Improving Monte Carlo Tree Search for Symbolic Regression

Zhengyao Huang[1], Daniel Zhengyu Huang[1], Tiannan Xiao[2], Dina Ma[2], Zhenyu Ming[2], Hao Shi[3], Yuanhui Wen[2]

[1]Center for Machine Learning Research, Peking University
[2]Huawei Technologies Ltd., Beijing
[3]Department of Mathematical Sciences, Tsinghua University

https://github.com/PKU-CMEGroup/MCTS-4-SR

NeurIPS 2025

Motivation & Contributions

- **Symbolic Regression (SR)** aims to recover interpretable mathematical expressions.
- Traditional GP or RL methods often suffer from weak exploration and unstable convergence.
- **Monte Carlo Tree Search (MCTS)** naturally explores expression trees but standard UCB focuses on **average reward**.

Our Key Ideas:

1. Replace UCB with **UCB-Extreme**, targeting **maximum** (not mean) reward.
2. Introduce **State-Jumping**, combined with **Bidirectional Propagation** to share high-reward structures globally and allow non-local transitions via mutation and crossover.

Why UCB-Extreme?

Standard UCB:

$$a = \arg\max_a \left[Q(s,a) + c\sqrt{\tfrac{2\ln T_s}{T_{s,a}}} \right]$$

- Optimizes the *expected mean reward*.
- Works well for average-regret minimization.
- Fails in SR where we only care about the **best expression found**.

UCB-Extreme:

$$I_{T+1} = \arg\max_k \left[\hat{Q}_{k,T_k} + 2c\left(\frac{\ln T}{T_{k,T}}\right)^{\gamma} \right]$$

- $\hat{Q}_{k,T_k}$: best reward from arm k so far.
- Theoretical guarantees under heavy-tailed rewards:

$$R(T) = O\left(\frac{\ln T}{T^{1+1/a_1}}\right)$$

$\rightarrow$ **Shifts from average performance to best-arm discovery, better suited for SR.**

Why State-Jumping?

- In symbolic regression, valid expressions are sparse in a huge combinatorial space.
- Standard MCTS expands nodes locally $\rightarrow$ easily trapped in subtrees.
- We enable **non-local exploration** via:
  - Each node maintains a **priority queue** of the top-N high-reward trajectories encountered during the search.
  - Before selection, the algorithm may apply:
    - **Mutation:** randomly alters sub-expressions from stored trajectories.
    - **Crossover:** recombines sub-expressions between high-reward trajectories.
- Combined with **bidirectional propagation**, the best structures are shared upward and downward:

$$\hat{V}(v) = \max_{v' \in \text{children}(v)} \hat{V}(v')$$

$\rightarrow$ **Encourages faster discovery and reuse of globally optimal substructures.**

Overview of the Improved MCTS Framework

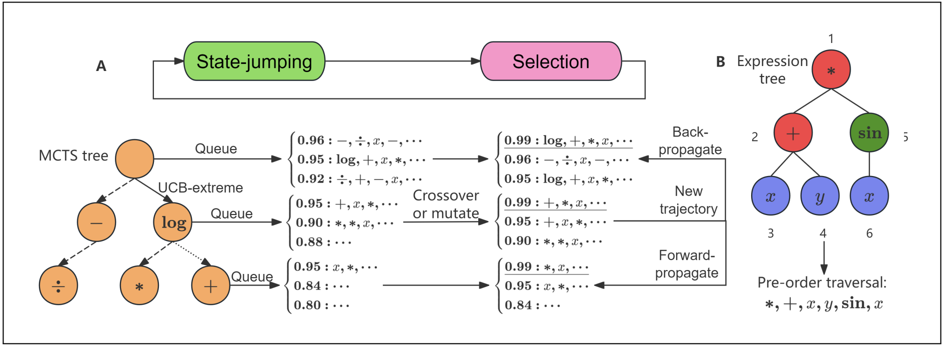

Figure: Overview of UCB-Extreme and State-Jumping within the MCTS workflow.

Experimental Setup

- **Benchmarks:** Nguyen, NguyenC, Jin, Livermore, and SRBench Black-box.
- **Metrics:**
  - Basic: recovery rate over 100 runs (2M evals/run).
  - SRBench Black-box: median test R^2 and model size (500k evals or 48h).
- **Baselines:** DSR, GEGL, NGGP, PySR, Operon, GP-GOMEA and several standard machine learning regressors.

Results on Ground-truth Benchmarks

Benchmark	Ours	DSR	GEGL	NGGP	PySR
Nguyen	**93.3**	83.6	86.0	92.3	74.4
NguyenC	**100.0**	100.0	100.0	100.0	65.4
Jin	**100.0**	70.3	95.7	100.0	72.2
Livermore	**71.4**	30.4	56.4	71.1	46.1

- Outperforms or matches all baselines across datasets.

Results on SRBench Black-box

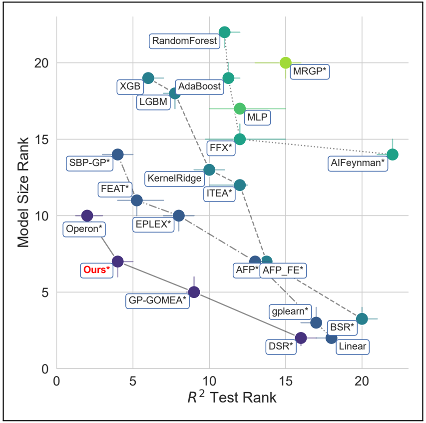

Figure: Pareto frontier of model size vs. median test R^2.

- Competitive accuracy with more compact expressions.

Conclusion

- Proposed an improved MCTS framework for Symbolic Regression:
  1. **Extreme-bandit allocation** for best-arm discovery.
  2. **Evolution-inspired state-jumping** combined with **Bidirectional Propagation** for non-local exploration and efficient information sharing.
- Demonstrated theoretical optimality and empirical competitiveness.

Code: `https://github.com/PKU-CMEGroup/MCTS-4-SR`

Thank you!

Questions?