AHISTER: Adaptive Hierarchical Isolated Sub-Branching with Tournament-Based Expansion and Reduction for Symbolic Regression

The Mathematical Colosseum

Markus Pollak
Diafunc
https://diafunc.com
markus.pollak@diafunc.com

Abstract

Symbolic regression seeks to discover closed-form mathematical expressions that accurately describe observed data, yet standard genetic programming (GP) approaches frequently suffer from premature convergence and diversity collapse as populations stagnate on local optima. Island-model GP mitigates this through spatial isolation, but relies on fixed topologies and periodic migration schedules that are insensitive to the evolutionary dynamics of individual subpopulations. We introduce **AHISTER** (**A**daptive **H**ierarchical **I**solated **S**ub-Branching with **T**ournament-Based **E**xpansion and **R**eduction), a dynamic multi-branch evolutionary framework—a *mathematical colosseum* in which independent evolutionary branches first develop in isolation, refining their populations without outside interference, until stagnation signals their readiness for the arena. AHISTER maintains a coordinating branch that aggregates a real-time Hall of Fame from asynchronous executing sub-branches. When a sub-branch enters the arena, it is matched against an equally mature opponent; their genetic material is asymmetrically partitioned into an elite *tournament branch* (intensification) and a diverse *advanced branch* (diversification), enabling simultaneous exploitation and exploration from each stagnation event. A dual stagnation clock governs termination at both the local branch and global coordination levels. We describe the method in detail and outline an experimental framework for evaluating its effectiveness against standard GP, island-model GP, and state-of-the-art symbolic regression systems on established benchmarks.

1 Introduction

Symbolic regression (SR) is the task of discovering closed-form mathematical expressions that describe the relationship between input variables and an observed response, given only a dataset of input–output pairs. Unlike parametric regression, which assumes a fixed functional form and optimizes its coefficients, SR searches over the space of both structure and parameters simultaneously. This makes SR a powerful tool for scientific discovery (Udrescu and Tegmark, 2020), data-driven modeling, and interpretable machine learning (Cranmer, 2023), but also renders it a combinatorially hard optimization problem.

Genetic programming (GP) (Koza, 1992) remains the dominant paradigm for symbolic regression. GP evolves a population of candidate expressions (represented as syntax trees) through selection, crossover, and mutation operators, guided by a fitness function that balances predictive accuracy against expression complexity. Despite decades of refinement (Poli et al., 2008), standard GP suffers from well-documented pathologies: premature convergence to local optima, code bloat, and diversity collapse within the population (Burke et al., 2004).

Island models and their limitations. The island-model GP (Whitley et al., 1999; Skolicki and De Jong, 2005) addresses diversity collapse by maintaining multiple isolated subpopulations (islands) that evolve independently and periodically exchange individuals through *migration*. While effective at preserving diversity, island models impose a rigid structure: the number of islands, migration topology, migration rate, and migration interval are all fixed hyperparameters that remain constant throughout the evolutionary run, regardless of the actual dynamics of each subpopulation. An island that has converged to a productive region receives the same migration treatment as one that is hopelessly stuck. This one-size-fits-all approach wastes computational resources and fails to adapt to the heterogeneous convergence behavior that arises in practice.

Contributions. We introduce AHISTER (**A**daptive **H**ierarchical **I**solated **S**ub-Branching with **T**ournament-Based **E**xpansion and **R**eduction), a dynamic multi-branch evolutionary framework for symbolic regression that replaces the static topology of island models with an adaptive, event-driven architecture.

Conceptually, AHISTER operates as a *mathematical colosseum*: independent evolutionary branches first develop in isolation, refining their populations through standard genetic programming without outside interference. Only when a branch has exhausted its individual potential—signaled by stagnation—does it enter the arena, where it is matched against an equally mature opponent. The genetic legacies of both contenders are then selectively recombined to spawn the next generation of competitors, one forged from the elite and another from the diverse rank-and-file.

The key contributions are:

1. **Dynamic branch lifecycle.** Executing sub-branches are spawned, evolve independently, and are terminated upon stagnation or reaching a hard generation cap. Unlike fixed islands, the set of active branches changes throughout the run (Section 3.1).
2. **Stagnation as a constructive signal.** Rather than treating stagnation as a failure, AHISTER uses it to trigger genetic material recycling. Stagnated branches enter a pairing pool where they are matched by evolutionary maturity and their populations are partitioned into elite and non-elite components (Section 3.6).
3. **Asymmetric offspring branches.** Each pairing event produces two successor branches: a *tournament branch* seeded from the combined elites of both parents (intensification) and an *advanced branch* seeded from the combined non-elites (diversification). This simultaneously drives exploitation and exploration from a single stagnation event (Section 3.7).
4. **Real-time Hall of Fame.** A coordinating branch maintains a global bounded leaderboard that is updated after every generation of every executing sub-branch, providing a persistent collective memory across all branches (Section 3.4).
5. **Dual stagnation clocks.** Independent stagnation thresholds at the executing branch level (local) and the coordinating branch level (global) govern termination at different time scales (Section 3.5).

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the AHISTER method in detail. Section 4 outlines the experimental framework. Section 5 discusses computational considerations, and Section 6 concludes.

2 Related Work

Genetic programming for symbolic regression. Koza (Koza, 1992) introduced tree-based GP as the foundational approach to symbolic regression, evolving populations of syntax trees through subtree crossover and point mutation. Subsequent work has focused on controlling bloat (Poli et al., 2008), improving operator design, and incorporating local search for numeric constants. Modern GP systems such as Operon (Kommenda et al., 2020) employ efficient data structures and highly optimized evaluation to scale to large datasets, while PySR (Cranmer, 2023) combines multi-population search with simulated annealing and has gained traction for scientific discovery applications.

Island-model evolutionary algorithms. The island (or multi-deme) model (Whitley et al., 1999) distributes the population across spatially isolated subpopulations that evolve independently and exchange individuals at fixed intervals through migration. Skolicki and De Jong (Skolicki and De Jong, 2005) analyzed the sensitivity of island models to topology, migration rate, and migration interval, demonstrating that these hyperparameters significantly affect performance but are difficult to tune a priori. Adaptive migration strategies (Lässig and Sudholt, 2010) attempt to address this by dynamically adjusting migration rates, but retain the fixed-island topology and synchronous communication model.

Multi-objective symbolic regression. Multi-objective formulations treat accuracy and complexity as competing objectives, using Pareto-based selection (Deb et al., 2002) to maintain a front of trade-off solutions. This naturally preserves diversity along the complexity axis but does not address the premature convergence of individual populations within complexity niches.

Quality-diversity and novelty search. Quality-diversity algorithms such as MAP-Elites (Mouret and Clune, 2015) maintain an archive of diverse high-performing solutions indexed by behavioral descriptors. Novelty search (Lehman and Stanley, 2011) replaces fitness-driven selection with novelty-driven exploration. While these approaches excel at maintaining diversity, they require domain-specific behavior characterization and have seen limited application to symbolic regression.

Positioning of AHISTER. AHISTER differs from island models in three fundamental ways: (1) branches have a finite lifecycle governed by stagnation detection rather than running for the entire search duration; (2) genetic material transfer is event-driven (triggered by stagnation) rather than periodic; and (3) offspring branches are asymmetrically seeded, simultaneously pursuing intensification and diversification. Unlike multi-objective approaches, AHISTER operates on a scalar quality function but achieves diversity through its multi-branch architecture. Unlike quality-diversity methods, AHISTER requires no behavior characterization—diversity emerges naturally from the dynamic creation and destruction of independent evolutionary lineages.

3 Method

We now describe the AHISTER framework in detail. Figure 1 provides an overview of the architecture, and Algorithm 1 formalizes the coordinating branch loop.

3.1 Architecture Overview

AHISTER organizes the evolutionary search into a two-level hierarchy of *branches*. Each branch $\mathcal{B}$ operates in one of two modes:

- **Coordinating** ($\mathcal{B}_{\text{coord}}$): A single top-level branch that does not perform evolutionary computation itself. It manages a pool of executing sub-branches, aggregates their results into a Hall of Fame, and orchestrates the stagnation-driven pairing protocol.
- **Executing** ($\mathcal{B}_{\text{exec}}^{(i)}, i = 1, \dots, K$): Sub-branches that perform the actual genetic programming search. Each executing branch maintains an independent population $\mathcal{P}_i$ of size N, evolves it

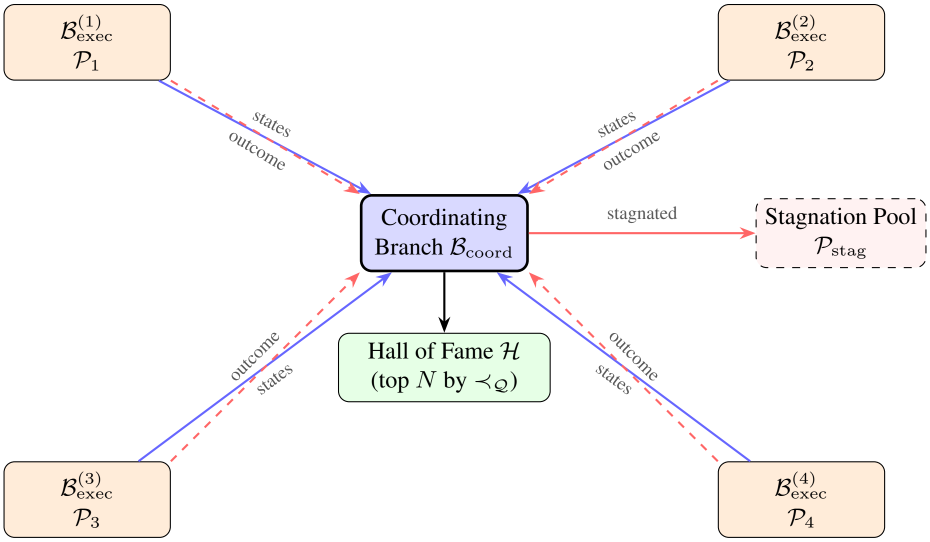

Figure 1: **AHISTER architecture.** A coordinating branch $\mathcal{B}_{\text{coord}}$ manages $K = 4$ executing sub-branches. Each executing branch emits state updates (solid blue arrows) after every generation and an outcome (dashed red arrows) upon termination. States are merged into the Hall of Fame $\mathcal{H}$. Completed branches enter the stagnation pool for pairing.

through standard GP operators, and communicates its state to the coordinating branch after every generation.

Communication between the coordinating and executing branches is asynchronous and unidirectional: executing branches emit *state updates* (containing their full current population) and *outcomes* (upon termination) to the coordinating branch via channels. The coordinating branch does not send information back to executing branches during their lifetime. This isolation ensures that each executing branch explores its region of the search space without interference.

At initialization, the coordinating branch launches K exploratory executing branches (default $K = 4$), each initialized with a random population drawn from the search space. As branches stagnate and are replaced, the total number of active branches fluctuates around K.

3.2 Within-Branch Evolution

Each executing branch runs a standard generational GP loop. In each generation, the current population $\mathcal{P}$ of size N (default $N = 100$) is transformed into a new population $\mathcal{P}'$ through the following steps:

1. **Elite preservation.** The top α fraction (default $\alpha = 0.05$) of individuals by quality are carried forward unconditionally into $\mathcal{P}'$.
2. **Selection.** A subset of the remaining individuals is selected for survival.
3. **Population expansion.** The population is expanded back to size N by sampling new individuals from three weighted operators:
    - *Creation* (weight $w_c = 0.10$): a new random expression is sampled from the search space.
    - *Crossover* (weight $w_x = 0.50$): two parent genotypes are selected and recombined via subtree crossover.

  - *Mutation* (weight $w_m = 0.40$): a single parent genotype is modified via subtree mutation.

4. **Refinement.** Each new genotype undergoes algebraic simplification, numeric constant optimization, and domain validation before being admitted to $\mathcal{P}'$.
5. **State emission.** After each generation, the branch emits a state update containing the full population $\mathcal{P}'$ to the coordinating branch.

3.3 Quality Function

The quality of a genotype g is a scalar $\mathcal{Q}(g) \in [0,1]$ computed as a weighted combination of predictive accuracy and structural simplicity:

$$\mathcal{Q}(g) = \frac{w_\varepsilon \cdot A(g) + w_\mathcal{C} \cdot \mathcal{S}(g)}{w_\varepsilon + w_\mathcal{C}} \tag{1}$$

where $w_\varepsilon = 0.95$ and $w_\mathcal{C} = 0.05$ are the error and complexity weights, respectively.

The **accuracy** term uses a logarithmic scale that rewards improvements across orders of magnitude:

$$A(g) = \min\left(\frac{-\log_{10}\big(\max(\varepsilon(g),\, 10^{-10})\big)}{6},\ 1 \right) \tag{2}$$

where $\varepsilon(g)$ is the prediction error (e.g., RMSE). This maps errors from 10^0 to 10^{-6} linearly onto $[0,1]$, with a floor at 10^{-10} for numerical stability.

The **simplicity** term penalizes structural complexity using a BIC-inspired logarithmic penalty:

$$\mathcal{S}(g) = 1 - \frac{\ln\big(\max(\mathcal{C}(g),\, 1)\big)}{\ln\big(\max(\mathcal{C}_{\max},\, 2)\big)} \tag{3}$$

where $\mathcal{C}(g)$ is the structural complexity of the expression tree and $\mathcal{C}_{\max}$ is the maximum allowed complexity. The floors ($\mathcal{C} \geq 1$, $\mathcal{C}_{\max} \geq 2$) prevent degenerate cases ($\ln 0$ and division by $\ln 1$). The logarithmic scaling ensures that the penalty is sublinear: doubling complexity has a diminishing marginal cost, reflecting the intuition that adding nodes to a simple expression is more costly than adding the same number to an already-complex one.

3.4 Hall of Fame

The coordinating branch maintains a **Hall of Fame** $\mathcal{H}$—a bounded population of size N that serves as the collective memory of the entire search. The Hall of Fame is updated whenever a state arrives from any executing sub-branch:

$$\mathcal{H} \leftarrow \mathrm{top}_N\big(\mathcal{H} \cup \mathcal{P}_{\text{incoming}},\ \prec_\mathcal{Q}\big) \tag{4}$$

where $\prec_\mathcal{Q}$ is a total ordering that sorts individuals by descending quality, then ascending error, then descending timestamp (most recent wins ties). Each merge operation increments the coordinating generation counter τ_{coord}.

The Hall of Fame serves two purposes: (1) it preserves the best-ever solutions found by any branch across the entire search history, and (2) its improvement rate drives the coordinating stagnation clock (Section 3.5). The best genotype g^* is tracked globally and updated whenever a strictly superior individual appears (lower complexity, higher quality, or lower error with all other metrics equal or better).

3.5 Stagnation Detection and Branch Lifecycle

AHISTER uses a **dual stagnation clock** operating at two time scales:

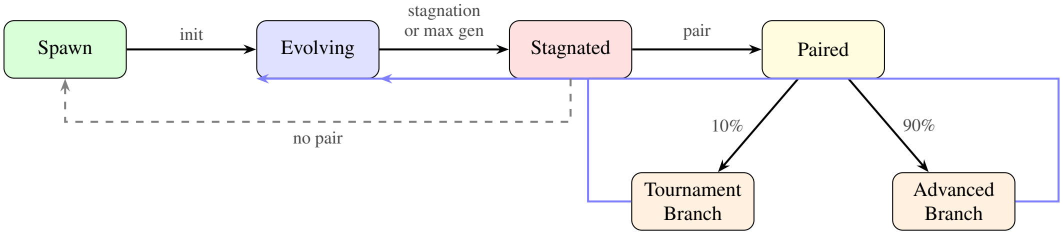

Figure 2: **Branch lifecycle.** A branch evolves until stagnation or the hard generation cap, then enters the stagnation pool for pairing. Paired branches produce a tournament branch (from elites) and an advanced branch (from non-elites), both re-entering the cycle. If no pair is available, an exploratory branch is launched.

Executing branch stagnation. An executing branch $\mathcal{B}_{\text{exec}}^{(i)}$ is considered stagnated when its best quality has not improved for Δ_{exec} consecutive generations (default $\Delta_{\text{exec}} = 100$). Additionally, a hard cap of $\tau_{\max}$ generations (default $\tau_{\max} = 1{,}000$) terminates any branch regardless of improvement, ensuring fresh genetic material enters the system even when a branch improves slowly. Upon termination (whether by stagnation or hard cap), the branch emits an *outcome* containing its final population, partitioned into elite and non-elite components.

Coordinating stagnation. The coordinating branch terminates when the Hall of Fame has not improved for Δ_{coord} coordinating generations (default $\Delta_{\text{coord}} = 10{,}000$). Since each state arrival from any sub-branch increments τ_{coord} by one, and K branches run in parallel, the coordinating clock advances approximately K times faster than the wall-clock time of a single branch.

3.6 Maturity-Based Pairing

When an executing branch completes, its outcome is placed in a **stagnation pool** $\mathcal{P}_{\text{stag}}$. The coordinating branch then attempts to form a pair by selecting the two outcomes in $\mathcal{P}_{\text{stag}}$ whose generation counts are closest:

$$(o_a, o_b) = \operatorname*{arg\,min}_{(o_i, o_j) \in \mathcal{P}_{\text{stag}}^2,\; i \neq j} |\tau(o_i) - \tau(o_j)| \tag{5}$$

This **maturity-based pairing** criterion ensures that branches at similar stages of evolutionary development are combined. The rationale is that branches that have explored their search space to a comparable depth are more likely to have complementary discoveries, as opposed to pairing a deeply converged branch with one that was terminated early.

If only one outcome is available in $\mathcal{P}_{\text{stag}}$ (no pair can be formed), the coordinating branch launches a new exploratory branch with a random initial population to maintain search throughput while waiting for a second stagnation event.

3.7 Asymmetric Offspring

Once a pair (o_a, o_b) is formed, the final populations of both outcomes are partitioned into elite and non-elite components. Let the sorted population of outcome o be $\mathcal{P}(o)$ and define:

$$\mathcal{E}(o) = \text{top } \lfloor 0.1 \cdot |\mathcal{P}(o)| \rfloor \text{ individuals by quality} \tag{6}$$

$$\overline{\mathcal{E}}(o) = \mathcal{P}(o) \setminus \mathcal{E}(o) \tag{7}$$

Two successor branches are then launched:

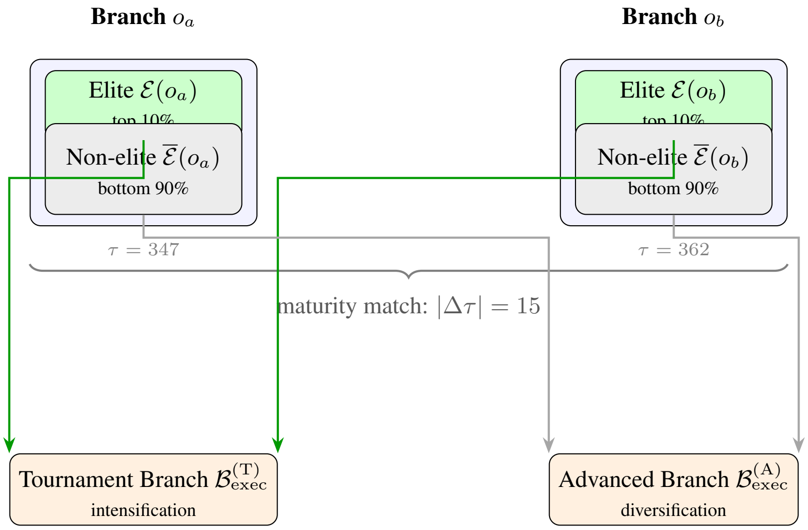

Figure 3: **Maturity-based pairing and asymmetric offspring.** Two stagnated branches with similar generation counts ($|\Delta\tau| = 15$) are paired. Their elite fractions (top 10%) are combined to seed a tournament branch (intensification), while their non-elite fractions (bottom 90%) seed an advanced branch (diversification).

1. **Tournament branch** $\mathcal{B}_{\text{exec}}^{(\text{T})}$: Initialized with the gene pool $\mathcal{G}_{\text{T}} = \mathcal{E}(o_a) \cup \mathcal{E}(o_b)$. This branch concentrates the highest-quality solutions from both parents, driving *intensification*—a focused search in the neighborhood of already-proven expressions.
2. **Advanced branch** $\mathcal{B}_{\text{exec}}^{(\text{A})}$: Initialized with the gene pool $\mathcal{G}_{\text{A}} = \overline{\mathcal{E}}(o_a) \cup \overline{\mathcal{E}}(o_b)$. This branch recombines the broader, more diverse genetic material from both parents, driving *diversification*—exploration of structural variants that did not reach elite status but may contain useful building blocks.

Both branches evolve independently with fresh random seeds derived from the coordinating branch's random number generator. This asymmetric partitioning is the central novelty of AHISTER: a single stagnation event simultaneously produces one branch that exploits proven solutions and another that explores diverse alternatives, without requiring any explicit diversity maintenance mechanism.

3.8 Algorithm Summary

Algorithm 1 summarizes the coordinating branch loop, and Algorithm 2 formalizes the pairing and offspring generation protocol.

4 Experimental Evaluation

We outline the experimental framework for evaluating AHISTER. Full results will be reported in a subsequent version of this paper.

4.1 Benchmarks

We evaluate on three benchmark suites:

Table 1: **AHISTER default hyperparameters.**

Parameter	Default	Description
Architecture		
K (executing branches)	4	Number of parallel executing sub-branches
N (population size)	100	Population size per branch / Hall of Fame capacity
Within-branch evolution		
α (elite fraction)	5%	Fraction preserved unconditionally each generation
w_c (creation weight)	0.10	Probability of random new expression
w_x (crossover weight)	0.50	Probability of subtree crossover
w_m (mutation weight)	0.40	Probability of subtree mutation
Stagnation and termination		
Δ_{exec}	100	Executing branch stagnation threshold (generations)
$\tau_{\max}$	1,000	Hard cap on generations per executing branch
Δ_{coord}	10,000	Coordinating stagnation threshold (HoF generations)
Quality function		
w_ε	0.95	Error (accuracy) weight
$w_\mathcal{C}$	0.05	Complexity (simplicity) weight
Pairing		
Elite ratio	10%	Fraction of outcome population classified as elite

- **SRBench** (La Cava et al., 2021): A standardized benchmark suite comprising 252 regression problems from the Penn Machine Learning Benchmarks (PMLB) collection, spanning ground-truth and black-box problems of varying dimensionality and noise levels.
- **Feynman Equations** (Udrescu and Tegmark, 2020): 100 physics equations from the Feynman Lectures, providing ground-truth expressions with known functional forms.
- **Synthetic targets**: A controlled set of expressions with known complexity, designed to probe specific capabilities (e.g., nested compositions, high-arity operators, constants requiring precise optimization).

4.2 Baselines

We compare AHISTER against the following methods:

- **Standard GP**: Single-population genetic programming with the same quality function, population size, and operator weights as AHISTER's executing branches.
- **Island-model GP**: A fixed K-island model with ring-topology migration at intervals of 100 generations, matching AHISTER's stagnation threshold.
- **PySR** (Cranmer, 2023): A state-of-the-art symbolic regression system combining multi-population search with simulated annealing.
- **Operon** (Kommenda et al., 2020): A high-performance GP system with optimized evaluation and offspring selection.

All methods are given equal computational budgets (total fitness evaluations) to ensure fair comparison.

Algorithm 1 AHISTER Coordinating Branch

```
**Require:** Search space $\mathcal{X}$, configuration $(K, N, \Delta_{\text{exec}}, \Delta_{\text{coord}}, \tau_{\text{max}})$
Initialize Hall of Fame $\mathcal{H} \leftarrow \emptyset$, stagnation pool $\mathcal{P}_{\text{stag}} \leftarrow \emptyset$
$\tau_{\text{coord}} \leftarrow 0$, $\tau_{\text{best}} \leftarrow 0$, $g^* \leftarrow$ null
for $i = 1$ to $K$ do
    Launch exploratory executing branch $\mathcal{B}_{\text{exec}}^{(i)}$ with random population from $\mathcal{X}$
end for
while $\tau_{\text{coord}} - \tau_{\text{best}} < \Delta_{\text{coord}}$ and not cancelled do
    // Drain state updates from executing branches
    for each incoming state $s$ from any $\mathcal{B}_{\text{exec}}^{(i)}$ do
        $\mathcal{H} \leftarrow \text{top}_N(\mathcal{H} \cup \mathcal{P}(s),\ \prec_{\mathcal{Q}})$
        $\tau_{\text{coord}} \leftarrow \tau_{\text{coord}} + 1$
        if $\mathcal{H}$ improved then
            $\tau_{\text{best}} \leftarrow \tau_{\text{coord}}$
            Update $g^*$ if strictly better individual found
        end if
    end for
    // Process completed branches
    for each incoming outcome $o$ from any $\mathcal{B}_{\text{exec}}^{(i)}$ do
        $\mathcal{P}_{\text{stag}} \leftarrow \mathcal{P}_{\text{stag}} \cup \{o\}$
        PairAndLaunch($\mathcal{P}_{\text{stag}}, \mathcal{X}$) {Algorithm 2}
    end for
end while
Cancel all active executing branches; drain remaining states into $\mathcal{H}$
return $\mathcal{H}$, $g^*$
```

Table 2: **Symbolic regression benchmark results.** Mean ± standard deviation over 30 independent runs. Best results in bold. *[Results to be completed.]*

Method	R^2 **(SRBench)**	R^2 **(Feynman)**	**Complexity**	**Exact Recovery (%)**	**Time (s)**
Standard GP	—	—	—	—	—
Island-model GP	—	—	—	—	—
PySR	—	—	—	—	—
Operon	—	—	—	—	—
AHISTER	—	—	—	—	—

4.3 Results

4.4 Ablation Study

To isolate the contribution of each AHISTER component, we evaluate the following ablated variants:

- **AHISTER w/o pairing**: Stagnated branches are replaced by new exploratory branches (random initialization) instead of being paired. This removes the genetic material recycling mechanism.
- **AHISTER w/ symmetric offspring**: Paired branches produce two identical successor branches, each seeded from the full combined population (no elite/non-elite split). This tests the value of asymmetric partitioning.
- **AHISTER w/o Hall of Fame**: The coordinating branch does not maintain a Hall of Fame; final results are taken from the best individual across all branch outcomes. This tests the value of the real-time global leaderboard.

Algorithm 2 Maturity-Based Pairing and Asymmetric Offspring

```
Require: Stagnation pool $\mathcal{P}_{\text{stag}}$, search space $\mathcal{X}$
if $|\mathcal{P}_{\text{stag}}| \geq 2$ then
    $(o_a, o_b) \leftarrow \arg\min_{(o_i,o_j)\in\mathcal{P}^2_{\text{stag}},\, i\neq j} |\tau(o_i) - \tau(o_j)|$
    $\mathcal{P}_{\text{stag}} \leftarrow \mathcal{P}_{\text{stag}} \setminus \{o_a, o_b\}$
    $\mathcal{E}_a \leftarrow$ top 10% of $\mathcal{P}(o_a)$;   $\overline{\mathcal{E}}_a \leftarrow \mathcal{P}(o_a) \setminus \mathcal{E}_a$
    $\mathcal{E}_b \leftarrow$ top 10% of $\mathcal{P}(o_b)$;   $\overline{\mathcal{E}}_b \leftarrow \mathcal{P}(o_b) \setminus \mathcal{E}_b$
    Launch tournament branch $\mathcal{B}^{(\mathrm{T})}_{\text{exec}}$ with gene pool $\mathcal{E}_a \cup \mathcal{E}_b$
    Launch advanced branch $\mathcal{B}^{(\mathrm{A})}_{\text{exec}}$ with gene pool $\overline{\mathcal{E}}_a \cup \overline{\mathcal{E}}_b$
else
    Launch exploratory branch with random population from $\mathcal{X}$
end if
```

- **AHISTER w/ random pairing**: Branches are paired randomly rather than by maturity. This tests the value of maturity-based matching.

Ablation results will be reported in a subsequent version.

5 Discussion

Computational overhead. The coordinating branch introduces overhead through Hall of Fame merge operations (sorting and truncation after each state arrival) and the pairing protocol. However, this overhead is minimal relative to the cost of fitness evaluation in the executing branches: the merge is $O(N \log N)$ per state arrival, where N is the population size, and pairing searches over a pool whose size rarely exceeds K. The primary computational cost remains in expression evaluation and refinement within the executing branches.

Sensitivity to K. The number of initial executing branches K controls the degree of parallelism. Larger K provides more diverse starting points but increases the time before the first pairing event (all K branches must stagnate before pairs can form). In practice, $K = 4$ provides a good balance: enough diversity for the initial exploratory phase, with stagnation events occurring frequently enough to keep the pairing mechanism active.

When AHISTER helps most. AHISTER is expected to provide the greatest advantage on problems with rugged, multi-modal fitness landscapes where standard GP populations are likely to converge to different local optima. For smooth, unimodal landscapes, the overhead of the multi-branch architecture may not be justified. The stagnation threshold Δ_{exec} acts as a natural detector: if the landscape is easy, branches will not stagnate within the hard cap and AHISTER degrades gracefully to a parallel random-restart strategy.

6 Conclusion

We have presented AHISTER, a dynamic multi-branch evolutionary framework for symbolic regression that introduces three novel mechanisms: (1) stagnation-driven branch recycling, where branch stagnation is treated as a constructive signal for genetic material exchange; (2) maturity-based pairing, which matches branches at similar evolutionary depths; and (3) asymmetric offspring generation, which simultaneously pursues intensification through elite recombination and diversification through non-elite exploration. A real-time Hall of Fame aggregates the best solutions across all branches, and a dual stagnation clock governs termination at both local and global scales.

The framework is fully implemented and deployed as the core search engine of Ufinq (Universal Function Inquiry), the symbolic AI engine of the Diafunc platform. Future work includes: (1) a comprehensive empirical evaluation on established benchmarks; (2) distributed AHISTER across cluster nodes, leveraging the existing asynchronous communication architecture; (3) adaptive control of the number of executing branches K based on landscape difficulty; and (4) knowledge accumulation across problems, where reusable symbolic subtrees discovered in one regression task are preserved and composed into solutions for subsequent tasks.

References

Edmund K. Burke, Steven Gustafson, and Graham Kendall. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

Michael Kommenda, Bogdan Burlacu, Gabriel Kronberger, and Michael Affenzeller. Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO)*, pages 1460–1467. ACM, 2020.

John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, 1992.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Jörg Lässig and Dirk Sudholt. Adaptive migration strategies for island model genetic algorithms. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1–8. ACM, 2010.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Riccardo Poli, William B. Langdon, and Nicholas F. McPhee. *A Field Guide to Genetic Programming*. Lulu Enterprises, 2008. Available at `http://www.gp-field-guide.org.uk`.

Zbigniew Skolicki and Kenneth De Jong. The influence of migration sizes and intervals on island models. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1295–1302. ACM, 2005.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Darrell Whitley, Soraya Rana, and Robert B. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7(1):33–47, 1999.