Pop.	Gen.	Steps (T)	Init.	Mutations	Weights (%)
Heuristic Baseline	Var.	250	500	60	Variable	Adapt. $\{15, 10, 6, 3\}$	(0, 35, 35, 30)
Drosophila (1D)	3	250	1000	90	Random	Static (3 genes)	(100, 0, 0, 0)
France (1D)	1	250	2000	90	Random	Static (1 gene)	(100, 0, 0, 0)
Hungary (2D)	Neumann	300	2000	75	Random	Adapt. $\{10, 8, 6, 4, 2\}$	(0, 80, 20, 0)
Japan (2D)	Moore	1000	2000	20	Seed	Adapt. $\{25, 18, 12, 8, 4\}$	(0, 20, 20, 60)

All experiments were executed on an Apple MacBook Air equipped with an Apple M1 chip (8-core CPU) and 8 GB of unified memory, running macOS. Table 1 details the fine-tuned parameters for the initial base cases stored in the CBR memory, alongside the generic Expert Heuristic baseline configuration that will be evaluated later as a control group.

5.1 1D Morphogenesis Base Cases

Density Classification Problem (DCP): On the DCP benchmark [7], our classic GA was compared against the GKL rule across 250 random configurations ($N \in \{50, 75, 150\}$). While the GA achieved competitive accuracy (93.80% for $N = 50$, dropping to 91.40% for $N = 150$), the GKL rule exhibited scale invariance, scoring between 96.20% and 98.00%.

French Flag Problem: In the French Flag problem, the $\mu + \lambda$ strategy successfully overcame the premature convergence commonly observed in classic GAs, achieving a 95% fitness and demonstrating excellent generalisation across different automaton sizes (e.g., 30, 60, and 90 cells) to form the three distinct axial regions.

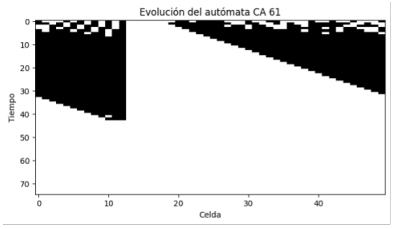
Drosophila Melanogaster Embryogenesis: As a systems biology proof-of-concept, the 1D framework modeled *Drosophila* embryogenesis. It successfully self-organised a random initial state into the biological 7-stripe pattern (70% fitness, 1000 gen.), demonstrating its ability to infer local rules from macroscopic phenotypes.

5.2 2D Morphogenesis Base Cases

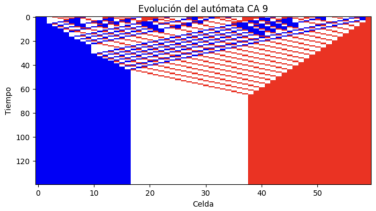
Hungarian Flag (Horizontal Stripes): Relying solely on SSIM initially resulted in monochromatic blocks. However, employing our composite metric—combining SSIM (80%) with the state-wise Jaccard Index (20%)—alongside dynamic mutation allowed the automaton to escape local minima and form the flag.

Japanese Flag (Temporal Emergence): Growing a circle from a single central seed utilised a composite metric heavily weighted towards MI (60%), supported by SSIM (20%) and the state-wise Jaccard Index (20%). We observed emergent

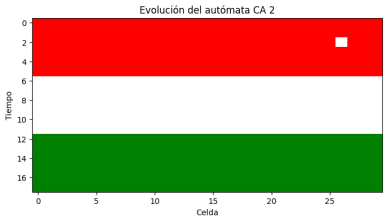
temporal coordination: the automaton expanded immediately under a strict 15-timestep limit, but learned to delay its growth when given 20-30 timesteps to avoid exceeding the target radius.



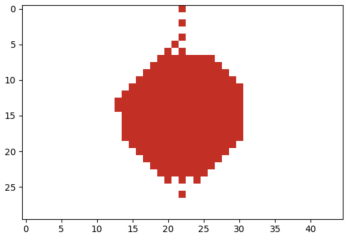
(a) Density Classification Problem



(b) French Flag ($N = 60$)



(c) Hungary (2D)



(d) Japan (2D)

Fig. 3: Morphogenesis Base Cases: 1D spatiotemporal emergence (top) and 2D evolved patterns (bottom).

5.3 CBR Evaluation and Performance Analysis

The CBR engine was tested on 12 new topological problems against two baselines: Random (starts randomly without configuration guidance) and Expert Heuristic. The latter was constructed by analyzing the parameterizations that proved effective across the initial base cases, applying a fixed configuration (pop: 250, adaptive mutation, weights: 35% SSIM / 35% Jaccard / 30% MI). It only adjusts image-dependent parameters automatically: neighborhood type (Moore for binary, von Neumann for ≥ 3 states), boundary conditions, initial state, and chromosome length. With no transfer learning and a random starting population, this baseline isolates the specific contribution of the CBR seeding mechanism.

As shown in Fig. 5, the Random approach consistently failed to converge to acceptable fitness levels. The CBR engine matched or slightly exceeded the heuristic fitness while drastically reducing the number of generations required to converge, demonstrating immense computational savings.

Beyond converging faster, the CBR-seeded evolutionary process demonstrated remarkable topological adaptation. For the Italian flag, the engine retrieved the Hungarian flag base case via rotational similarity, injected the permuted 1D rule,

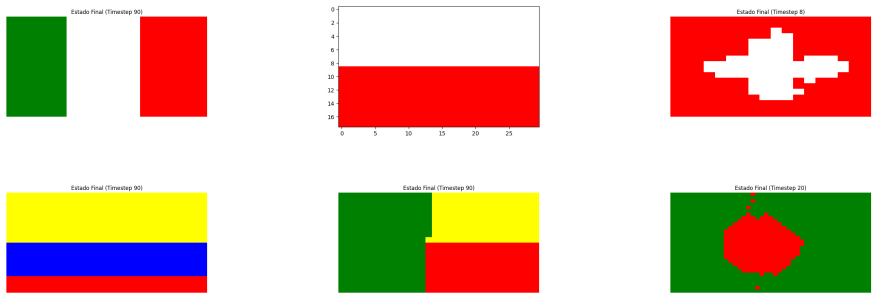


Fig. 4: Evolved 2D patterns successfully generated using the CBR Engine.

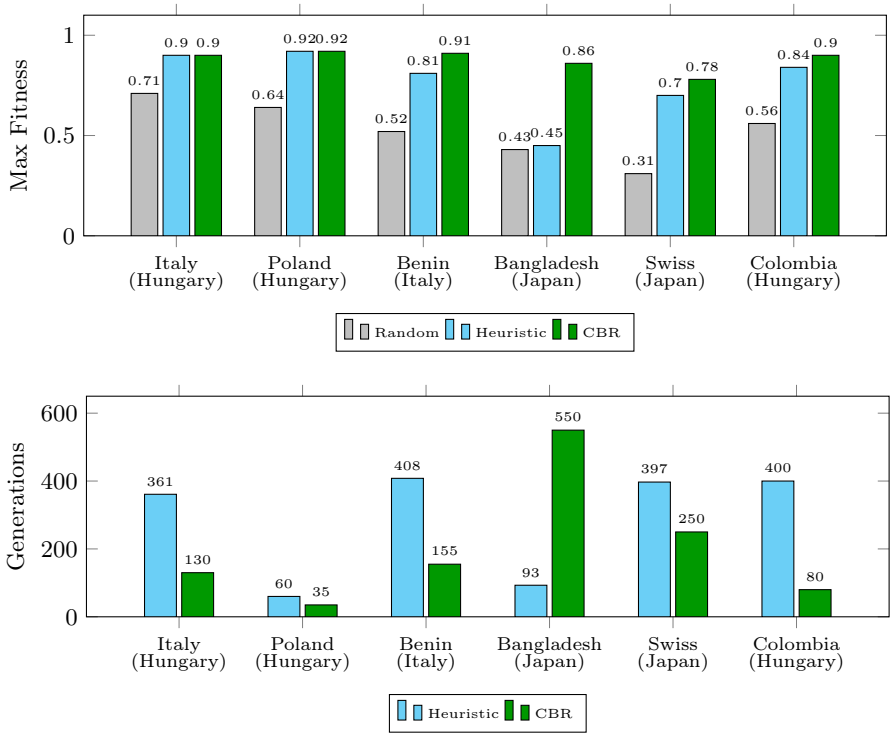
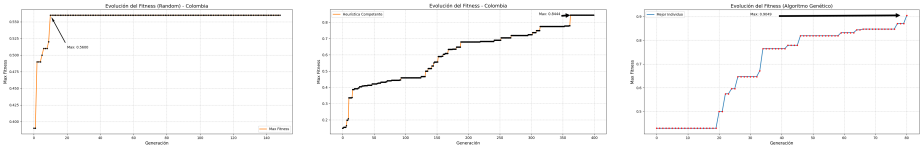


Fig. 5: Performance and computational cost comparison. Note: Bangladesh-CBR reaches the stopping condition (550 gen., fitness 0.86) vs. heuristic premature convergence (93 gen., fitness 0.45). The labels below the x-axis indicate the best matched base case retrieved by the CBR engine.

and successfully evolved vertical stripes. In the Bangladesh flag problem, the engine retrieved Japan and successfully adapted the injected rule to perform a spatial translation of the circle.



(a) Colombia (Random) (b) Colombia (Heuristic) (c) Colombia (CBR)

Fig. 6: Comparison of the maximum fitness evolution for the Colombia flag problem under the three studied scenarios: random search, heuristic configuration, and optimisation using the CBR engine.

6 Discussion and Conclusions

This work demonstrates that integrating evolutionary strategies and CBR provides a robust framework for optimizing CA rules in morphogenesis. The $(\mu + \lambda)$-ES strategy outperforms classical GAs via adaptive mutation, while the composite fitness framework (SSIM, Jaccard, MI) guides evolution toward structurally sound patterns. Crucially, with a negligible retrieval overhead (< 15 ms) offsetting a high 4 min/generation computation cost, the CBR engine improves efficiency via transfer learning and rotational invariance, reducing the required generations by up to 80% for Colombia (320 gen. / 21 h), 64% for Italy (231 gen. / 15.5 h), and 62% for Benin (253 gen. / 17 h) while maintaining competitive fitness.

Limitations and Future Work. While effective for clustered and axial patterns, deterministic discrete CAs struggle with: (i) perfectly symmetrical shapes (e.g., Austria or Czech flags), which require symmetry-breaking directional bias, and (ii) continuous animal textures (e.g., leopard coats). Additionally, high computational demands (up to 4 min/generation) restricted our evaluation to single-run executions. Consequently, multi-seed statistical validation (means, standard deviations) and granular ablation studies—such as isolating rule injection from rotational handling—were omitted. Overcoming these constraints through extensive statistical runs remains an immediate future priority.

Hence, future work will focus on three main pillars: (i) automating case-base expansion by systematically generating random target patterns to enrich the case base, (ii) exploring Neural Cellular Automata (NCA) to leverage continuous states and convolutional networks for complex textures, and (iii) moving beyond synthetic flag benchmarks toward real biological phenomena, such as tumor growth or wound healing [15], to provide a stronger validation of the framework.

Acknowledgments. Supported by UCM (Group 921330) and Grant PID2023-150566OB-I00 (AUDITIA-X) funded by MCIN/AEI/10.13039/501100011033 and ERDF/EU (https://gaia.fdi.ucm.es/research/auditia-x/).

Generative AI Statement: Claude Sonnet 4.5 and Gemini Pro were used for proofreading and improving the English language of the manuscript, as well as assisting with formatting. The authors take full responsibility for the content of the publication.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications **7**(1), 39–59 (1994)
2. Antunes, L.M.: Cellpylib: A python library for working with cellular automata. Journal of Open Source Software **6**(67), 3608 (2021). https://doi.org/10.21105/joss.03608, https://doi.org/10.21105/joss.03608
3. Colan, J., Davila, A., Hasegawa, Y.: Transfer learning optimization based on evolutionary selective fine tuning. In: Proceedings of the 20th International Conference on Case-Based Reasoning. pp. 210–225. Springer (2023)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, 2nd edn. (2006)
5. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research **13**, 2171–2175 (2012), http://jmlr.org/papers/v13/fortin12a.html
6. Gacs, P., Kurdyumov, G., Levin, L.A.: One-dimensional homogenous arrays iterative organized from cells. Problemy Peredachi Informatsii **14**(3), 92–98 (1978)
7. Hoyrup, M.: The density classification problem on finite 1d cellular automata. https://inria.hal.science/hal-01290290/document (2011), accessed: 2025-11-06
8. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist **11**(2), 37–50 (1912)
9. Landge, A.N., Jordan, B.M., Diego, X., Müller, P.: Pattern formation mechanisms of self-organizing reaction-diffusion systems. Developmental Biology **460**(1), 2–11 (2020). https://doi.org/10.1016/j.ydbio.2019.10.031
10. McIntosh, G.: Clases de wolfram. https://delta.cs.cinvestav.mx/~mcintosh/comun/tesismaestria/genaro/node7.html (2001), accedido: 2025-09-29
11. Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular automata with genetic algorithms: A review of recent work. Proceedings of the First International Conference on Evolutionary Computation and Its Applications (1996), https://csc.ucdavis.edu/~evca/Papers/evca-review.pdf
12. Mordvintsev, A., Randazzo, E., Niklasson, E., Levin, M.: Growing neural cellular automata. Distill (2020), https://distill.pub/2020/growing-ca/
13. Mouret, J.B., Doncieux, S.: Using a genetic algorithm to evolve cellular automata for 2d/3d computational development. In: HAL Open Science (2018), https://hal.science/hal-01741125/document
14. Schwefel, H.P.: Evolution and optimum seeking: the prediction of numerical processes. John Wiley & Sons, Inc. (1993)
15. Silveira, P.S.P., Massad, E.: Modeling and simulating morphological evolution in an artificial life environment. Computers in Biology and Medicine **27**(4), 255–266 (1997). https://doi.org/10.1016/S0010-4809(97)00015-X
16. Turing, A.M.: The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences **237**(641), 37–72 (1952)
17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
18. Yeguas, E., Luzón, M.V., Pavón, R., Laza, R., Arroyo, G., Díaz, F.: Automatic parameter tuning for evolutionary algorithms using a bayesian case-based reasoning system. Information Sciences (2018)