# Enhancing Evolutionary Computation through Phylogenetic Analysis

Chenfei Peng
peng0368@morris.umn.edu

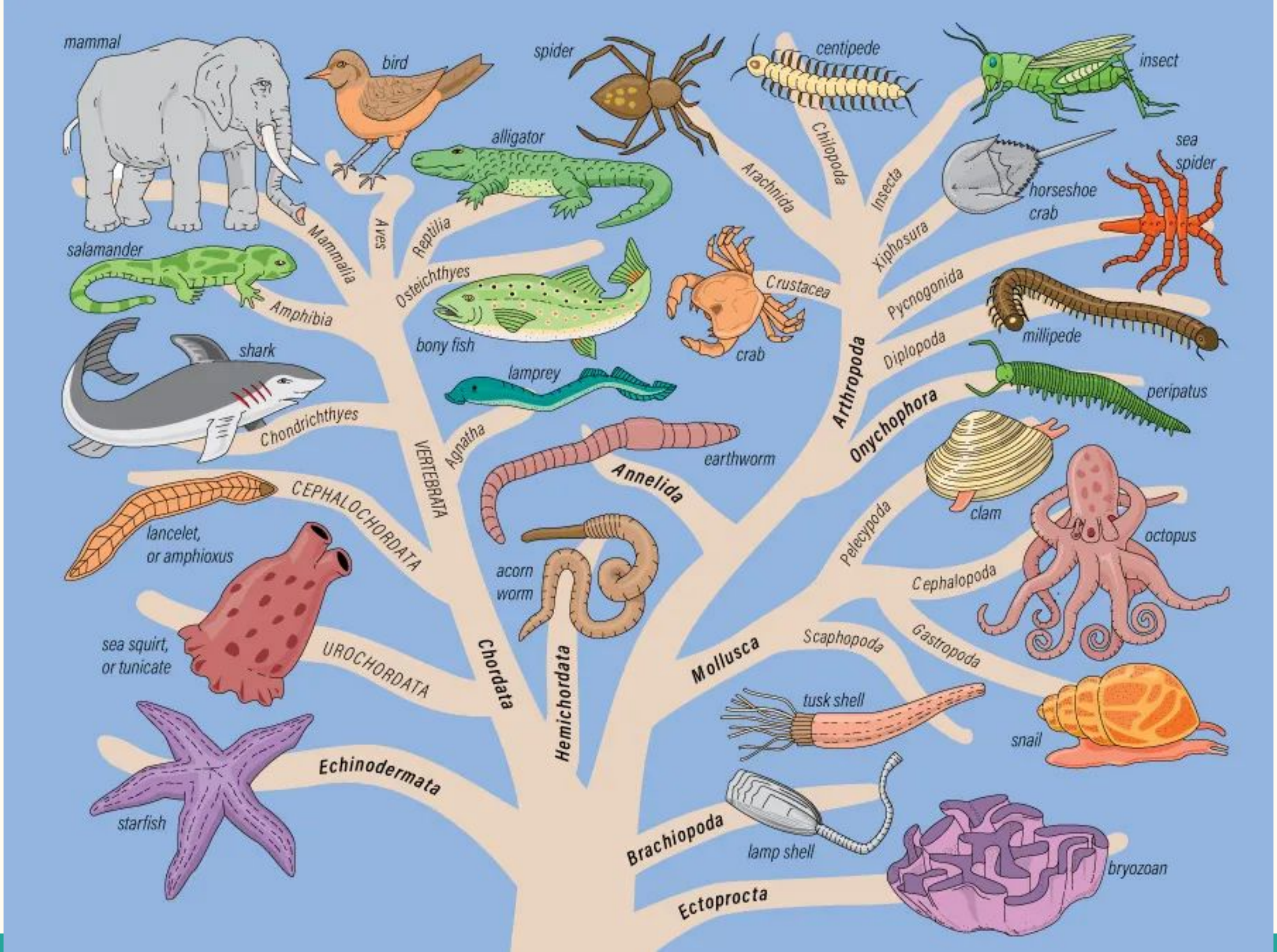
# Outline

0. Motivation
1. Introduction
2. Background
3. Phylogeny-informed fitness estimation
4. Genetic programming experiments
5. Results
6. Conclusion

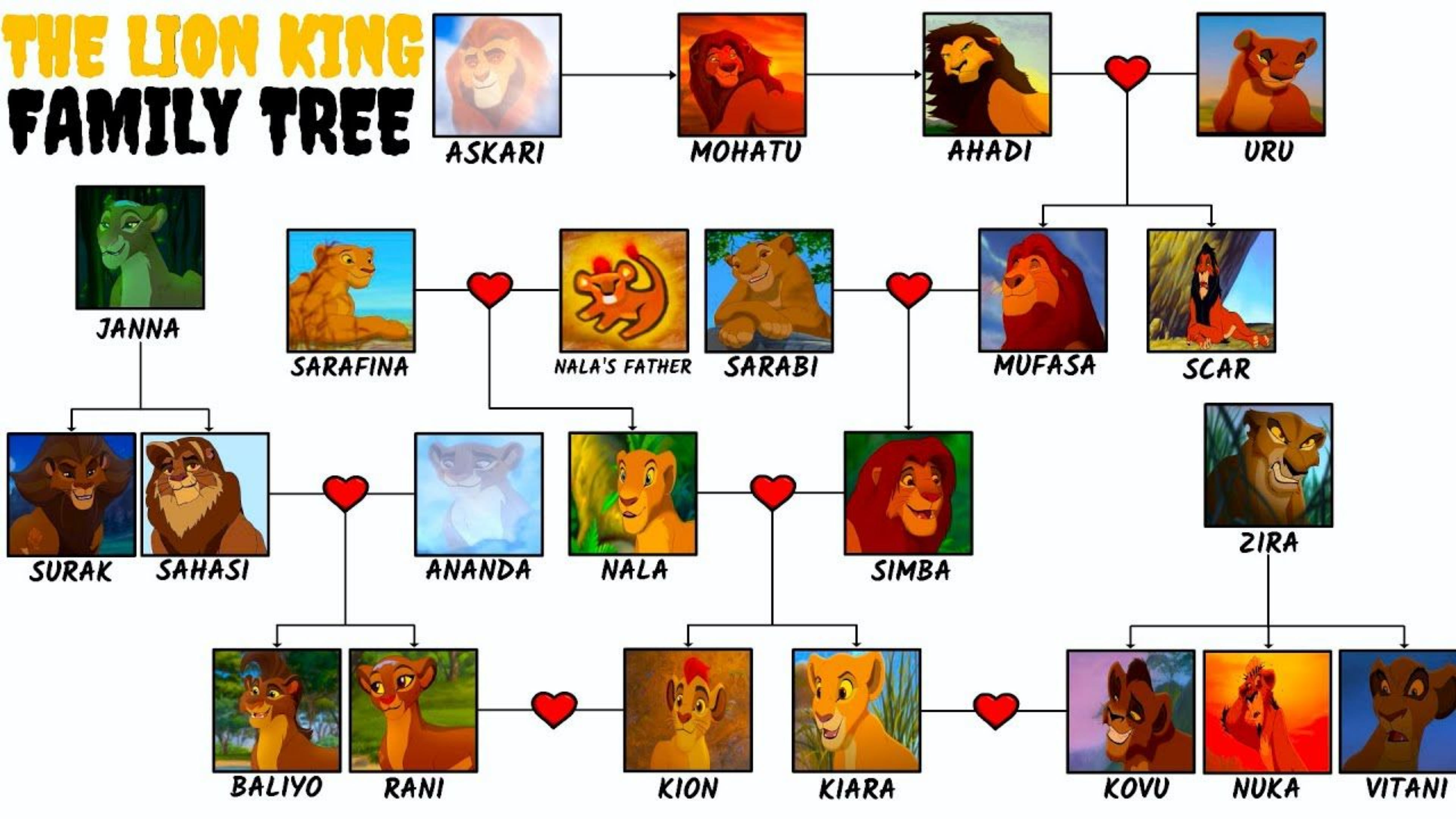
**Phylogeny-informed fitness estimation**

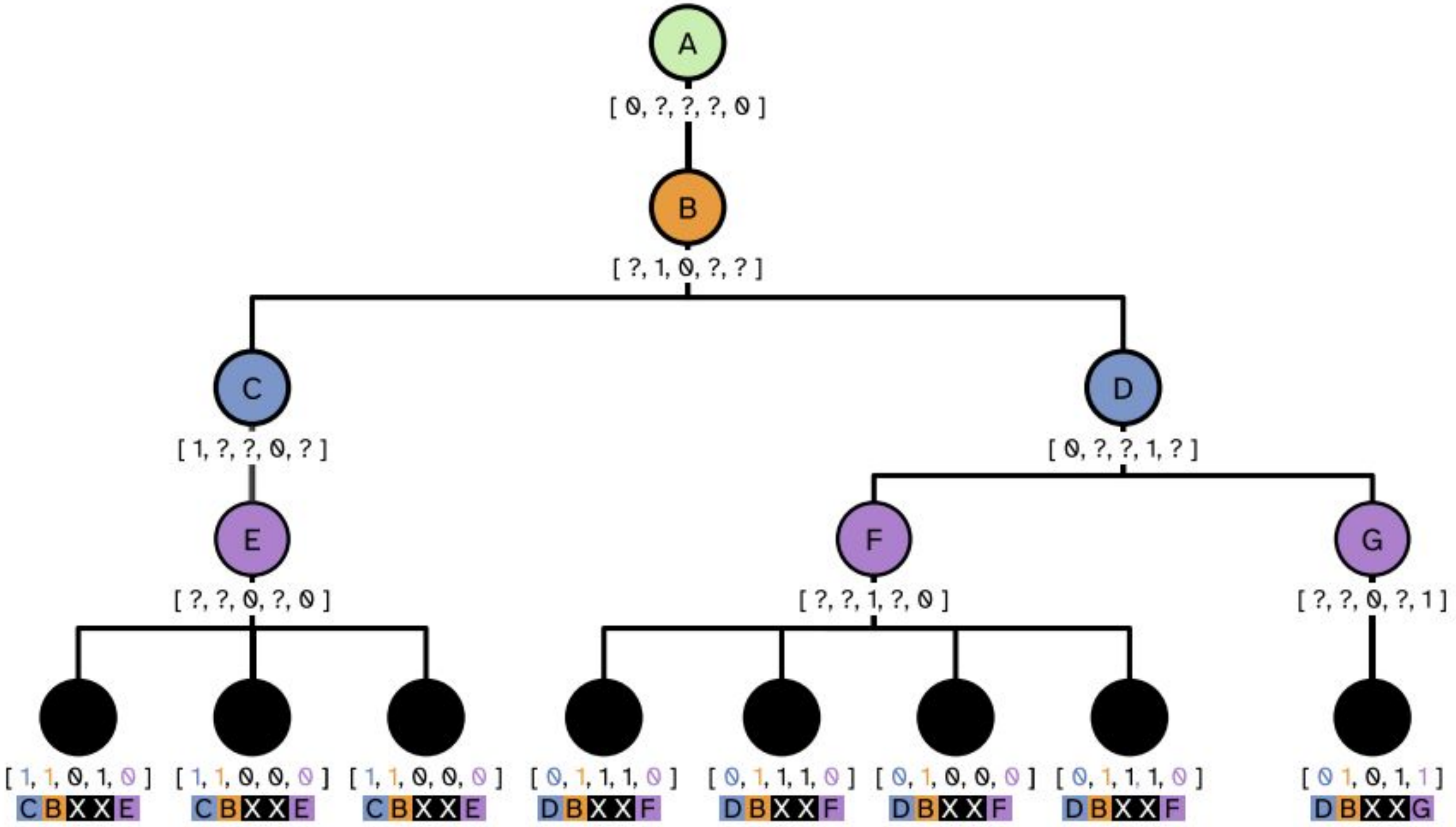
Alexander Lalejini[1], Matthew Andres Moreno[2], Jose Guadalupe Hernandez[3], and Emily Dolson[3]

[1] Grand Valley State University
[2] University of Michigan
[3] Michigan State University



mammal
bird
spider
centipede
insect
alligator
sea spider
horseshoe crab
salamander
Mammalia
Aves
Reptilia
Arachnida
Chilopoda
Insecta
Xiphosura
Osteichthyes
Crustacea
Pycnogonida
Amphibia
bony fish
crab
millipede
Diplopoda
shark
lamprey
peripatus
Arthropoda
Onychophora
Chondrichthyes
Agnatha
earthworm
VERTEBRATA
Annelida
CEPHALOCHORDATA
clam
lancelet,
or amphioxus
octopus
Pelecypoda
acorn
worm
Cephalopoda
sea squirt,
or tunicate
UROCHORDATA
Chordata
Hemichordata
Mollusca
Scaphopoda
Gastropoda
tusk shell
snail
Echinodermata
starfish
Brachiopoda
lamp shell
bryozoan
Ectoprocta

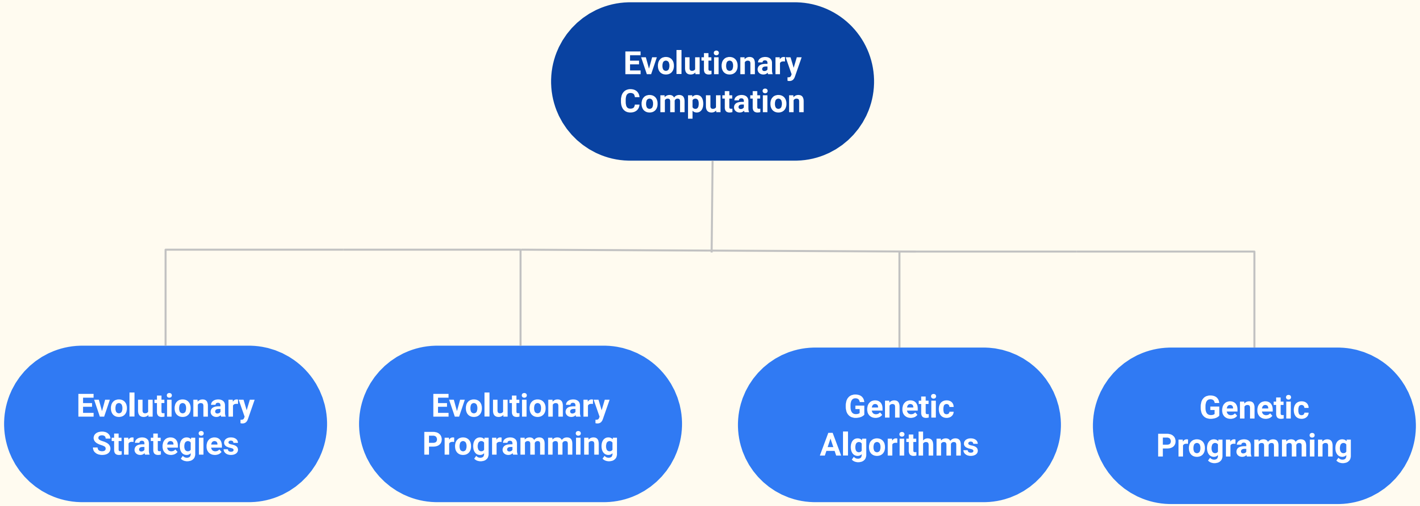

THE LION KING
FAMILY TREE
ASKARI
MOHATU
AHADI
URU
JANNA
SARAFINA
NALA'S FATHER
SARABI
MUFASA
SCAR
ZIRA
SURAK
SAHASI
ANANDA
NALA
SIMBA
BALIYO
RANI
KION
KIARA
KOVU
NUKA
VITANI


A
[ 0, ?, ?, ?, 0 ]
B
[ ?, 1, 0, ?, ? ]
C
[ 1, ?, ?, 0, ? ]
D
[ 0, ?, ?, 1, ? ]
E
[ ?, ?, 0, ?, 0 ]
F
[ ?, ?, 1, ?, 0 ]
G
[ ?, ?, 0, ?, 1 ]
[ 1, 1, 0, 1, 0 ]
C B X X E
[ 1, 1, 0, 0, 0 ]
C B X X E
[ 1, 1, 0, 0, 0 ]
C B X X E
[ 0, 1, 1, 1, 0 ]
D B X X F
[ 0, 1, 1, 1, 0 ]
D B X X F
[ 0, 1, 0, 0, 0 ]
D B X X F
[ 0, 1, 1, 1, 0 ]
D B X X F
[ 0 1, 0, 1, 1 ]
D B X X G

# Introduction

# Introduction

- Evolutionary Computation (EC)
- Genetic Programming (GP)



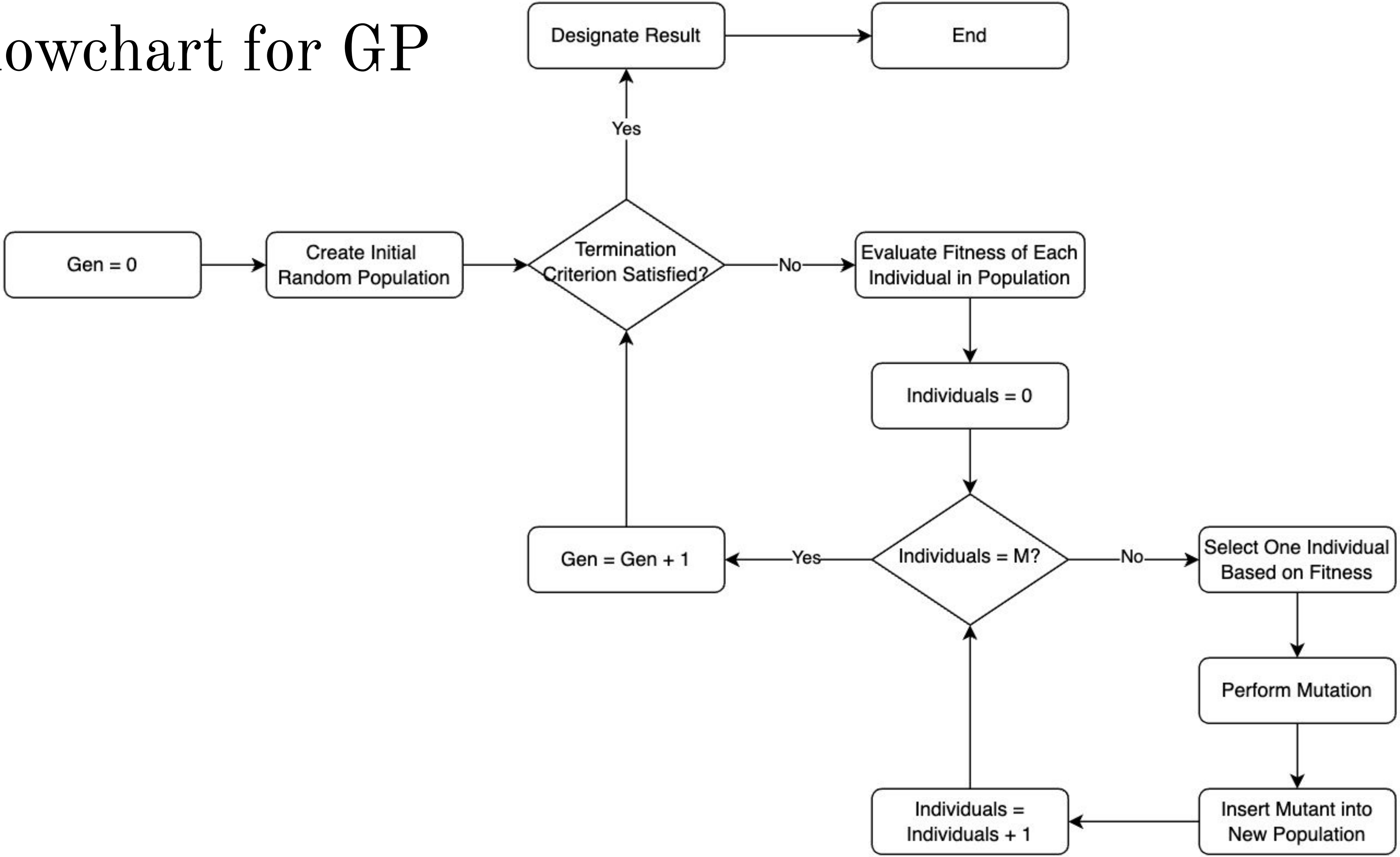
# Evolutionary Computation (EC)

- A sub-field in artificial intelligence that solves problems using evolution's basic principles
- Similar to natural selection, it refines solutions using selection and variation.
- Less effective solutions gradually disappear, while more promising ones continue to improve.
- Goal of this iterative process: to progressively enhance solution quality, aiming for an optimal or satisfactory solution.

# Genetic Programming (GP)

- A specialized branch of EC that focuses on evolving computer programs, mathematical expressions and algorithms
- Fixed-size strings or vectors VS tree-like structures or variable-length vectors
- Applications: automated software engineering, symbolic regression, the evolution of control algorithms for robotics

# Flowchart for GP



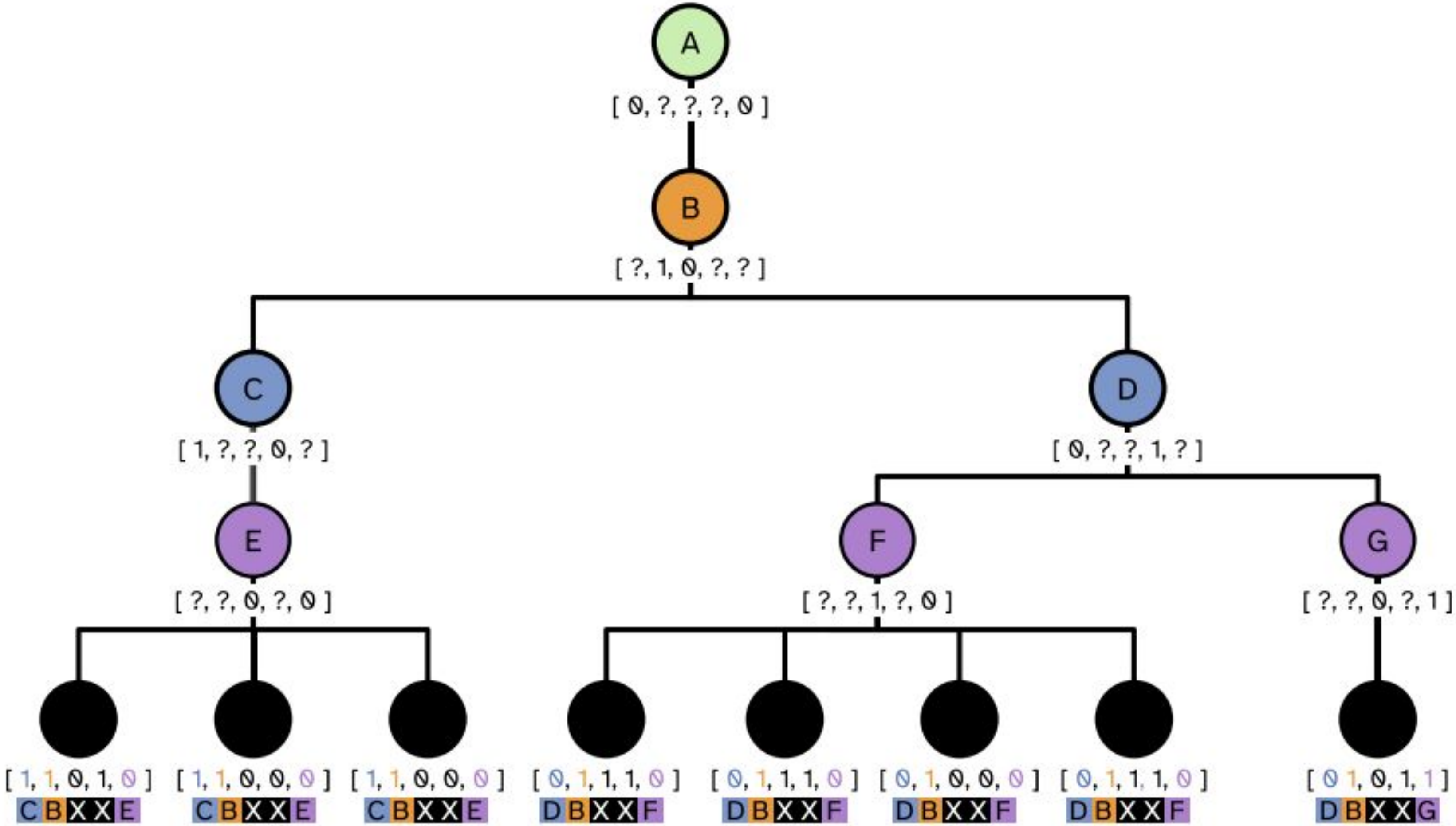
# Background

# Background

- Parent selection
  - Lexicase selection
  - Down-sampled lexicase selection

# What is selection?
# What are training cases?


Snapple


A
[ Ø, ?, ?, ?, Ø ]
B
[ ?, 1, Ø, ?, ? ]
C
[ 1, ?, ?, Ø, ? ]
D
[ Ø, ?, ?, 1, ? ]
E
[ ?, ?, Ø, ?, Ø ]
F
[ ?, ?, 1, ?, Ø ]
G
[ ?, ?, Ø, ?, 1 ]
[ 1, 1, Ø, 1, Ø ]
C B X X E
[ 1, 1, Ø, Ø, Ø ]
C B X X E
[ 1, 1, Ø, Ø, Ø ]
C B X X E
[ Ø, 1, 1, 1, Ø ]
D B X X F
[ Ø, 1, 1, 1, Ø ]
D B X X F
[ Ø, 1, Ø, Ø, Ø ]
D B X X F
[ Ø, 1, 1, 1, Ø ]
D B X X F
[ Ø 1, Ø, 1, 1 ]
D B X X G

**Result:** Individual to be used as a parent

```
candidates := the entire population
cases := list of all test cases in a random order
while True do
    candidates := candidates who perform best on case[0]
    if only one candidate exists in candidates then
        return candidate
    end
    if cases is empty then
        return a randomly selected candidate from candidates
    end
    delete case[0]
end
```

**Algorithm 1:** Lexicase Selection

shuffled cases: ~~2~~, ~~0~~, 1, 3

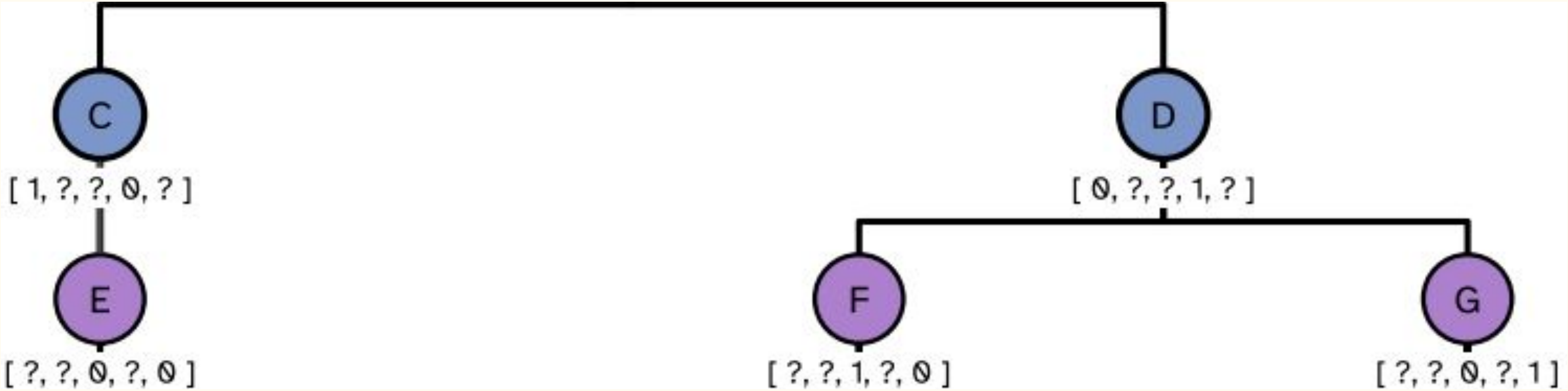
| case | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| ~~e0~~ | ~~36~~ | ~~80~~ | ~~84~~ | ~~40~~ |
| e1 | 47 | 2 | 84 | 30 |
| ~~e2~~ | ~~34~~ | ~~72~~ | ~~38~~ | ~~72~~ |
| ~~e3~~ | ~~32~~ | ~~96~~ | ~~84~~ | ~~72~~ |
| e4 | 47 | 12 | 84 | 36 |
| ~~e5~~ | ~~17~~ | ~~37~~ | ~~84~~ | ~~80~~ |
| e6 | 47 | 18 | 84 | 37 |
| e7 | 47 | 23 | 84 | 84 |
| ~~e8~~ | ~~40~~ | ~~20~~ | ~~32~~ | ~~17~~ |
| ~~e9~~ | ~~87~~ | ~~25~~ | ~~6~~ | ~~84~~ |

shuffled cases: 2, 0, 1, 3

| case | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| e0 | 36 | 80 | 84 | 40 |
| e1 | 47 | 2 | 84 | 30 |
| e2 | 34 | 72 | 38 | 72 |
| e3 | 32 | 96 | 84 | 72 |
| e4 | 47 | 12 | 84 | 36 |
| e5 | 17 | 37 | 84 | 80 |
| e6 | 47 | 18 | 84 | 37 |
| e7 | 47 | 23 | 84 | 84 |
| e8 | 40 | 20 | 38 | 17 |
| e9 | 87 | 25 | 6 | 84 |

# Down-sampled Lexicase Selection

- Random subsampling the training set for each generation
- Significantly reducing the computational demands

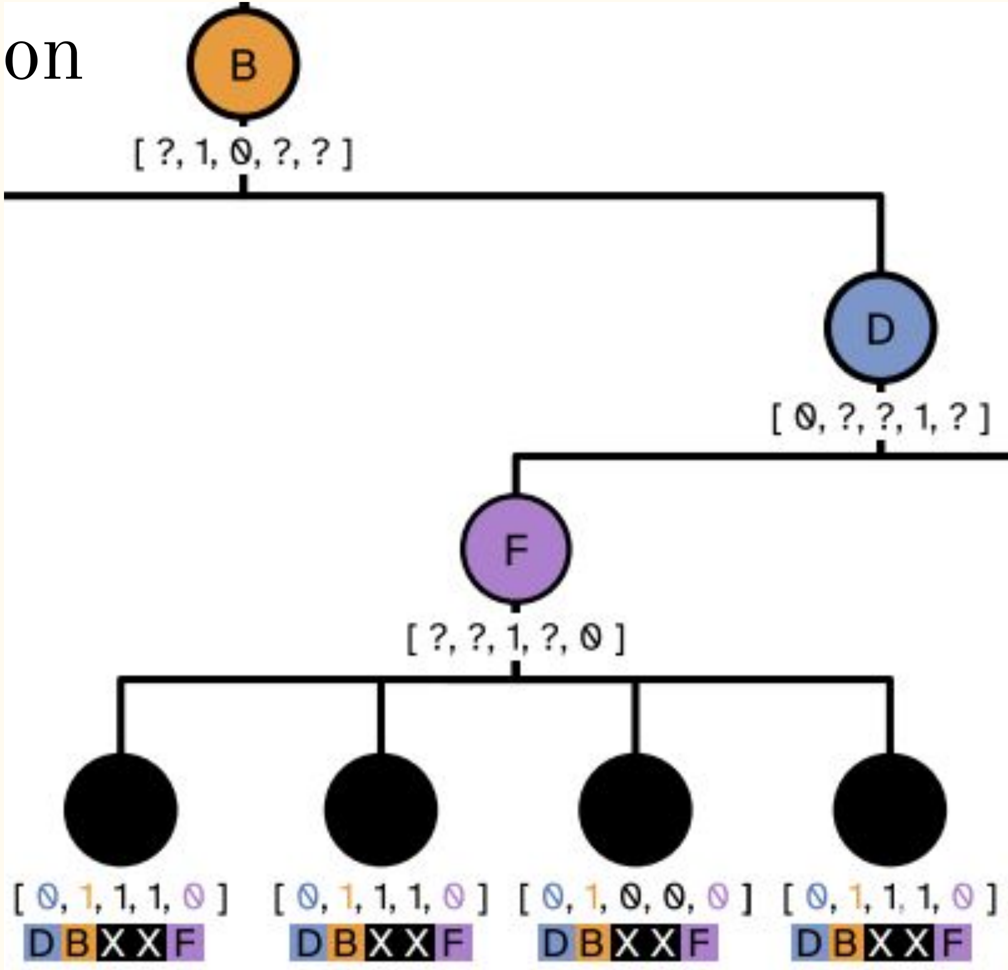


# Methods

# Methods

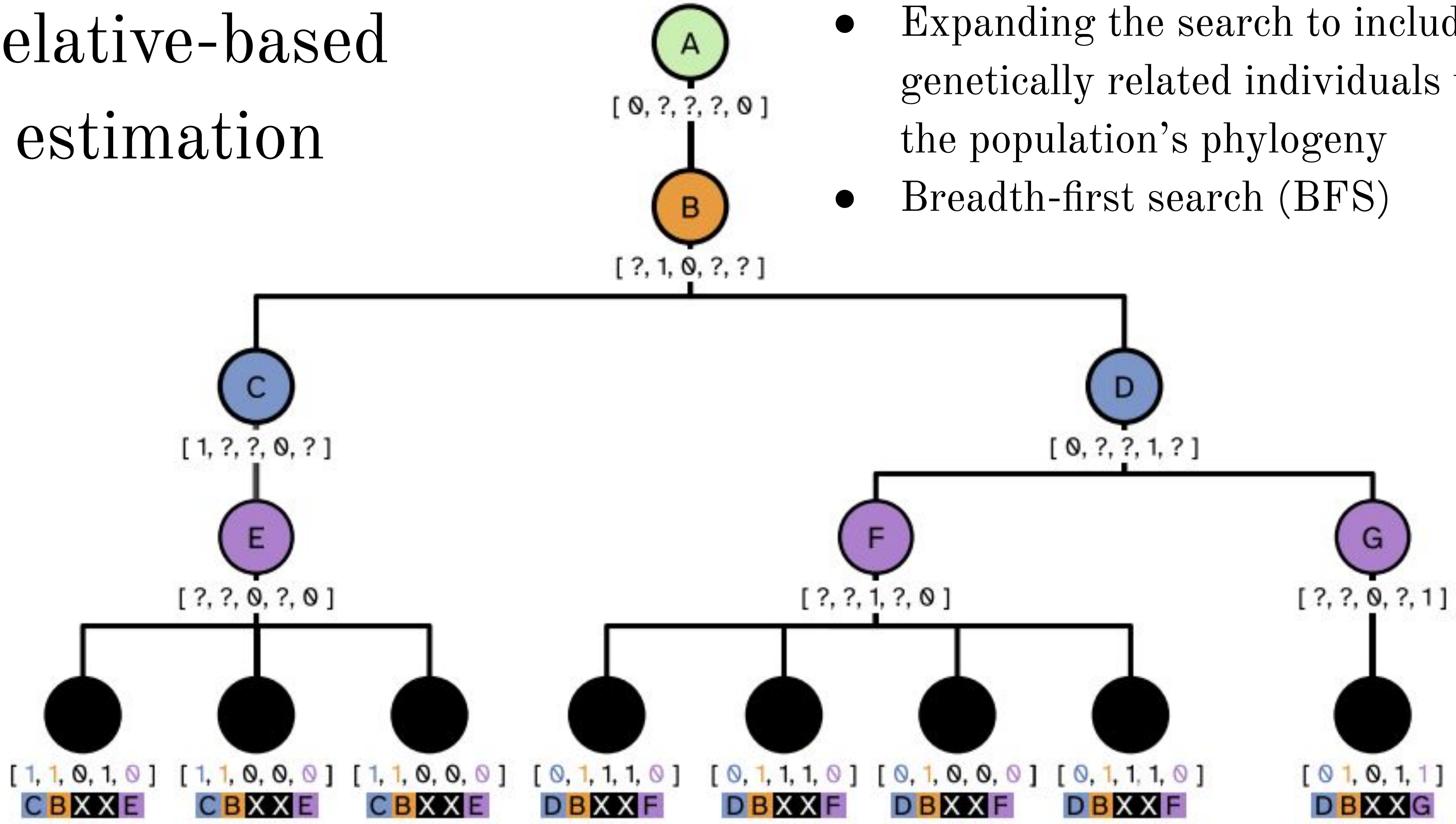
- Phylogeny-informed fitness estimation
  - Ancestor-based estimation
  - Relative-based estimation
- Genetic programming experiments
  - Experimental setup
  - Program synthesis problems

# Ancestor-based estimation




- Tracing the lineage of an individual backward through its ancestors
- Preserving the integrity of the evolutionary process
- Streamlining the computational demands

# Relative-based estimation




- Expanding the search to include any genetically related individuals within the population's phylogeny
- Breadth-first search (BFS)

# Experimental setup

- Comparing the problem-solving success among:
  - **3** estimation models
  - **4** GP problems
  - **4** subsampling levels
- Phylogeny searches depth is limited to **5**
- Running **30** replicates of each condition
- Evolving a population of **1,000** linear genetic programs in each replicate

# Program synthesis problems

- **Median**

  Programs are given three integer inputs ($-100 \leq$ input $i \leq 100$) and must output the median value.

- **Small or Large**

  Programs are given an integer $n$ and must output

  - "small" if $n < 1000$
  - "large" if $n \geq 2000$
  - "neither" if $1000 \leq n < 2000$

# Program synthesis problems

- **Grade**

  Programs receive five integer inputs in the range $[0, 100]$: $A$, $B$, $C$, $D$, and score. $A$, $B$, $C$ and $D$ are monotonically decreasing and unique, each defining the minimum score needed to receive that "grade". The program must read these thresholds and return the appropriate letter grade for the given score or return $F$ if score $< D$.

- **Fizz Buzz**

  Given an integer $x$, the program must return *"Fizz"* if x is divisible by 3, *"Buzz"* if $x$ is divisible by 5, *"FizzBuzz"* if $x$ is divisible by both 3 and 5, and $x$ if none of the prior conditions are true.

# Results

# Results

- Phylogeny-informed estimation reduces diversity loss caused by subsampling
- Phylogeny-informed estimation improves poor exploration caused by down-sampling
- Phylogeny-informed estimation can enable extreme subsampling for some genetic programming problems

# Enabling extreme subsampling for some genetic programming problems

a **Down-sampled lexicase**

| Problem | 1% subsampling | | | 5% subsampling | | | 10% subsampling | | | 50% subsampling | | | Full |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | None | Anc. | Rel. | None | Anc. | Rel. | None | Anc. | Rel. | None | Anc. | Rel. | None |
| Median | 8 | 13 | 14 | 4 | ***19*** | ***23*** | 16 | 21 | 22 | 2 | ***15*** | ***13*** | 1 |
| Small or large | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Grade | 1 | ***10*** | ***11*** | $22^{\dagger}$ | 12 | 11 | ***22*** | 13 | 11 | 5 | 9 | 4 | 1 |
| Fizz buzz | 0 | 0 | 0 | ***20*** | 2 | 2 | 8 | 8 | 7 | 0 | ***7*** | ***7*** | 0 |

- In most combinations, the performances of ancestor-based estimation and relative-based estimation are close.
- In some combinations, there are statistically significant differences between no-estimation control and phylogeny-informed fitness estimation.

# Conclusion

# Conclusion

- Ancestor-based estimation > relative-based estimation for more efficient optimization
- The phylogeny-informed approach allows for individual evaluations on varied training set subsets, potentially increasing accuracy and problem-solving success if training cases are subsampled to minimize phylogenetic distance.
- Beyond fitness estimation, runtime phylogeny tracking might enhance evolutionary search broadly, particularly in quality diversity algorithms that emphasize phenotypic or behavioral diversity.

# Acknowledgements

# References

[1] Alexander Lalejini, Matthew Andres Moreno, Jose Guadalupe Hernandez, and Emily Dolson 2023. Phylogeny-Informed Fitness Estimation for Test-Based Parent Selection

[2] R. Boldi, M. Briesch, D. Sobania, A. Lalejini, T. Helmuth, F. Rothlauf, C. Ofria and L. Spector. 2023. Informed Down-Sampled Lexicase Selection: Identifying productive training cases for efficient problem solving

[3] Jose Guadalupe Hernandez, Alexander Lalejini, Emily Dolson, Charles Ofria
2019. Random subsampling improves performance in lexicase selection
Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion)

[4] Thomas Helmuth, Lee Spector
2021. Problem-Solving Benefits of Down-Sampled Lexicase Selection

[5] Blossom Metevier, Anil Kumar Saini, Lee Spector
2019. Lexicase Selection Beyond Genetic Programming