VanillaGP: Genetic Algorithm for Inductive Program Synthesis

1. Background

IPS is the problem of generating programs from input and output samples

> The space of programs is too large to exhaustively enumberate Recent best-first search algorithm **Brute [1]** navigates the search space with a certain heuristic

- Brute performs well but struggles to escape locally optimal solutions
- > Genetic Algorithms introduce more exploratory behavior

4. The Algorithm



The selection algorithm is **SUS** (Stochastic Universal Sampling)

n-point crossover (n=2):

P_1	f_0	f_1	f_2	f_3	f_4	f_5	f_6
P_2	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}



References:

[1] Andrew Cropper and Sebastijan Dumancic. Learning large logic programs by going beyond entailment. CoRR, abs/2004.09855, 2020. [2] Thomas Helmuth, Nicholas Freitag McPhee, and Lee Spector. Program synthesis using uniform mutation by addition and deletion. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, page 1127â1134, New York, NY, USA, 2018. Association for Computing Machinery.



- > Can VanillaGP solve more complex tasks by exploring the search space stochastically?
- > Can VanillaGP produce generalizable solutions? How does the
- training performance reflect on the test performance?

Supervisor: Sebastijan Dumancic | Research Project CSE3000 | Farhad Azimzade F.Azimzade@student.tudelft.nl

3. Experiments

- > **Pixel Domain**: convert empty grid canvas into an ASCIcharacter image
- > **Robot Domain**: guide a robot across a grid to place a ball on the correct cell
- > **String Domain**: find and apply the rule that is behind a mapping of one string to another

5. Results