

Escaping Local Optima in Inductive Program Synthesis

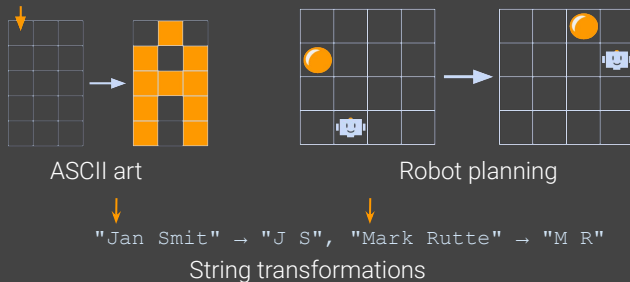
Nadia Matulewicz
n.i.matulewicz@student.tudelft.nl

Supervisor: Sebastijan Dumančić

1. Background

Inductive Program Synthesis (IPS):

- Automation of finding program based on I/O examples
- Domains:



IPS system:

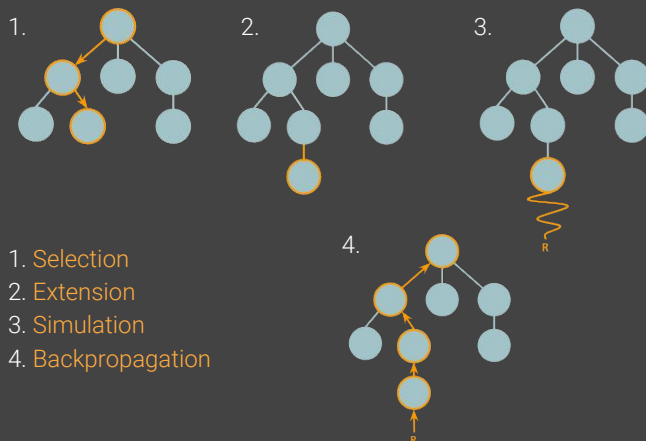
- Input: Input/output examples, tokens, loss function
- Output: Program (sequence of tokens)

Brute

- Shows potential of **heuristic-based loss functions**
- Problem: Fails when **local optima** occur.

Monte Carlo Tree Search (MCTS)

- Balances between **exploration** and **exploitation**.



2. Research Question

Can a new IPS system that uses MCTS guided by a heuristic-based loss function...

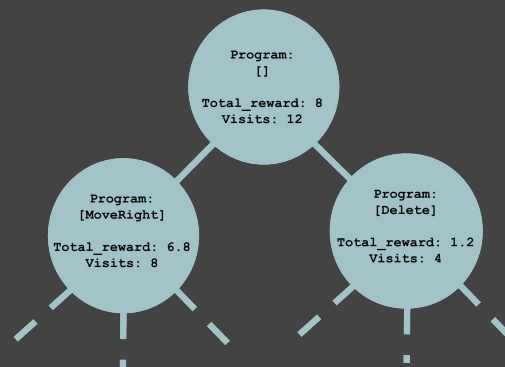
1. escape local optima
2. outperform Brute and other IPS systems?

3. Method

- Literature research
- Design and implement MUTE
- Compare performance of MUTE with other IPL systems

4. MUTE

- New IPS system that uses MCTS guided by a heuristic-based loss function
- Each node represents addition of a token



- Reward is computed using heuristic-based loss function
- Formula for selecting child node:

$$UCT = \underbrace{\frac{\text{total_reward}}{\text{visits}}}_{\text{Exploitation component}} + 2 \sqrt{\frac{\ln(\text{visits}_{\text{parent}})}{\text{visits}}}_{\text{Exploration component}}$$

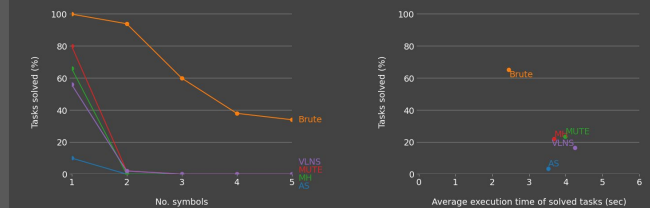
- First major improvements after **removal of similar programs** and **removal of low potential tokens**

5. Results

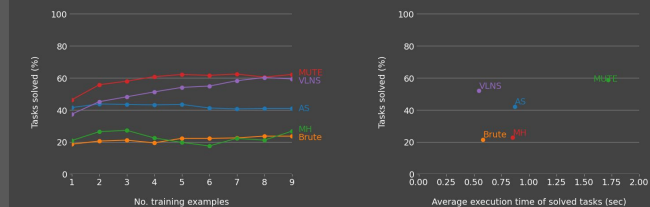
Robot planning:

- 100% solving rate for all IPS systems

ASCII art:



String transformations:



Manual analysis

- MUTE is able to escape local optima. E.g.:

Input examples:

"22022002" → "2002"
"1252010" → "2010"

Best found programs:

Iteration 18566: [Drop, Drop, Drop, Drop]
Iteration 41288: [While(NotAtEnd [MoveRight]), MoveLeft, MoveLeft, ..., Drop]

5. Conclusion

1. Yes, MUTE escapes local optima
2. First indication MUTE can outperform other IPS systems

Important note: Removal of similar programs and tokens without potential is essential

References

[1] Cropper, Andrew, and Sebastijan Dumančić. "Learning large logic programs by going beyond entailment." arXiv preprint arXiv:2004.09855 (2020).