How to train your dragon:

On the application of the Metropolis-Hastings method for Program Synthesis

Author: Bo Hofstede

Supervisor: S. Dumančić

TU Delft

Program Synthesis 1

Goal: Create an algorithm to generate other programs

How?
1. Establish Intent
2. Efficiently search through all possible program

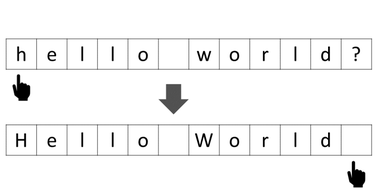
Intent: User Input-Output Examples, used to evaluate program's 'cost'

Domain Specific Language (DSL): a smaller version of a programming language, consists of tokens:
- Transition, affect program state (e.g. MoveRight)
- Boolean, based on state (e.g. IsUpperCase)
- Control, affect program flow (e.g. LoopWhile)

Domains: Robot, Pixel, String

Metrics: Success Rate, Execution Time, Program Length

Example: String



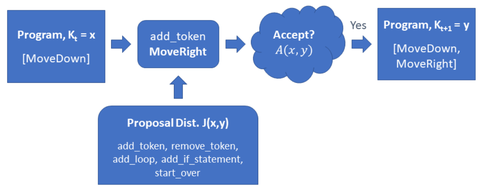
Cost Function: Levenshtein (Min. no. single-char edits)

Program: [MakeUpperCase, LoopWhile(IsLetter, MoveRight), MoveRight, MakeUpperCase, While(IsLetter, MoveRight), Drop]

Metropolis-Hastings 2

What?
Markov Chain Monte Carlo method.

Goal: approximate a distribution that can be evaluated but not sampled from.



Idea: mutate a program, accept with probability:

$A(x', x) = \min(1, \frac{\pi(y)}{\pi(y)} \frac{J(y,x)}{J(x,y)})$, where $\pi(x) = e^{-\alpha * Cost(x)}$

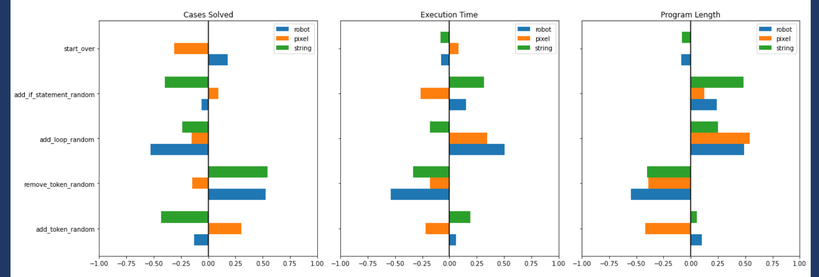
Research Question 3

Is it possible to improve the performance of a stochastic search Metropolis-Hasting program synthesis algorithm by changing the configuration of components?

Results: Mutations 4a

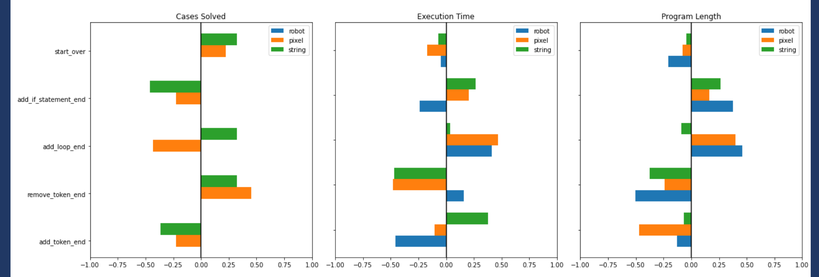
Possible mutations: add_token, remove_token, add_loop, add_if_statement, start_over

Randomized Locality Mutations (Kendall Correlation)**:**



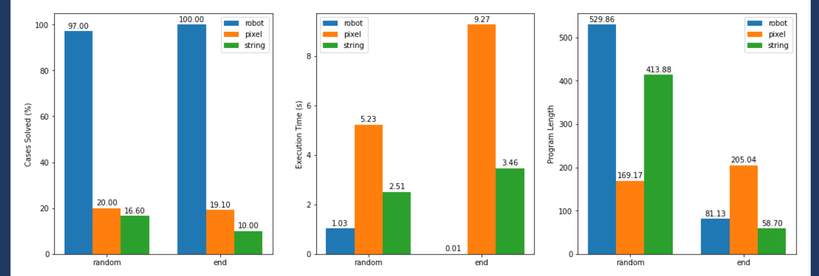
→ Randomized locality overfits on domains

End locality Mutations (Kendall Correlation)**:**



→ Fixed-at-end locality can be optimized
→ DSL Matters

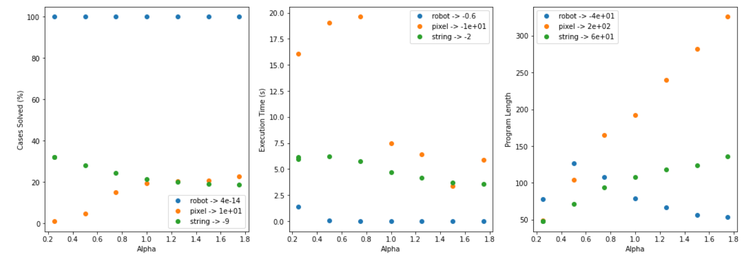
Locality Comparison:



→ No difference in performance

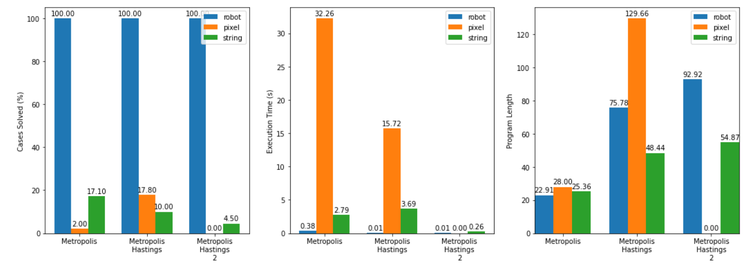
Results: Acceptance Function 4b

Cost Normalization (α):



→ Domain dependent, possibly relating to the program's likelihood to get stuck in certain local optima's

Metropolis-vs-Hastings:



→ It is very difficult to define explicit inverses
→ Underestimate transition probabilities

Conclusion 5

Improvements can be made, however they fail to perform better than more random but domain specific approaches.

Responsible Research 6

Credit has to be given to the people that created on the codebase prior to this paper:

1. Victor van Wieringen, with contributions from S Dumančić (mentor), and C B Poulsen (mentor). Comparative ´ analysis of the metropolis-hastings algorithm as applied to the domain of program synthesis, 1 2022.
2. Andrew Cropper and Sebastijan Dumančić . Learning large logic programs by going beyond entailment. 2020.