MultiGA: Leveraging Multi-Source Seeding in Genetic Algorithms

Isabelle Diana May-Xin Ng*[1,2], Tharindu Cyril Weerasooriya[1], Haitao Zhu[1], Wei Wei[1]

[1]Center for Advanced AI, Accenture, [2]University of California, Berkeley

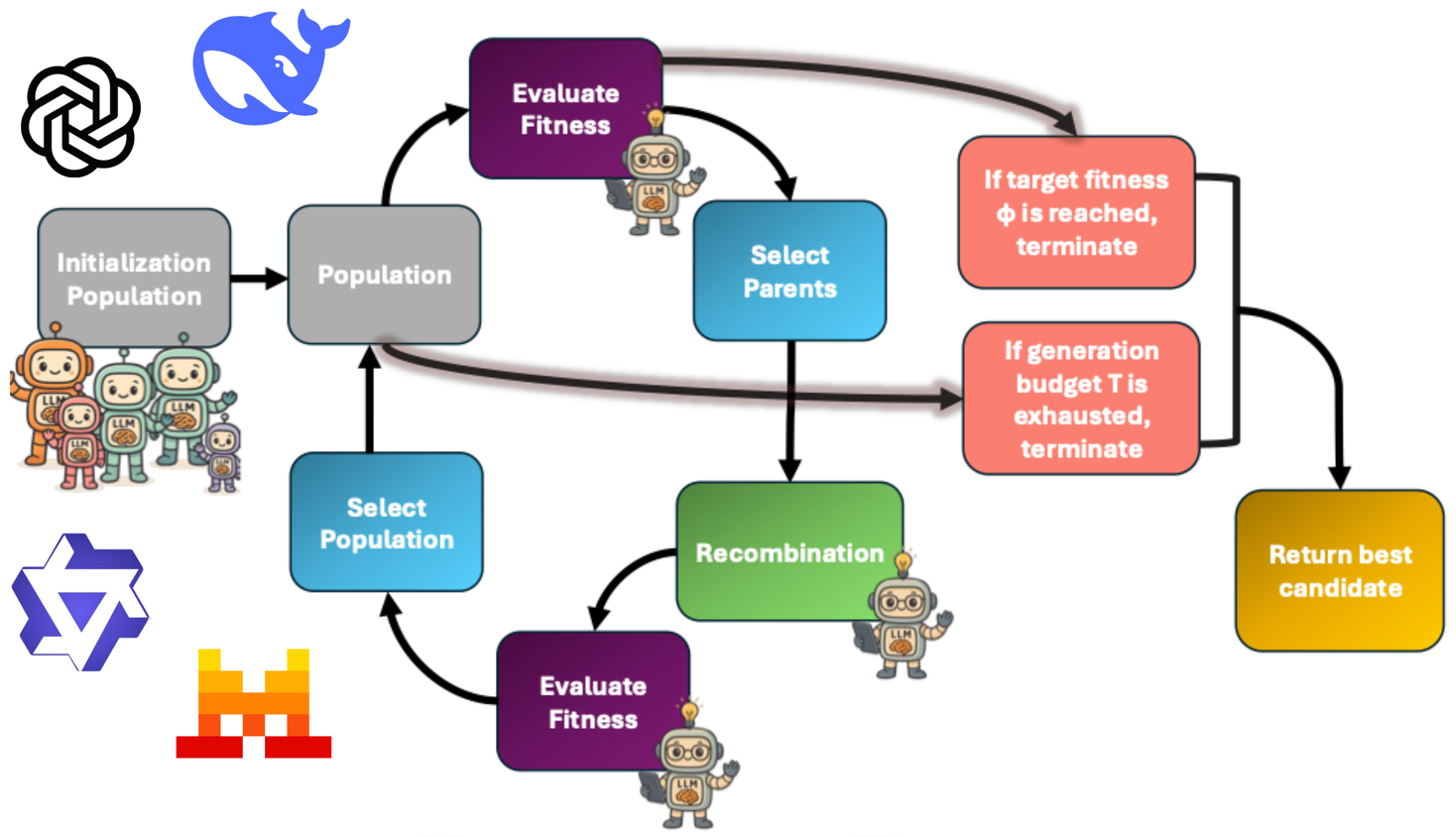
*Presenting Author, isabelle-ng.com and email at isabelle.ng@berkeley.edu

1

Can heterogeneous LLMs, each with different strengths, jointly improve robustness on code generation and complex reasoning tasks?

2

Idea: Initialize the genetic algorithm cycle with candidates produced by multiple LLMs, creating a diverse starting population.



Overview of the MultiGA framework. Populations are initialized with multiple LLMs, while an independent LLM E handles fitness evaluation (scoring candidates) and recombination (combining two parent solutions). The process terminates once target fitness ϕ or maximum number T generations is reached. Then, the top candidate solution is returned.

Algorithm 1: MultiGA: Multi–Source Genetic Algorithm

Input: Task spec Q; Generator LLMs $\mathcal{G} = \{g_1, \ldots, g_m\}$; Evaluator/crossover LLM E with fitness $f : \mathcal{X} \to [0,1]$; Population size n; Top-k; Threshold $\tau \in [0,1]$; Max generations T; Target fitness $\phi \in [0,1]$.

Output: Best solution $\hat{x} \in \mathcal{X}$.

$P_0 \leftarrow$ initialize_population$(\mathcal{G}, Q)$;
$t \leftarrow 0$;
while $t < T - 1$ **do**
  $scores \leftarrow$ evaluate_fitness(P_t, f) ;
  if $\max(scores) \geq \phi$ **then**
    break
  $S_t \leftarrow$ select_parents$(P_t, scores, k)$;
  $C_t \leftarrow$ recombination(S_t, P_t, E, Q) *# Create next generation (children solutions)* ;
  $P_{t+1} \leftarrow$ select_population(P_t, C_t, τ) *# Retire unfit candidates* ;
  $t \leftarrow t + 1$;
$scores \leftarrow$ evaluate_fitness(P_t, f) ;
$\hat{x} \leftarrow \arg\max_{x \in P_t} f(x)$;
return $\hat{x}$;

Gen: 1, Model: gpt-4o, Score: 0.95

```
SELECT publisher.publisher_name FROM
superhero INNER JOIN hero_attribute ON
superhero.id = hero_attribute.hero_id INNER
JOIN attribute ON hero_attribute.attribute_id =
attribute.id INNER JOIN publisher ON
superhero.publisher_id = publisher.id WHERE
attribute.attribute_name = 'Speed' ORDER BY
hero_attribute.attribute_value ASC LIMIT 1
```

Gen: 1, Model: deepseek-r1-distill-qwen-32b, Score: 0.1

```
SELECT `publisher`.`publisher_name`,
MIN(`hero_attribute`.`attribute_value`) AS
`min_speed` FROM `superhero` INNER JOIN
`publisher` ON `superhero`.`publisher_id` =
`publisher`.`id` INNER JOIN `hero_attribute` ON
`superhero`.`id` = `hero_attribute`.`hero_id` WHERE
`hero_attribute`.`attribute_name` = 'Speed' GROUP BY
`publisher`.`publisher_name`, `superhero`.`id`
ORDER BY `min_speed` ASC LIMIT 1
```

Gen: 2, Model: gpt-4o + r1, Score: 1.0

```
SELECT publisher.publisher_name FROM superhero INNER JOIN
hero_attribute ON superhero.id = hero_attribute.hero_id INNER JOIN
attribute ON hero_attribute.attribute_id = attribute.id INNER JOIN
publisher ON superhero.publisher_id = publisher.id WHERE
attribute.attribute_name = 'Speed' GROUP BY publisher.publisher_name
ORDER BY MIN(hero_attribute.attribute_value) ASC LIMIT 1
```

3

This diversity gives crossover and mutation far more variation to work with, enabling hybrid reasoning and code patterns that no single model would generate.

Example of recombination on a text-to-SQL task ("Which publisher published the slowest superhero?"). Parent 1 (gpt-4o, score 0.95) was paired with a randomly selected parent (deepseek-r1, score 0.1). The resulting child achieved a perfect score of 1.0 by preserving gpt-4o's overall structure while incorporating the MIN aggregation from r1.

Dataset	Test Rows	Total Items	Label Choices
$\mathcal{D}_{\text{SQL}}$ (BIRD Mini-Dev)	100	500	∞
$\mathcal{D}_{\text{NATPLAN}}$ (Meeting Planning)	100	1000	∞
$\mathcal{D}_{\text{GPQA}}$ (Grad-Level Science Questions)	198	198	4
$\mathcal{D}_{\text{BBQ}}$ (BBQ Bias Data)	104	6879	3

Datasets used in our experiments, with test set size, total available items, and number of label choices.

4

MultiGA produces accuracy that consistently approaches the performance of the strongest single model baseline and reduces reliance on any single model through iterative refinement and recombination.

Seed Model	$\mathcal{D}_{\text{SQL}}$	$\mathcal{D}_{\text{NATPLAN}}$	$\mathcal{D}_{\text{GPQA}}$	$\mathcal{D}_{\text{BBQ}}$
gpt-4o	**0.56**	0.39	**0.44**	**1.00**
qwen2.5-coder-32b-instruct	0.45	0.28	0.30	0.87
deepseek-r1-distill-qwen-32b	0.46	0.23	0.41	0.97
openai/gpt-oss-20b	0.48	0.19	0.41	0.93
mistral-7b-instruct-v0.3	0.25	0.10	0.22	0.87
$\mathcal{G}$ (MultiGA)	0.55	**0.40**	0.40	**1.00**

Accuracy across tasks when seeding with each individual model versus seeding with all five simultaneously (G).

Thanks to Accenture for providing mentorship, compute resources, and a supportive research environment that enabled this project. We are especially grateful to the Center for Advanced AI for guidance throughout the development of this work.

I am applying to CS PhD programs for the Fall 2026 cycle & seeking potential advisors!