

Automated Design Competition at GECCO 2026

Maciej Komosinski Konrad Miazga

www.framsticks.com

Automated
design

Framsticks

Competition

Participants

Results

Automated design

Automated
design

Framsticks

Competition

Participants

Results

To avoid duplication of slides, this section has been skipped.

For this content, see the full 2024 presentation at <https://www.framsticks.com/presentations>

Automated
design

Framsticks

Competition

Participants

Results

Framsticks

Automated
design

Framsticks

Competition

Participants

Results

To avoid duplication of slides, this section has been skipped.

For this content, see the full 2024 presentation at <https://www.framsticks.com/presentations>

Automated
design

Framsticks

Competition

Participants

Results

Competition

Automated
design

Framsticks

Competition

Participants

Results

To avoid duplication of slides, this section has been skipped.

For this content, see the full 2024 presentation at <https://www.framsticks.com/presentations>

Automated
design

Framsticks

Competition

Participants

Results

Participants

Automated
design

Framsticks

Competition

Participants

Results

2024:

- **AdaptMut-Diversity** ← 2024 overall winner and 2025 winner in the **f0** representation, becomes the **f0** baseline for 2026

2024:

- AdaptMut-Diversity ← 2024 overall winner and 2025 winner in the **f0** representation, becomes the **f0** baseline for 2026

2025:

- GP-GOMEA-Island-Mut ← 2025 winner in the **f1** representation, becomes the **f1** baseline for 2026

Submission: AdaptMut-Diversity

2024 overall winner and 2025 winner in the f0 representation, becomes the f0 baseline for 2026

Automated
design

Framsticks

Competition

Participants

Results

This submission uses the **f0** encoding, but other encodings can be used as well.

Two mechanisms introduced aimed at promoting explorative capabilities:

- 1 Adaptive mutation strength – the mutation strength (i.e., the number of mutation operations applied to a genotype) is adjusted during evolution. Starts from mutation strength = 1.0. If the maximal fitness of the population has not changed by more than 1% for the last 4 generations, the mutation strength is multiplied by 1.1. Otherwise, it is multiplied by 0.9. Mutation strength is limited to the range [1, 5]. The motivation was to help the algorithm escape the local optima.
- 2 Introducing random individuals – each mutation operation has a small probability (1%) of introducing a randomly generated individual to the population instead of mutating the current one. This allows to explore the search space more effectively – by introducing new genetic material.

Submission: GP-GOMEA-Island-Mut

2025 winner in the f1 representation, becomes the f1 baseline for 2026

Automated
design

Framsticks

Competition

Participants

Results

- This submission uses the **f1** encoding represented as a Genetic Programming (GP) tree.
- There are 10 islands (populations), 30 individuals each.
- Genotypes in the initial island populations are randomly generated as **f1** genotypes using the Framsticks library function:
`getRandomGenotype(initial_genotype="X", parts_min=20,
parts_max=30, neurons_min=6, neurons_max=8, iter_max=100,
return_even_if_failed=True).`
- Processing each population – see the next slide.

Submission: GP-GOMEA-Island-Mut

2025 winner in the f1 representation, becomes the f1 baseline for 2026

Automated
design

Framsticks

Competition

Participants

Results

Islands (populations) are processed sequentially, each undergoing the following steps:

- Genotypes are converted from **f1** to GP trees (in normal Polish notation, NPN).
- Linkage tree is built using hierarchical clustering.
- Gene-pool Optimal Mixing (GOM) is performed. Each genotype undergoes GOM.
- In the GOM operator, the shorter genotype is padded with special symbols at the end, and mixing is performed. After that, the special symbols are filtered out.
- Genotypes are converted from GP trees (NPN) to **f1** for evaluation.
- When stagnation is detected (best fitness in the population not improving for 2 generations), mutation is performed (on the entire population with probability of 80%). The default, native **f1** mutation operator is used.
- If the similarity of any two populations is above 0.8, one of them is reinitialized randomly. Similarity is estimated using the identity of genotypes and the Jaccard index: $\text{intersection over union} = \frac{|\text{unique genotypes, the same in each population}|}{|\text{unique genotypes in merged both populations}|}$.
- If the uniqueness of the population = $\frac{\text{number of unique genotypes}}{\text{population size}}$ is below 0.2, individuals from other islands migrate to it (25% random originals stay, the remaining 75% is migrated as random genotypes from random islands).

MAGE (Multimodal Adaptive Graph Evolution) is a multi-chromosome extension of Cartesian Genetic Programming (CGP).

The algorithm “BaBy-MAGE” uses CGP with two separate chromosomes: one for the agent Brain (“Ba”) and one for its Body (“By”).

- Evolution uses an island model with 5 islands of 9 individuals each, totaling 45 individuals.
- Islands evolve independently and exchange their best individuals every 20 generations via circular migration.
- Selection combines elitism (3 best preserved) and tournament selection (size 3).
- Adaptive mutation performs 50 mutations instead of 5 when stagnation is detected over 10 generations.

Details are provided on subsequent slides.

CGP encoding:

- Programs are represented as directed acyclic graphs encoded as integer arrays. Each node contains 3 genes: 1 function index and 2 connection indices.
- Two independent chromosomes:
 - Body chromosome: 60 nodes (180 node genes) + 194 output genes = 374 integers.
 - Brain chromosome: 180 node genes + 779 output genes = 959 integers.
- Both chromosomes share 10 fixed inputs:
[0, 1, 2, -1, -2, 0.5, -0.25, 3.14, -3.14, -1.5].
- Uses 9 mathematical functions: addition, subtraction, multiplication, sine, cosine, absolute value, negation, identity, and tangent.
- Execution generates a sequential value list of 70 values (10 inputs + 60 node results), which output genes index into.
- Neutral drift is exploited through inactive nodes that have no phenotypic effect.

Submission: CGP-MAGE-BrainBody-Island

Automated
design

Framsticks

Competition

Participants

Results

Body decoding:

- The 194 body outputs define the agent's physical structure as follows.
- Output 0: Sets the number of parts, clipped to $[3, 20]$.
- Outputs 1 to 20: Set the size of each part, clipped to $[0.1, 1.9]$.
- Outputs 21 to 40: Set the density, clipped to $[0.2, 2.0]$.
- Outputs 41 to 60: Set the friction coefficient, clipped to $[0.0, 1.0]$.
- Outputs 61 to 79: Specify the tree topology parent index, clipped to $[0, \text{current_part_index} - 1]$.
- Outputs 80 to 174: Set 5 geometric parameters per joint (dx, dy, dz and rx, rz). Displacements are normalized if max joint length $\sqrt{dx^2 + dy^2 + dz^2}$ exceeds 1.9, and rotations span $[-\pi, \pi]$.
- Outputs 175 to 193: Set joint stiffness, clipped to $[0.1, 1.0]$.

Brain decoding:

- The 779 brain outputs encode the neural controller.
- 3 outputs per joint (for up to 19 joints) determine: neuron presence (active if ≥ 0.3), neuron type, and its parameter.
- 7 available neuron types, including sinusoidal generator (Sin), gyroscope sensor (G), and muscle effectors (I and @).
- The remaining $19 \times 19 \times 2 = 722$ outputs encode connections between neurons: 2 values determine connection existence (present if ≥ 0.3) and weight (clipped to $[-5.0, 5.0]$).

Island model:

- 5 independent islands.
- A circular migration topology: the best individual from island i is copied into island $i + 1 \pmod{5}$, replacing its worst individual.

Selection:

- 5 offspring are generated per parent per generation.
- All parents and offspring are merged into a single selection pool.
- Elitism is applied by preserving the top 3 individuals.
- The remaining population slots are filled using a repeated tournament selection of size 3 from the rest of the pool.

Adaptive mutation:

- Offspring are produced by mutating copies of parent chromosomes.
- A single mutation selects a random gene index:
 - If it is a node gene (index < 180), it modifies either the function index or one connection index (respecting acyclicity).
 - If it is an output gene, it is replaced by a random index in $[0, 69]$.
- Normally, 5 mutations are applied per offspring.
- Stagnation is detected if the best fitness does not improve by more than 10^{-3} over the last 10 generations.
- During stagnation, the number of mutations per offspring adaptively increases from 5 to 50 to force exploration.

Submission: AdaptMut-DynMut-Restarts

Automated
design

Framsticks

Competition

Participants

Results

The algorithm extends the AdaptMut submission from 2024. Sources are [here](#).

This submission uses the **f0** encoding, but other encodings can be used as well.

- popsize=50, pmut=0.8, pxov=0.2.
- The initial population is randomly generated.
- Infeasible genotypes (detected with `framsLib.isValidCreature(genotype)`) are not evaluated, so they don't count towards the evaluation counter. This saves on average 300 evaluations per run.
- Triggered hypermutation (**AdaptMut**): If stagnation is detected (no fitness increase 1% in the last 5 generations), the number of applied mutations is increased (max range 1-5 mutations).
- **Restarts** mechanism:
 - Activated when no improvement for 10 generations.
 - New population = [25% clone best and mutate 0-4 times] + [75% randomly generated individuals].

Submission: AdaptMut-DynMut-Restarts

Automated
design

Framsticks

Competition

Participants

Results

Dynamic mutation probabilities (**DynMut**): After each generation, adjust the probabilities of each type of mutation:

- The “Mutation Result Cache” stores the fitness increase/decrease caused by each mutation type in a rolling window (decay = 0.985).
 - The mutation types are the same as the Framsticks mutation parameters (e.g., `f0_n_new` or `f1_smX`).
 - Rolling window decay was chosen so a 300 fitness decrease is ‘forgotten’ after ≈ 20 generations ≈ 1000 evaluations: $-300 \cdot 0.985^{987 \text{ generations}} < 0.0001$.
 - ‘crossover’ and ‘invalid’ operations are also recorded, but their probabilities are not altered during the run.
- The mutation probabilities are recomputed after each generation.
- The mutation probabilities are the same for all individuals in a generation (i.e., no per-individual mutation probabilities).
- The probability of each mutation type is, surprisingly, proportional to the fitness *decrease* it causes. This strategy outperformed several other evaluated approaches.
- To prevent the extinction of any mutation type, a value is added to all fitness decreases before they are normalized to sum to 1, yielding the new mutation probabilities.
- Mutation probabilities which are initially 0 (e.g., `f0_p_color`) are kept at 0.

Automated
design

Framsticks

Competition

Participants

Results

Results

Best solutions

Automated
design

Framsticks

Competition

Participants

Results

Distance:

<https://www.framsticks.com/files/varia/automated-design-competition-2026-best-distance.mp4>

Tall runners:

<https://www.framsticks.com/files/varia/automated-design-competition-2026-best-tall-runners.mp4>

Averaged normalized performance on individual benchmark tasks

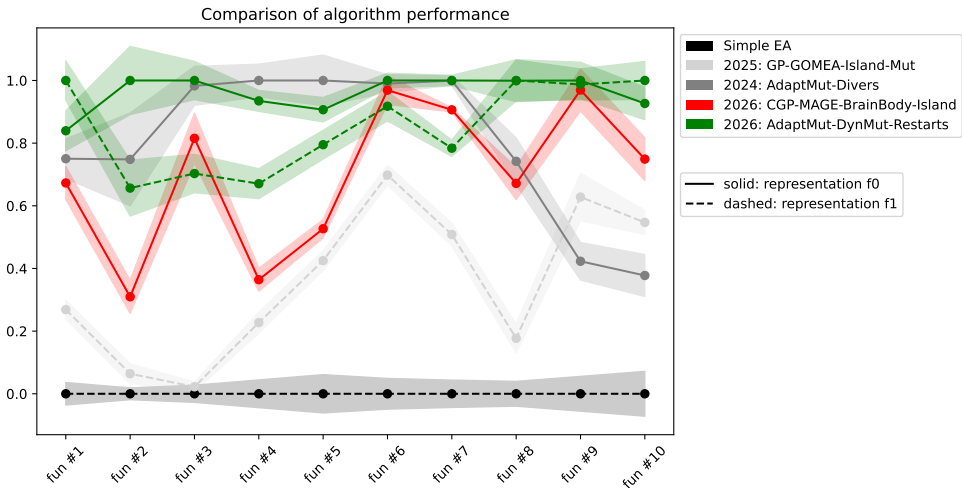
Automated
design

Framsticks

Competition

Participants

Results



Aggregated performance

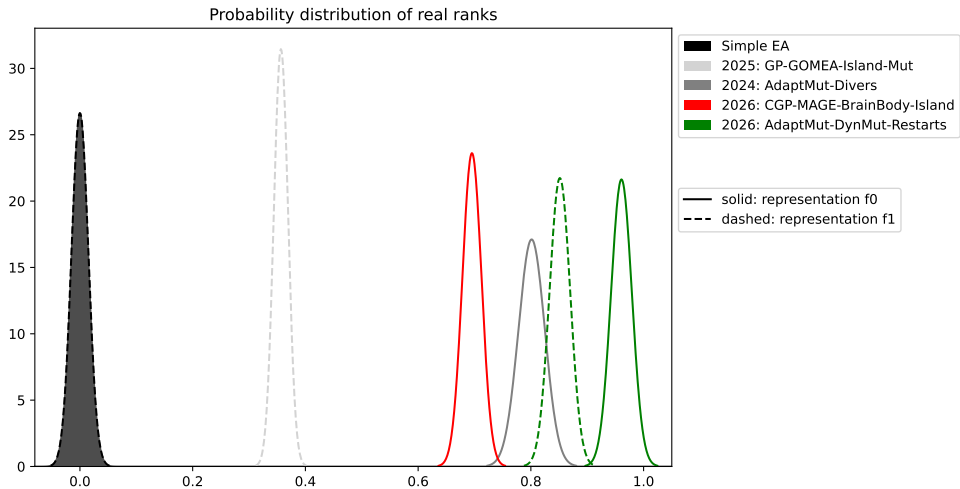
Automated
design

Framsticks

Competition

Participants

Results



Aggregated performance

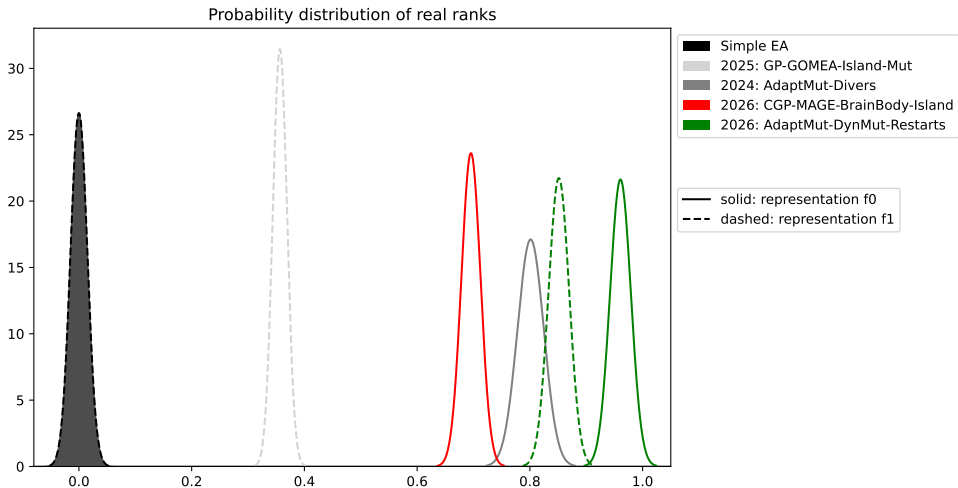
Automated
design

Framsticks

Competition

Participants

Results



1. **AdaptMut-DynMut-Restarts** surpasses all prior submissions across both the **f0** and **f1** genetic representations.
2. **CGP-MAGE-BrainBody-Island** outperforms the **f1** baseline, but does not outperform the **f0** baseline.