

# Generating Structured Test Data with Specific Properties using Metaheuristic and Nested Monte-Carlo Search

---

Simon Poulding

Blekinge Institute of Technology

Robert Feldt

Chalmers University

## Context

software testing: generating test data for software for which inputs are highly structured

## Problem

how to enable the test engineer to generate test data that is both valid and has desirable properties?

## Solution

- (1) allow the test engineer to specify the construction of valid test inputs using generators written in a general purpose language
- (2) apply metaheuristic and Monte-Carlo tree search algorithms to optimise the generation process for the desirable properties

GödelTest

+ Metaheuristic Search

+ Nested Monte-Carlo Search

Application: Robustness Testing

generator

```
@generator AZStrings begin
  start() = String(mult(letter))
  letter() = choose(Bool)? 'A' : 'B'
end
```

generator

```
@generator AZStrings begin
  start() = String(mult(letter))
  letter() = choose(Bool)? 'A' : 'B'
end
```

choice model

choice point

range:  $[0, \infty)$

distribution: geometric ( $p_1$ )

choice point

range:  $[false, true]$

distribution: Bernoulli ( $p_2$ )

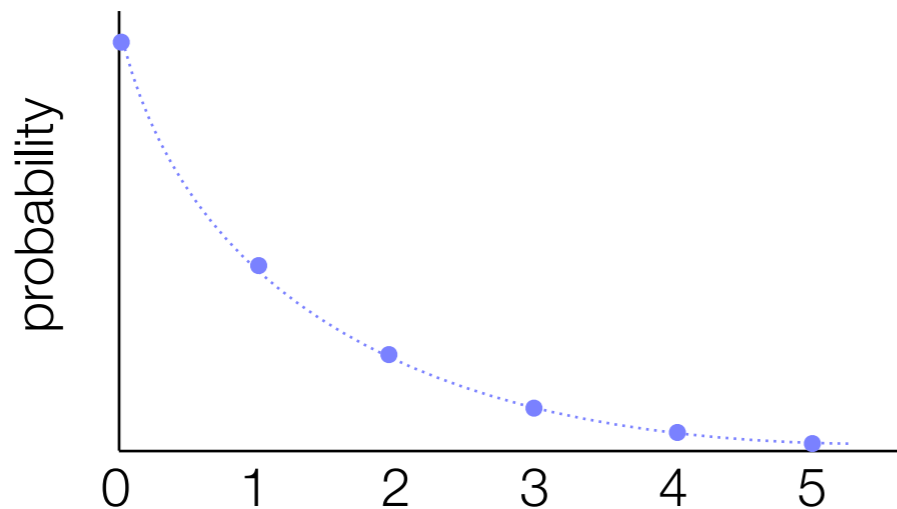
## generator

```
@generator ABStrings begin
  start() = String(mult(letter))
  letter() = choose(Bool)? 'A' : 'B'
end
```

## choice point

range:  $[0, \infty)$

distribution: geometric ( $p_1=0.5$ )



“AB”  
“A”  
“”  
“BBA”

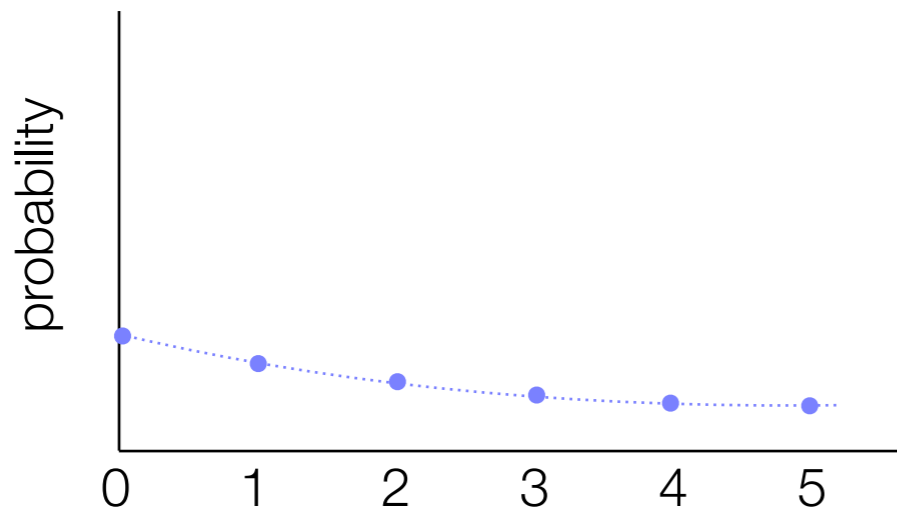
## generator

```
@generator ABStrings begin
  start() = String(mult(letter))
  letter() = choose(Bool)? 'A' : 'B'
end
```

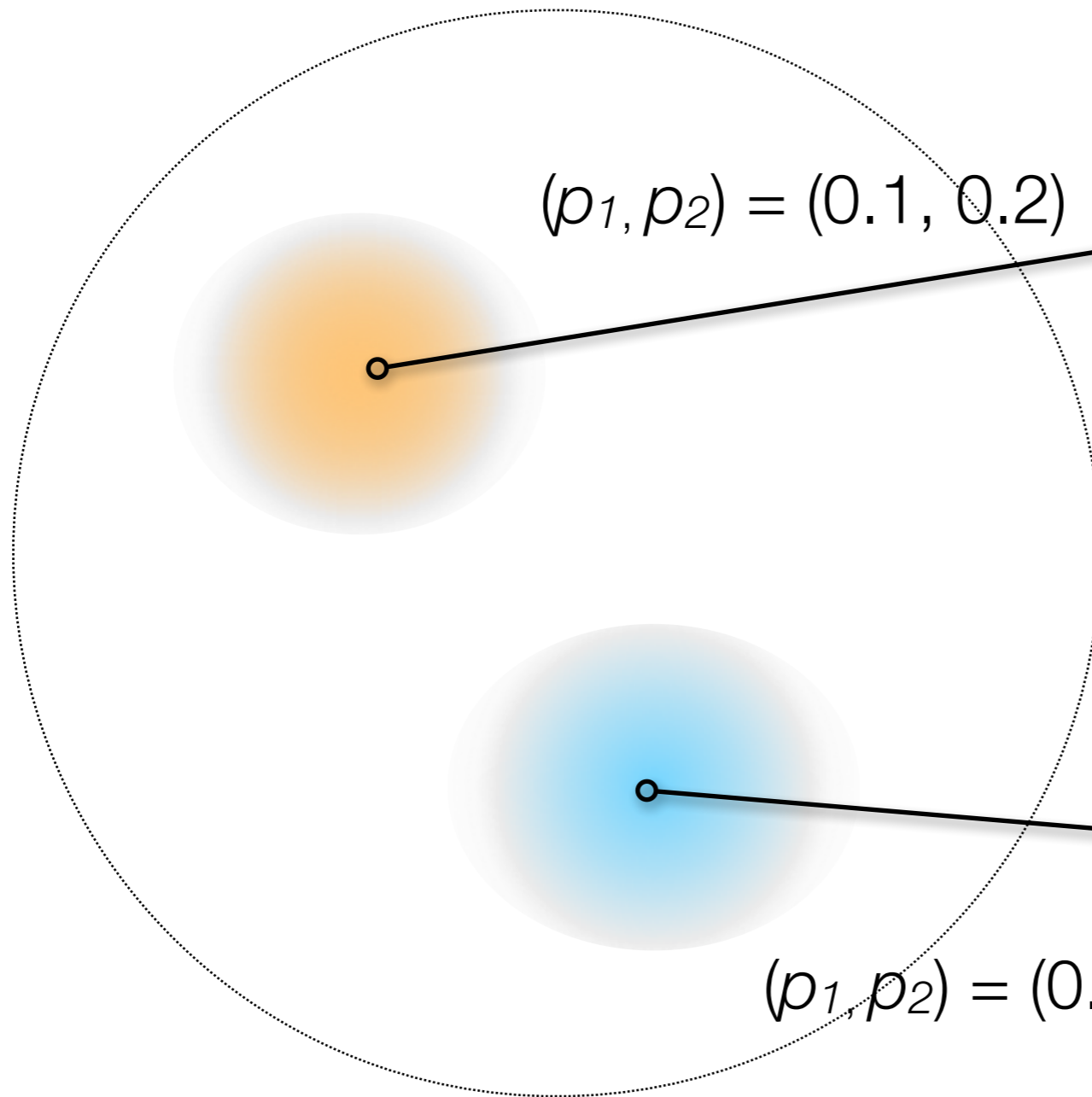
## choice point

range:  $[0, \infty)$

distribution: geometric ( $p_1=0.1$ )



“BBABAB”  
“BBAAABAA”  
“ABBA”  
“AABAAB”



“AAABAA”  
“ABAAAAAA”  
“AAA”  
“BAAAAB”

“BAB”  
“B”  
“”  
“ABBA”

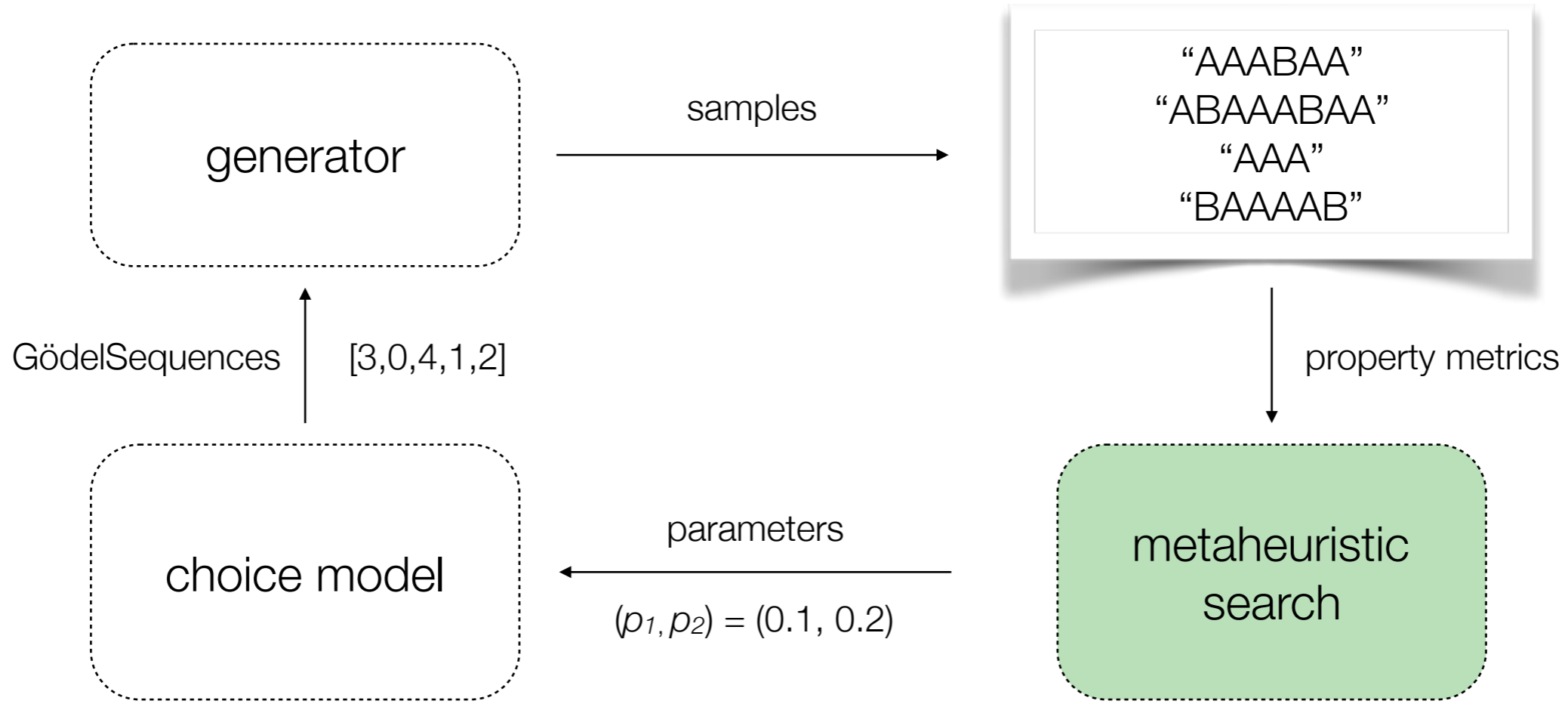


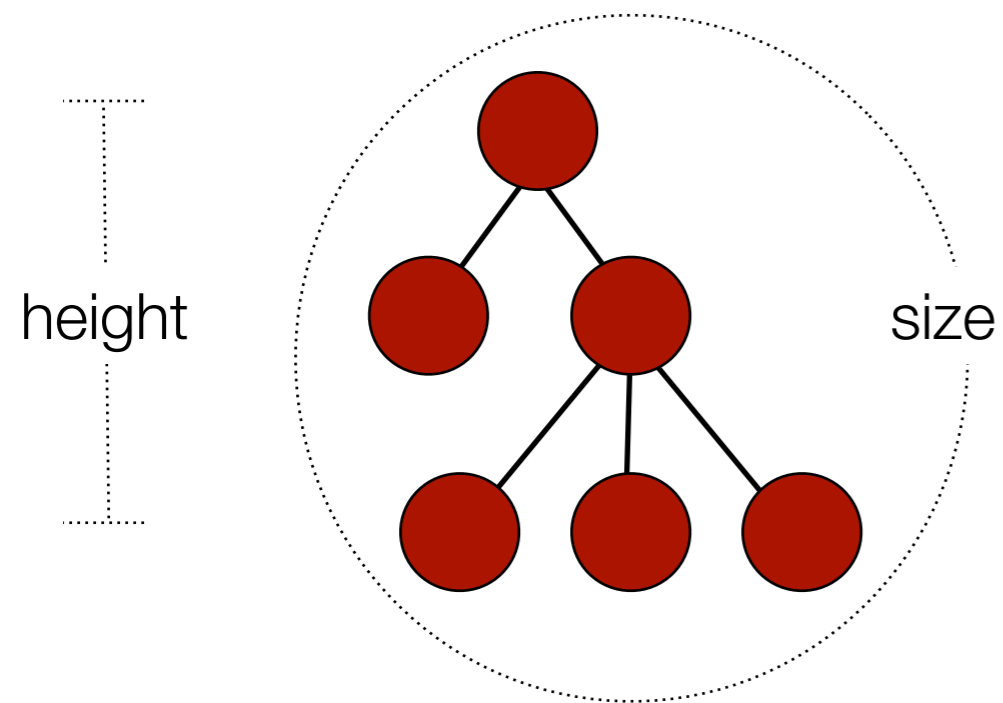
GödelTest

+ Metaheuristic Search

+ Nested Monte-Carlo Search

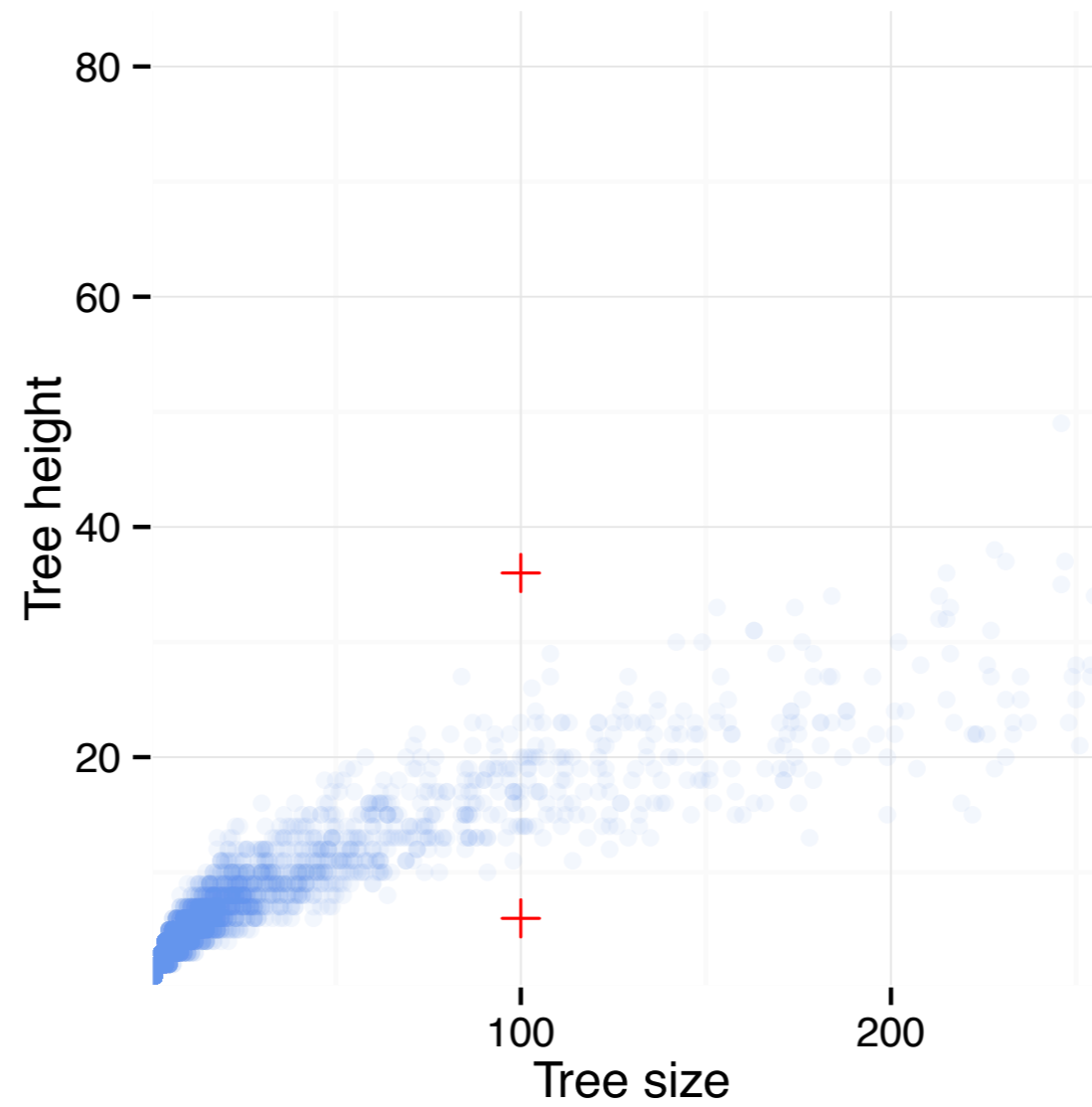
Application: Robustness Testing



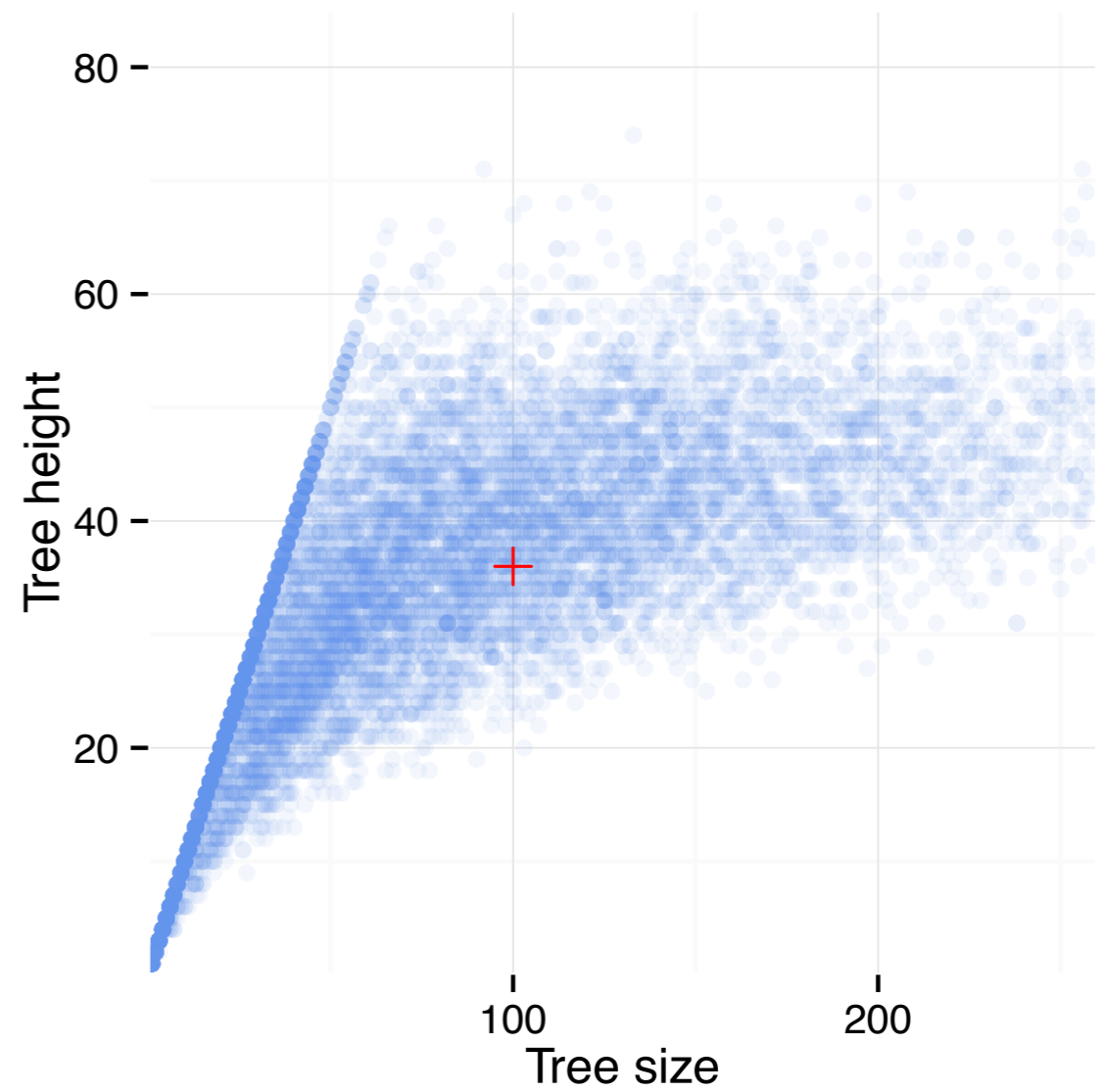
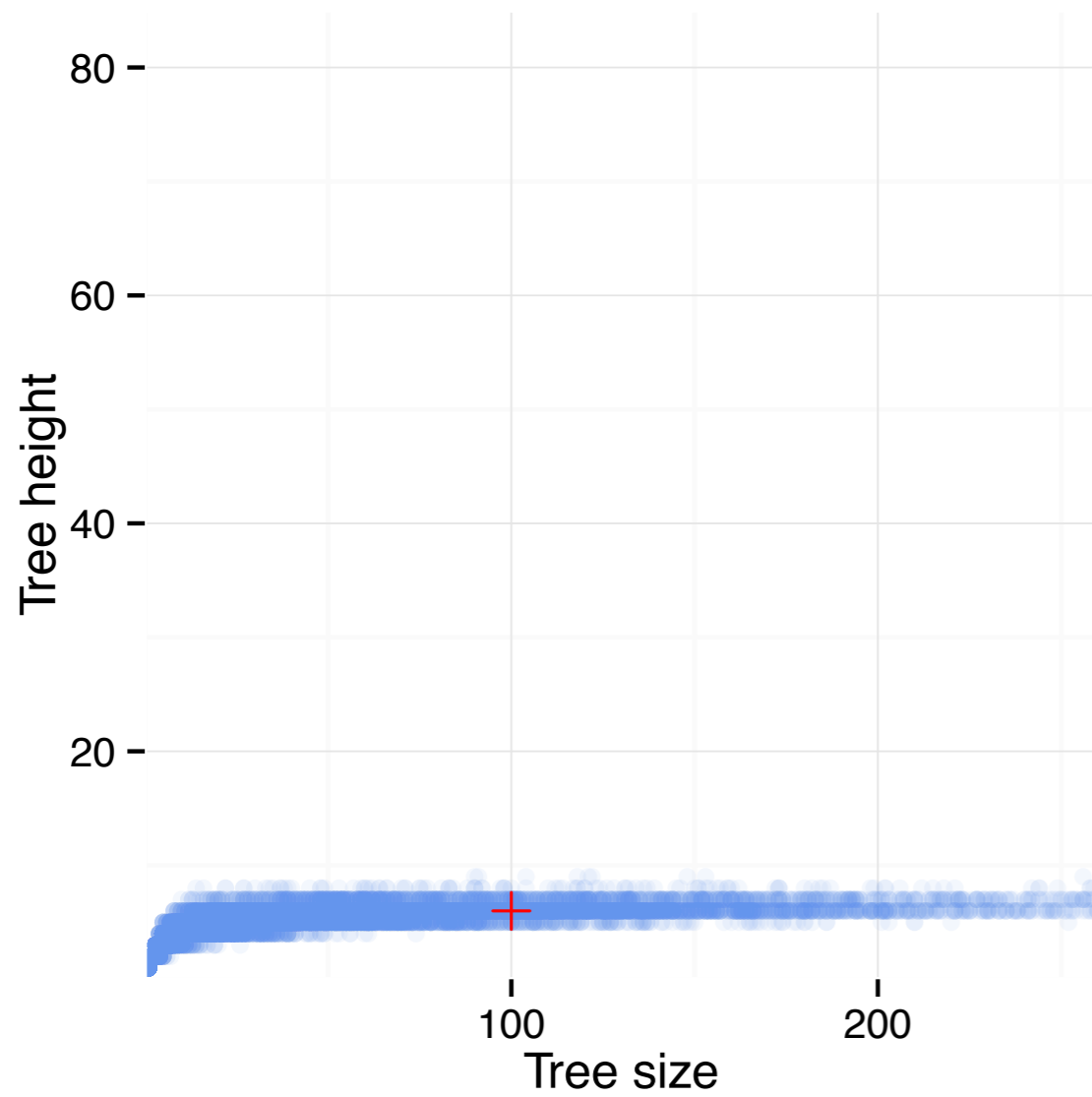


**target 1:** size = 100 AND height = 36

**target 2:** size = 100 AND height = 6



Boltzmann Sampler



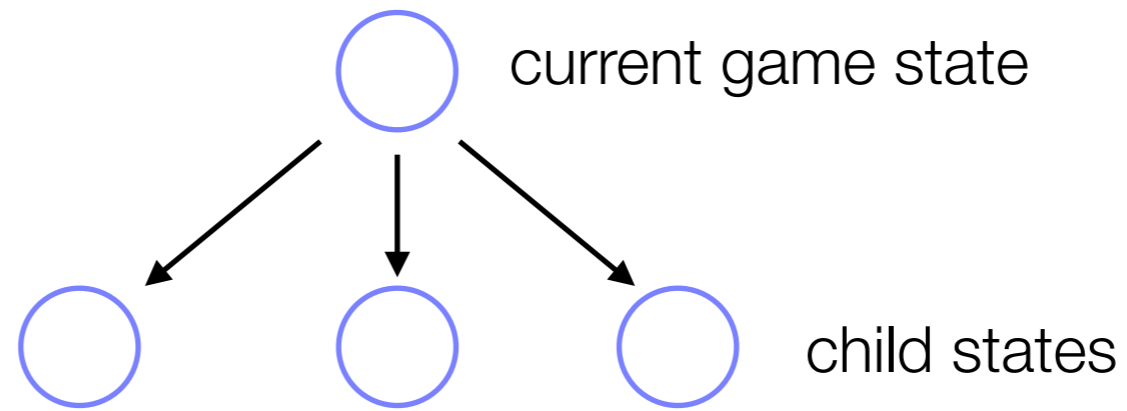
GödelTest using Differential Evolution

GödelTest

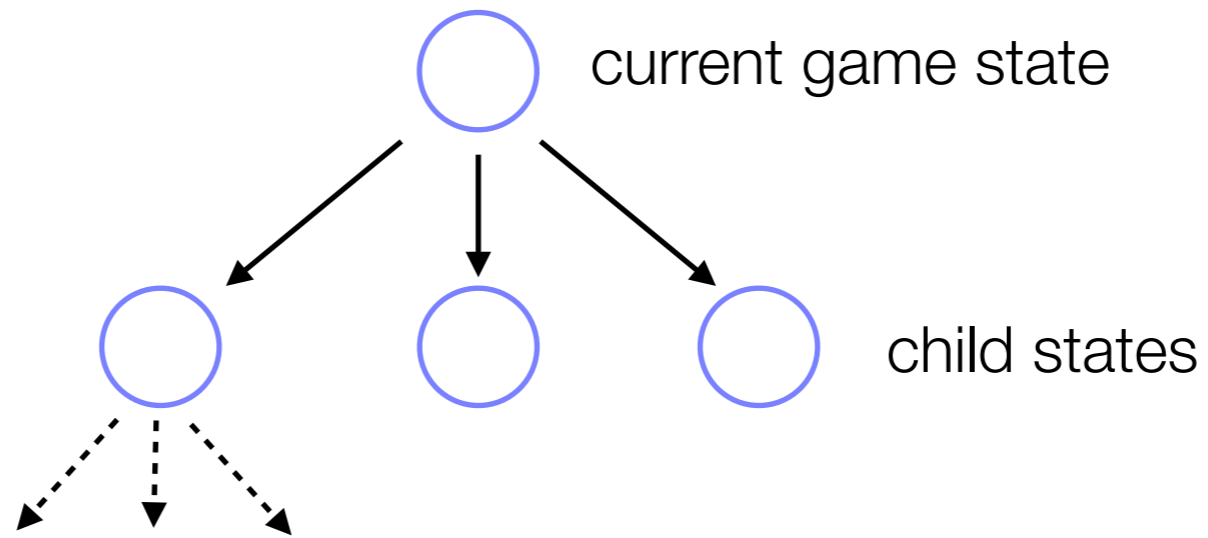
+ Metaheuristic Search

+ Nested Monte-Carlo Search

Application: Robustness Testing



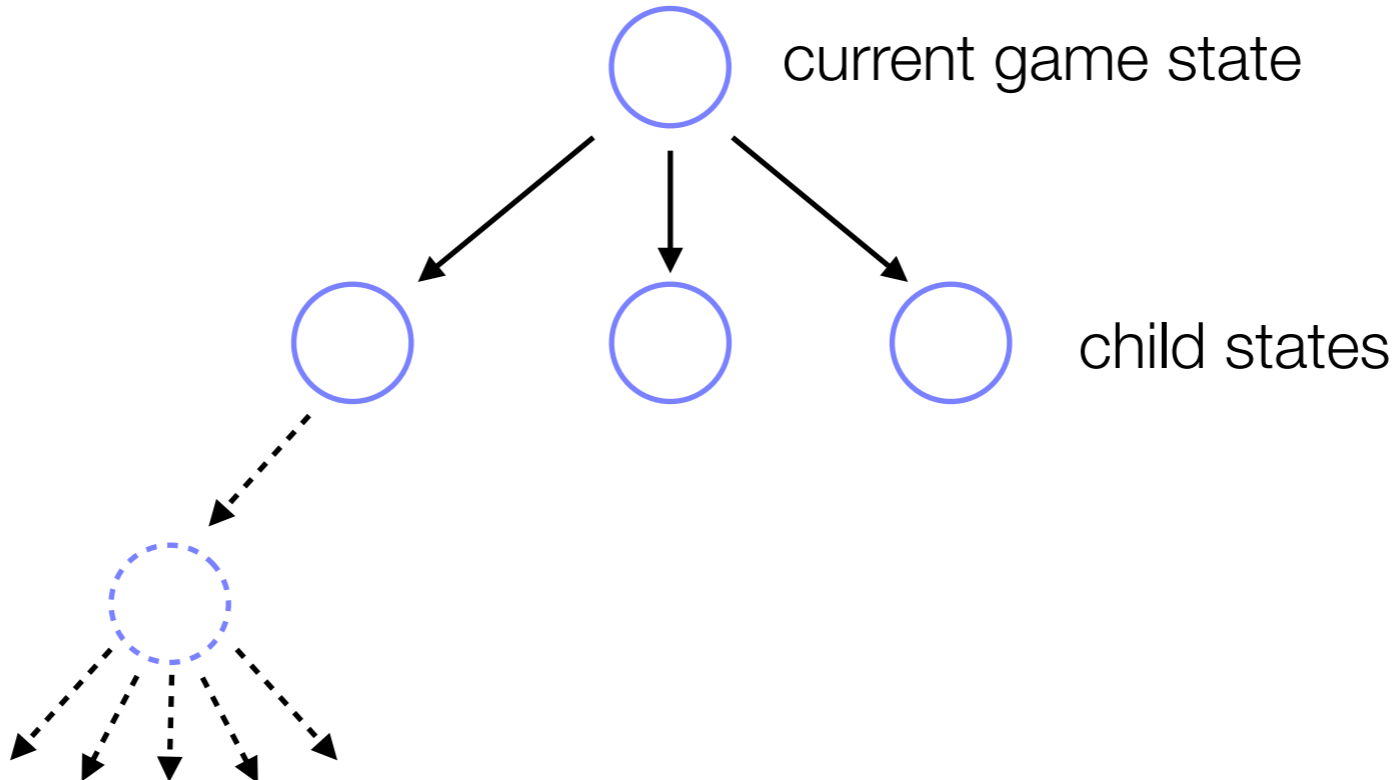
simulation



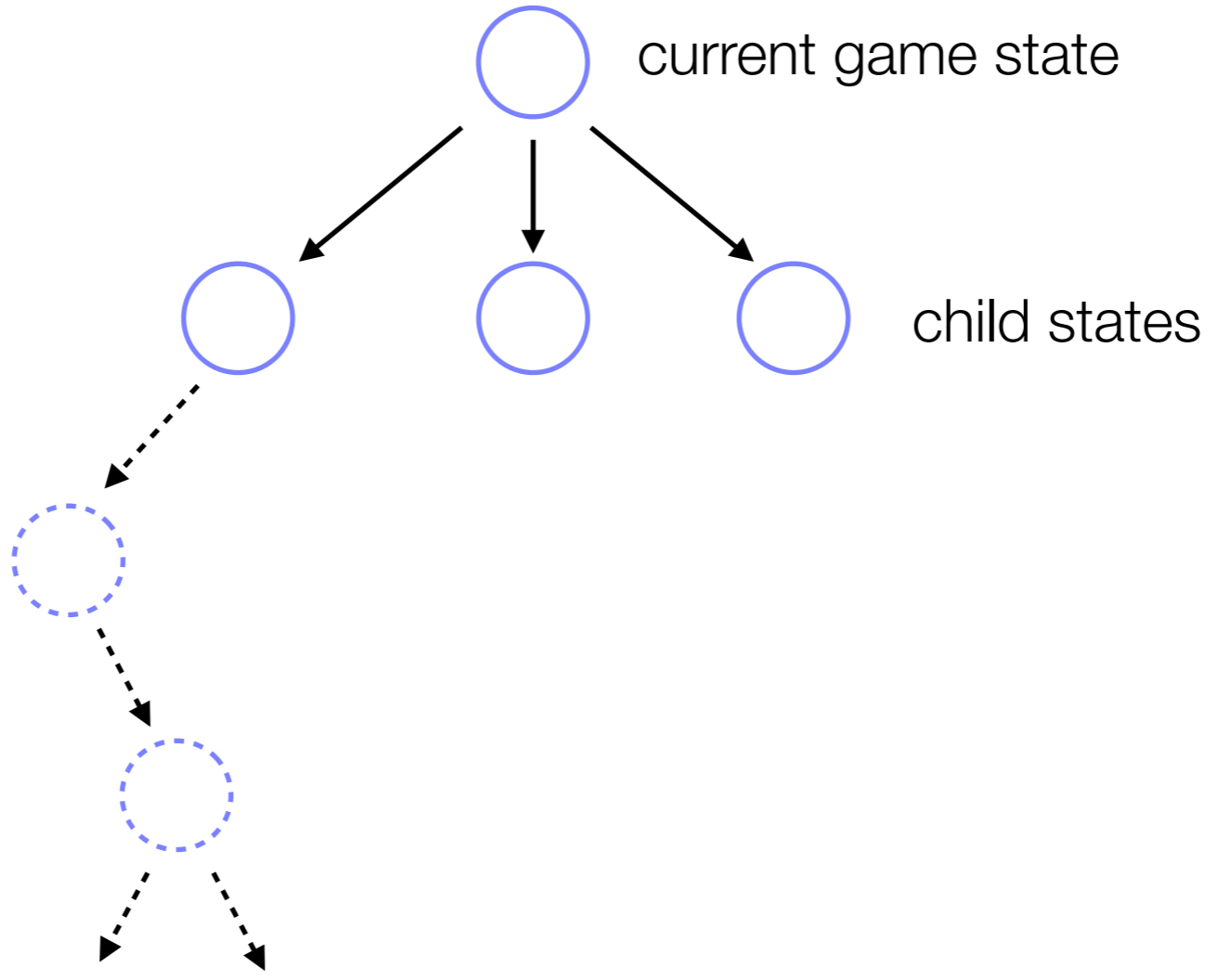
simulation



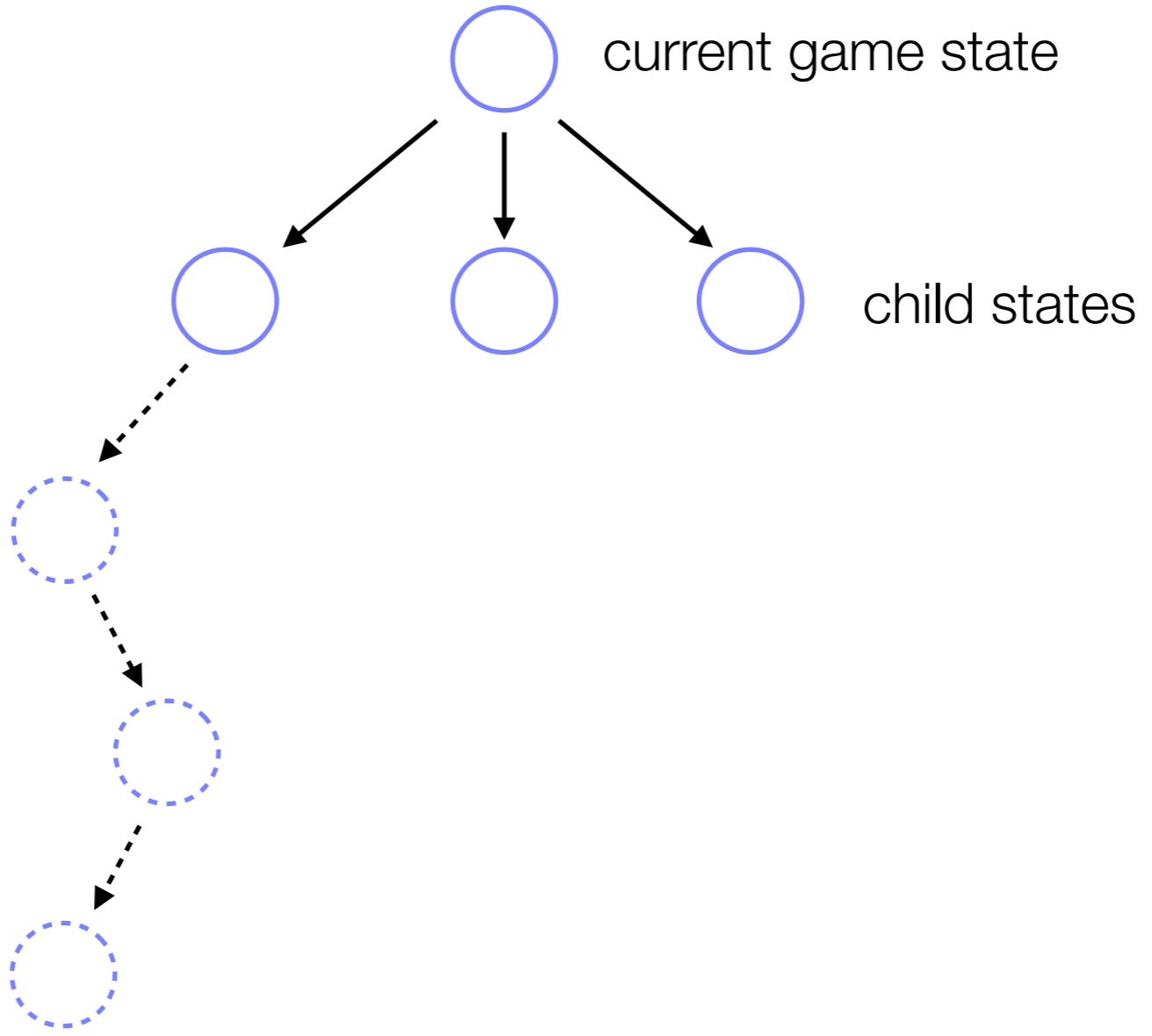
simulation



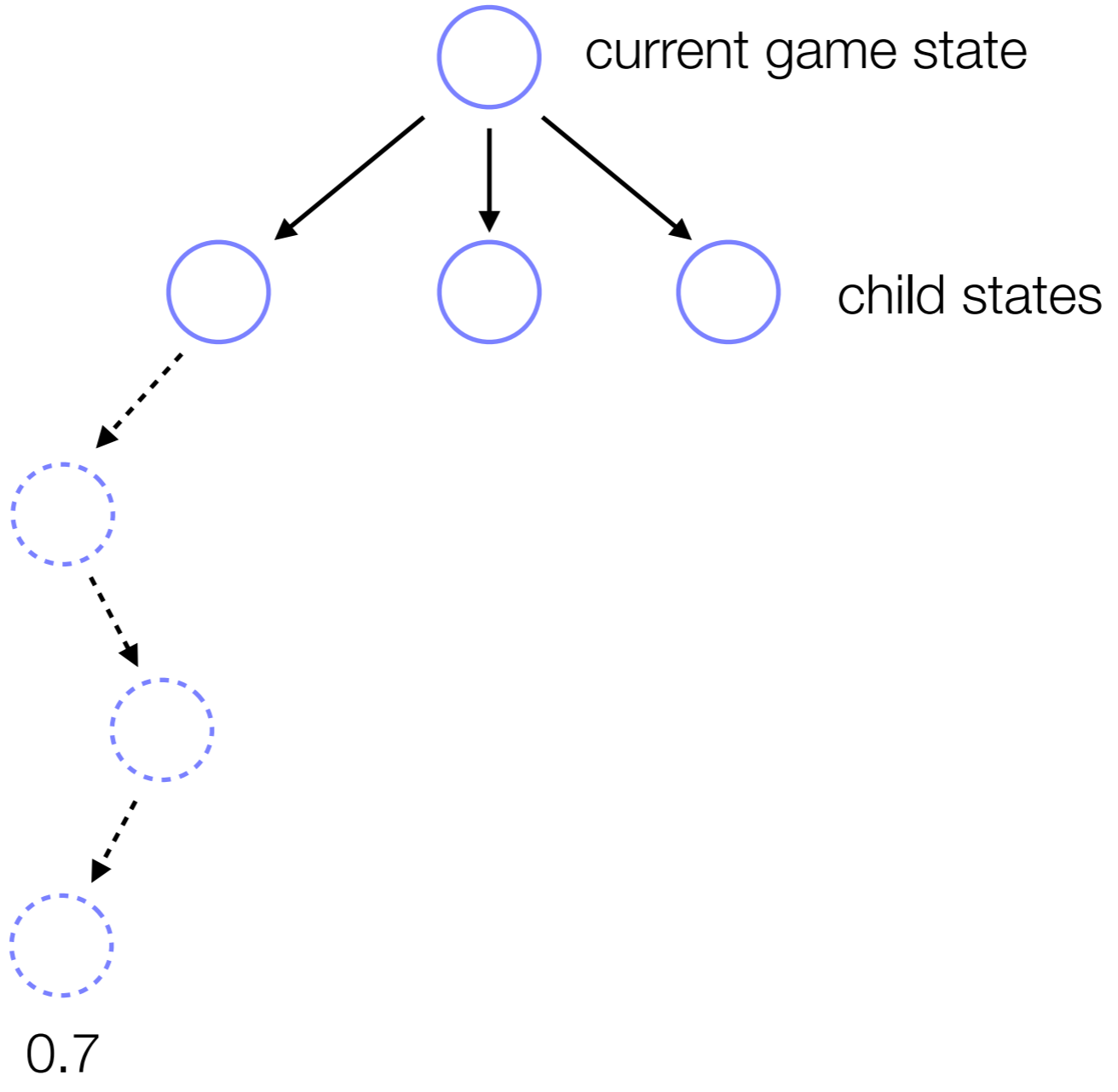
simulation



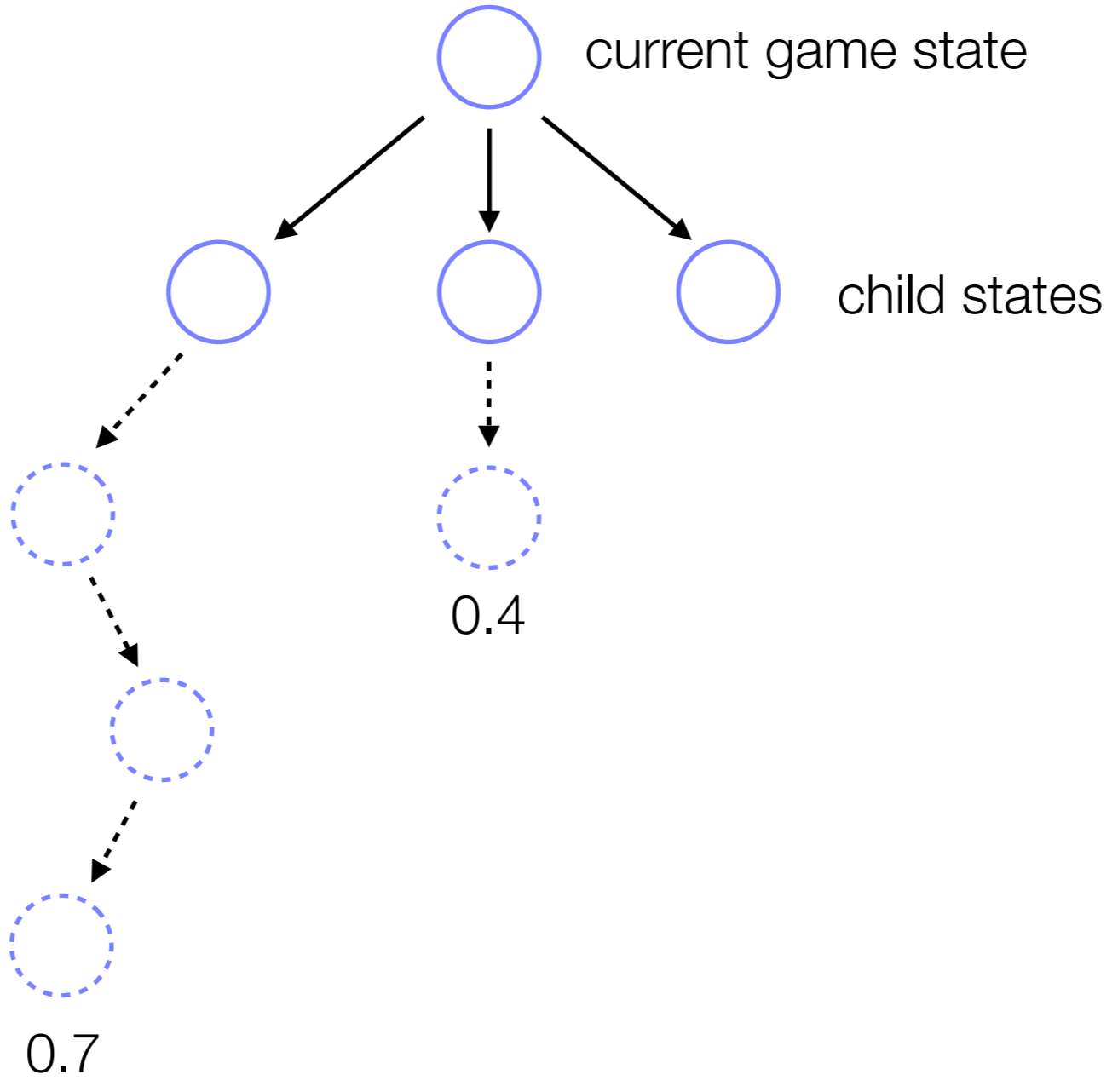
simulation



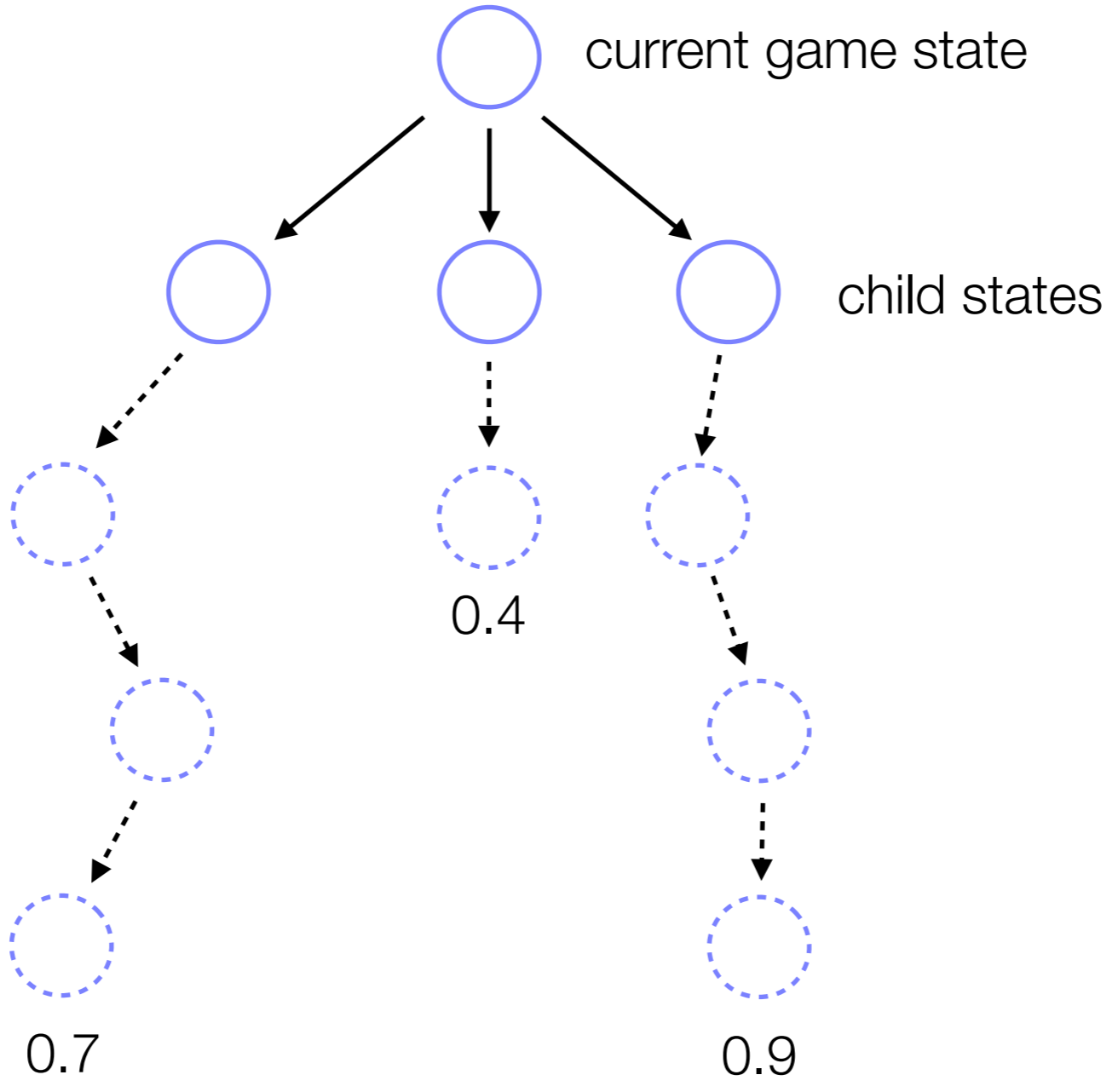
simulation

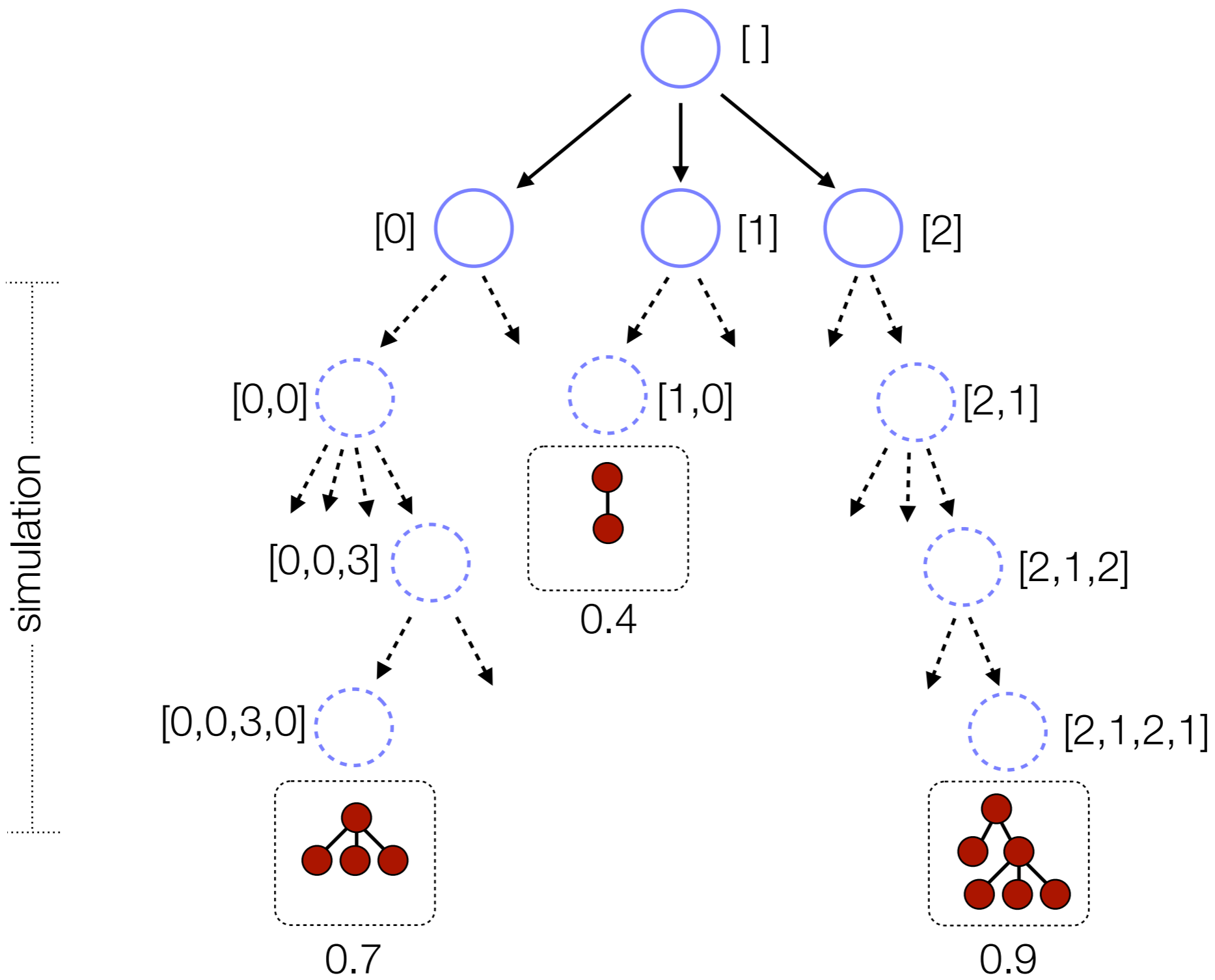


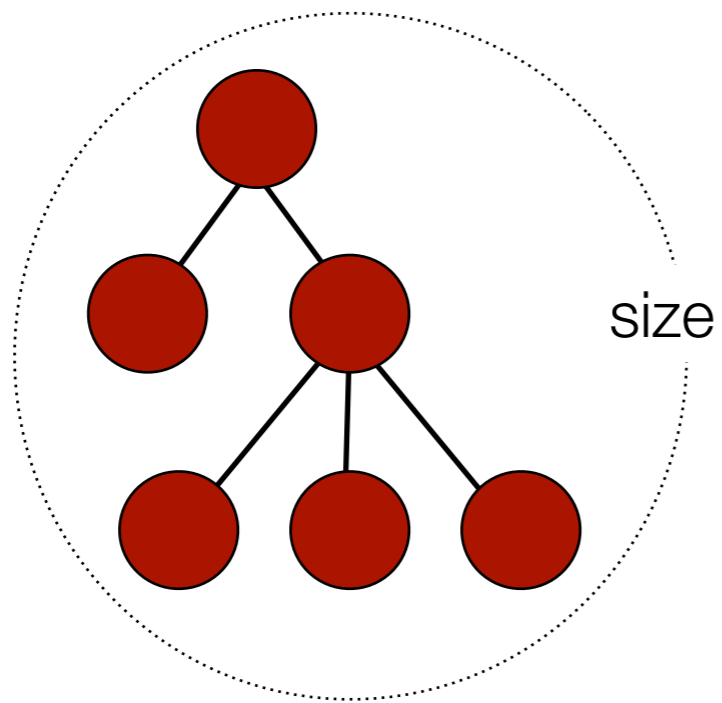
simulation



simulation



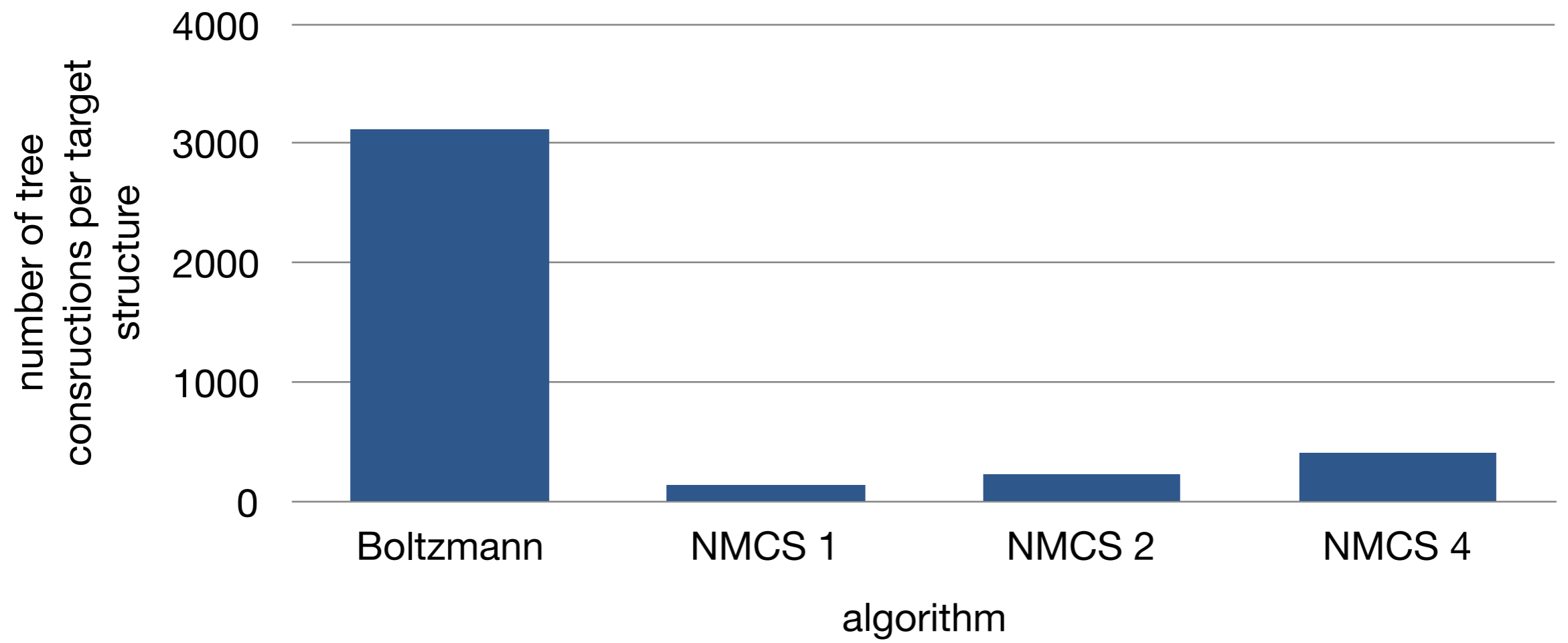


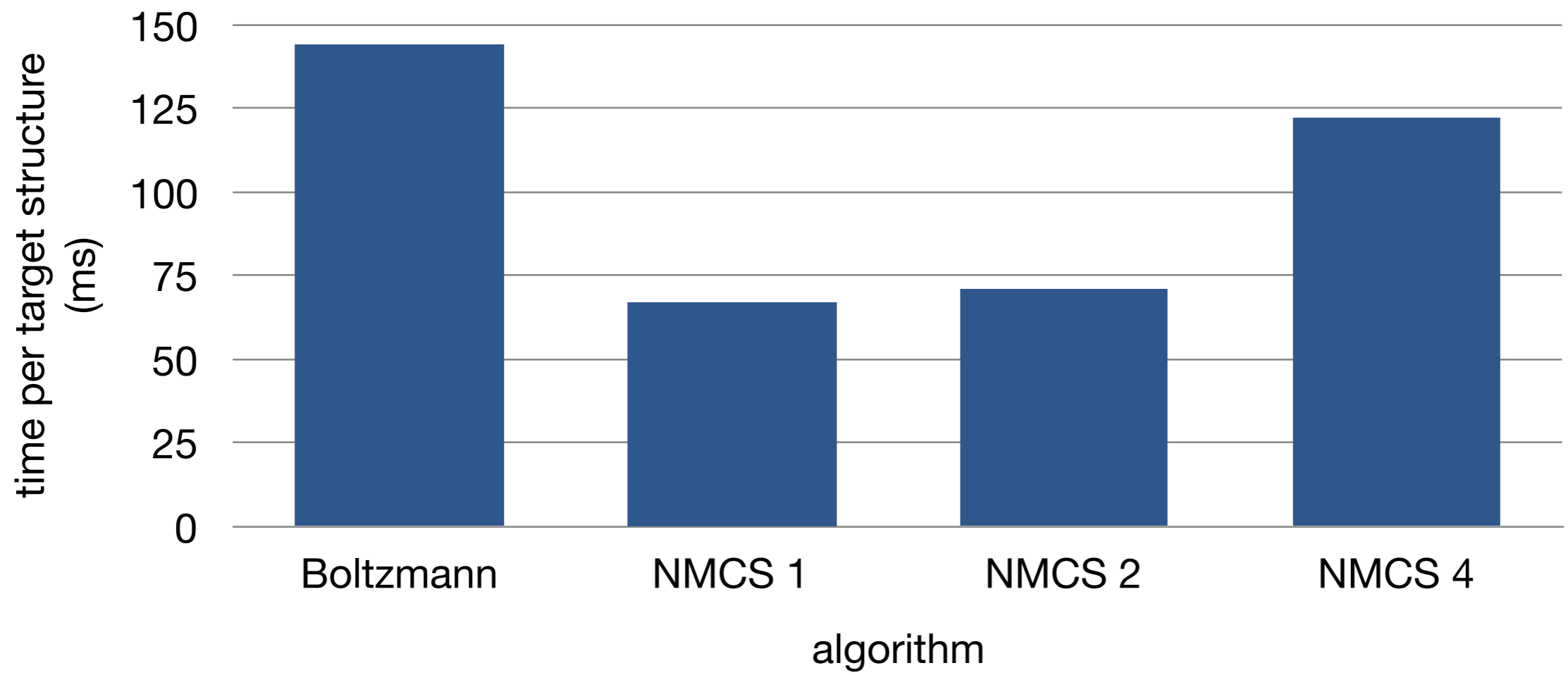


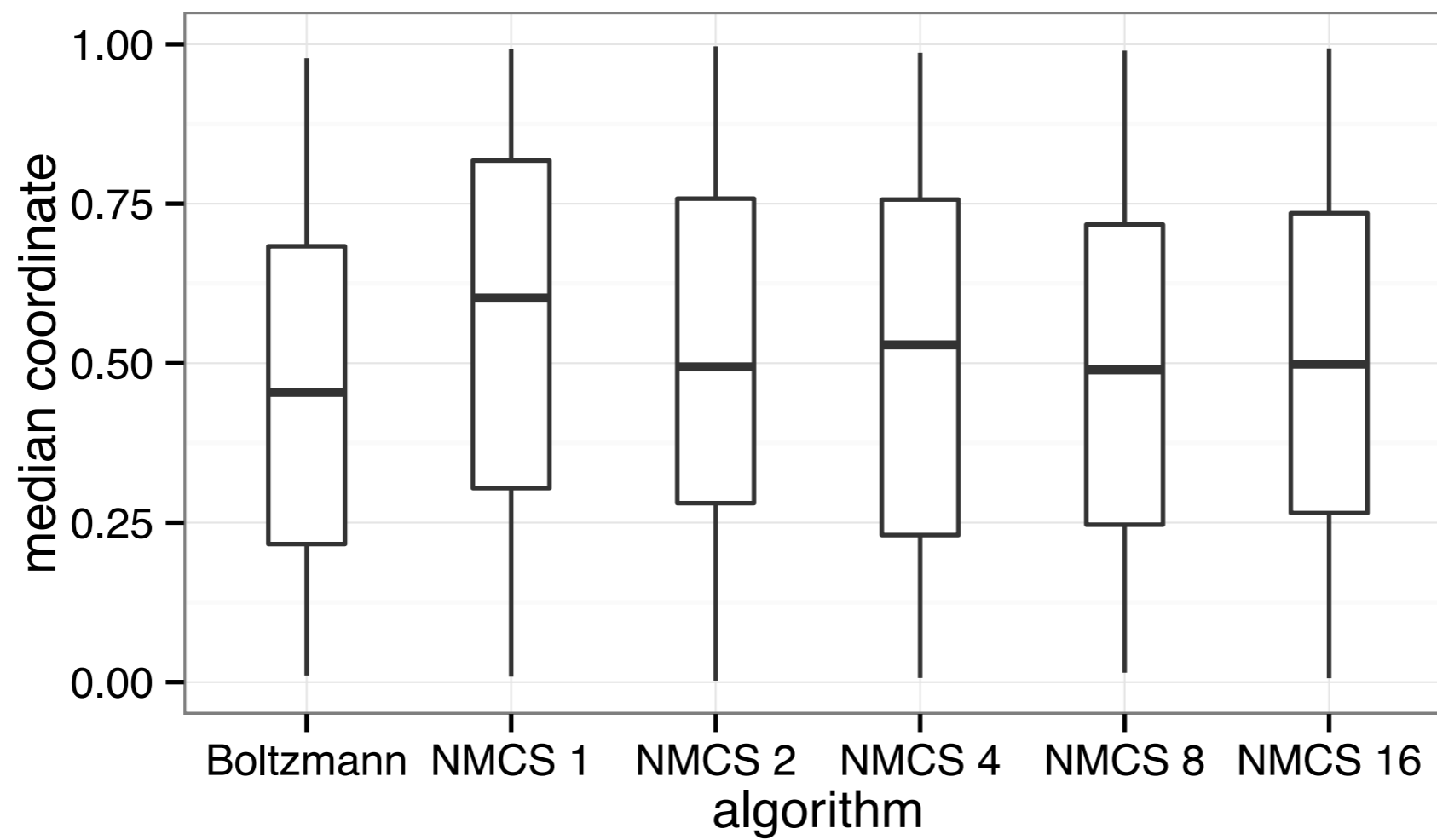
**target:** size = 100

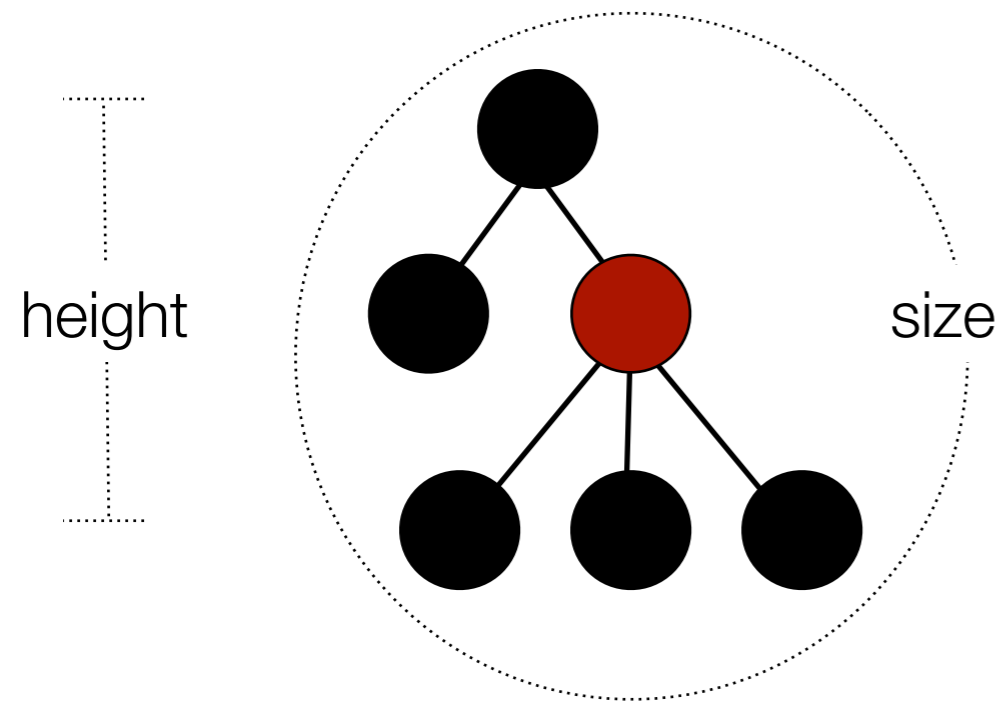
Simon Poulding and Robert Feldt,  
Generating structured test data with specific properties using nested Monte-Carlo search,  
Proc. Genetic and Evolutionary Computation Conference (GECCO), 1279-1286, 2014



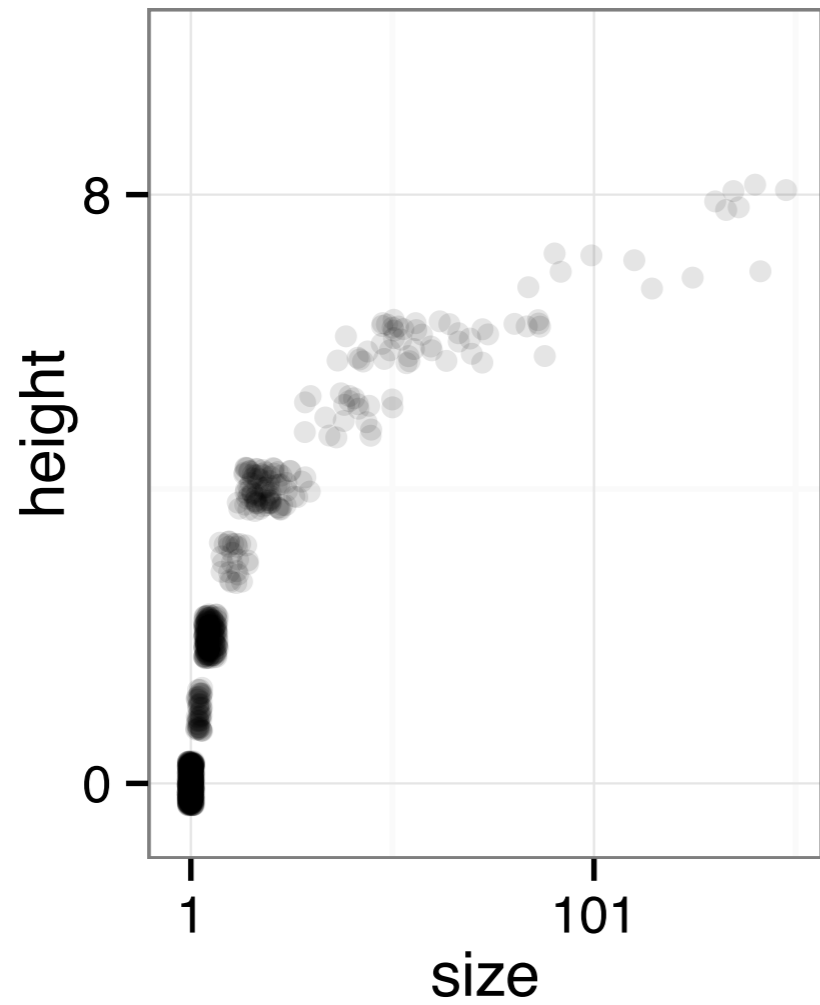




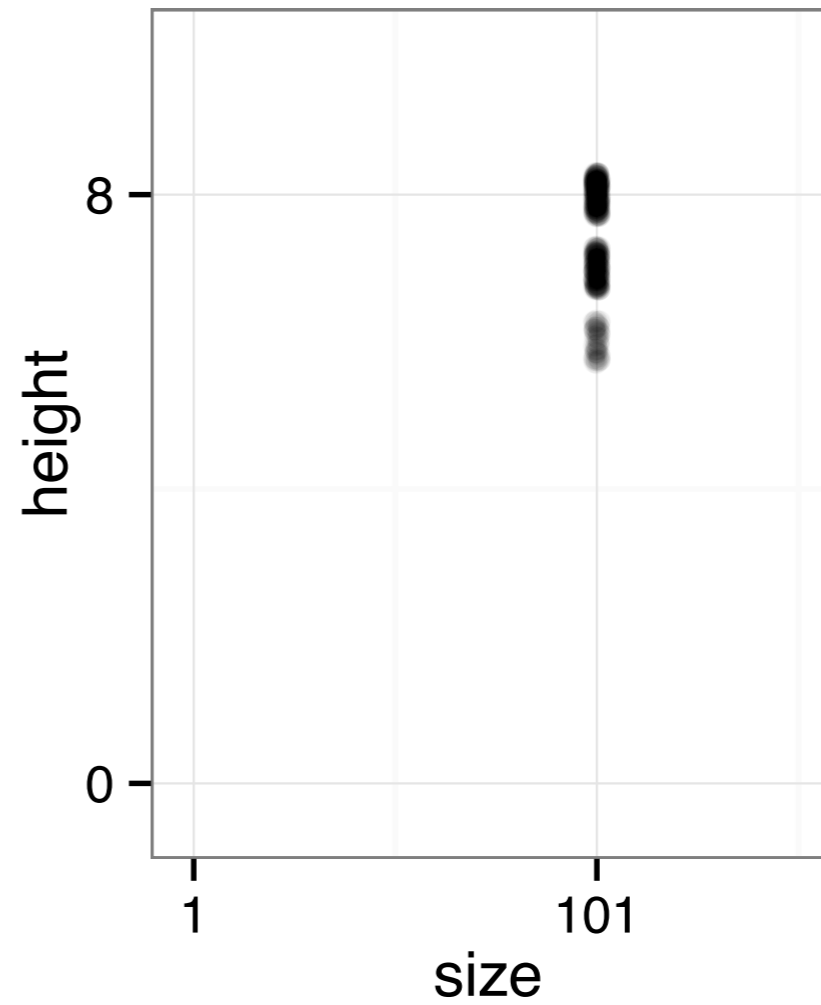




**target:** size = 101 AND height = 8



'random' sampler



NMCS

GödelTest

+ Metaheuristic Search

+ Nested Monte-Carlo Search

Application: Robustness Testing

```
<math>
  <frac>
    <row>
      <mn> 1 </mn>
      <mo> + </mo>
      <msqrt>
        <mn> 5 </mn>
      </msqrt>
    </row>
    <mn> 2 </mn>
  </frac>
</math>
```

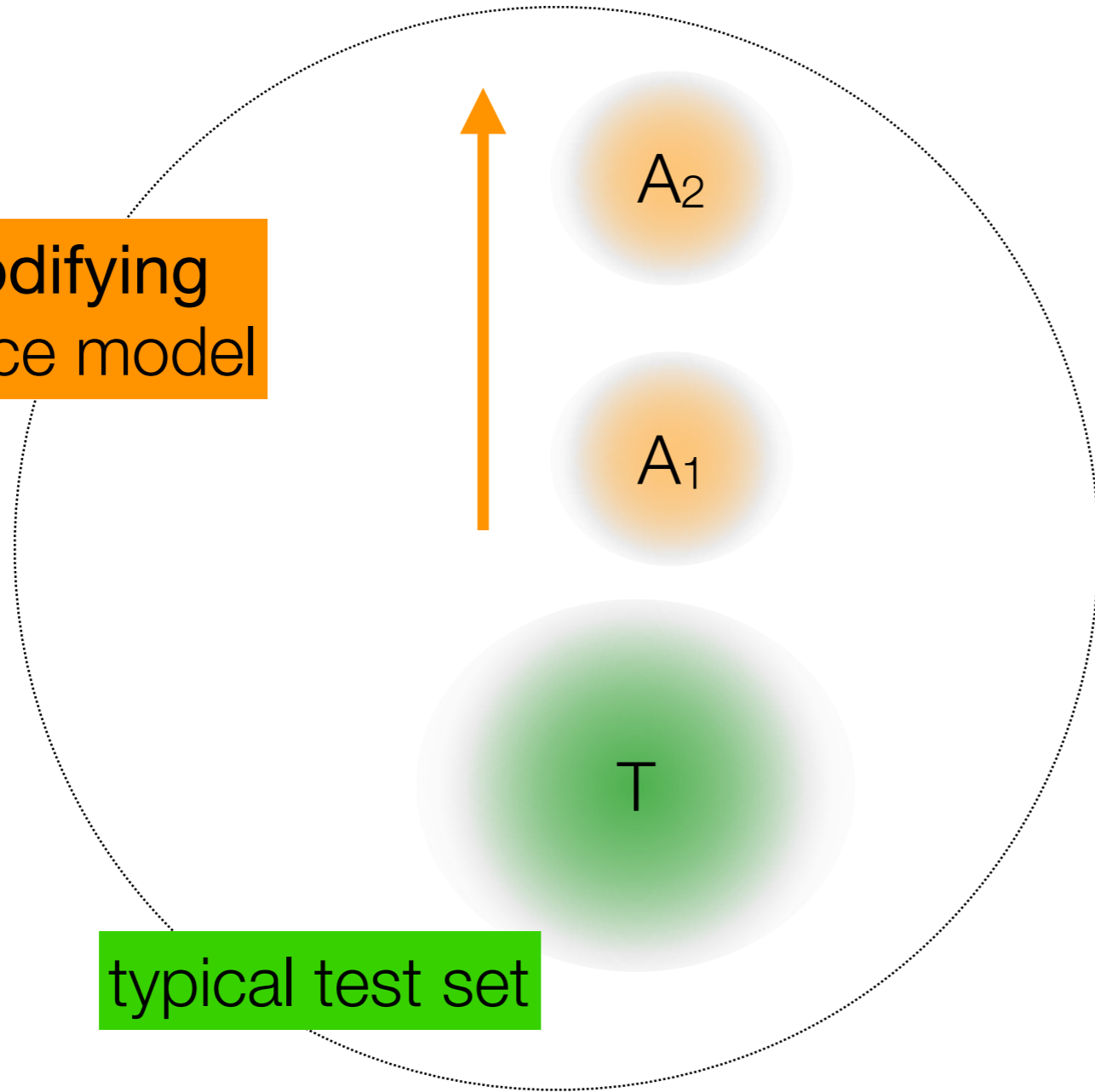


**JEuclid**

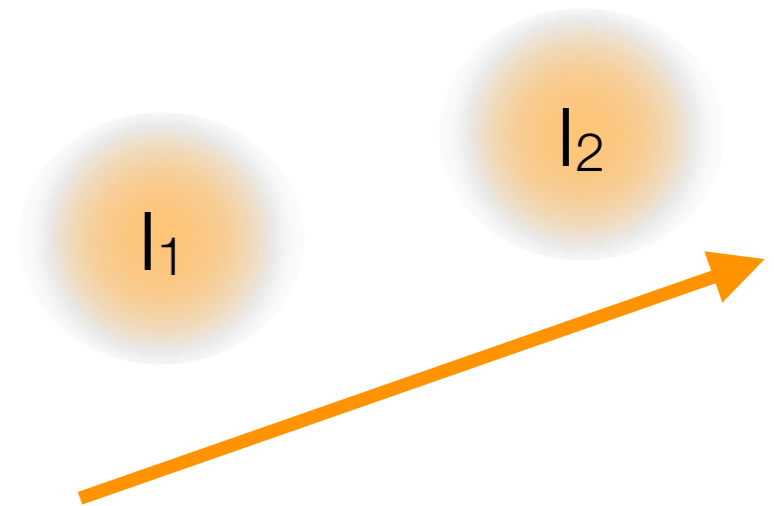


$$\frac{1 + \sqrt{5}}{2}$$

modifying  
choice model

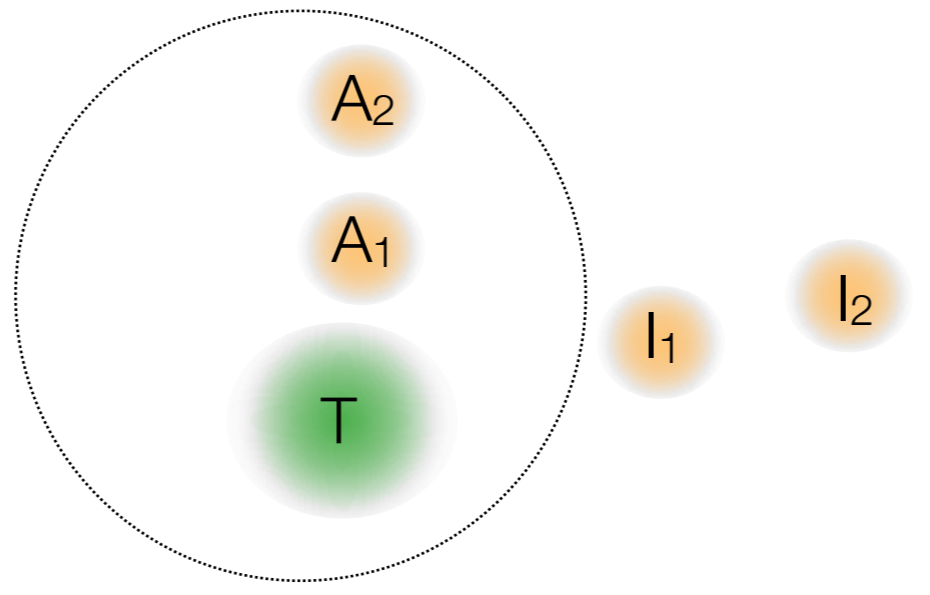
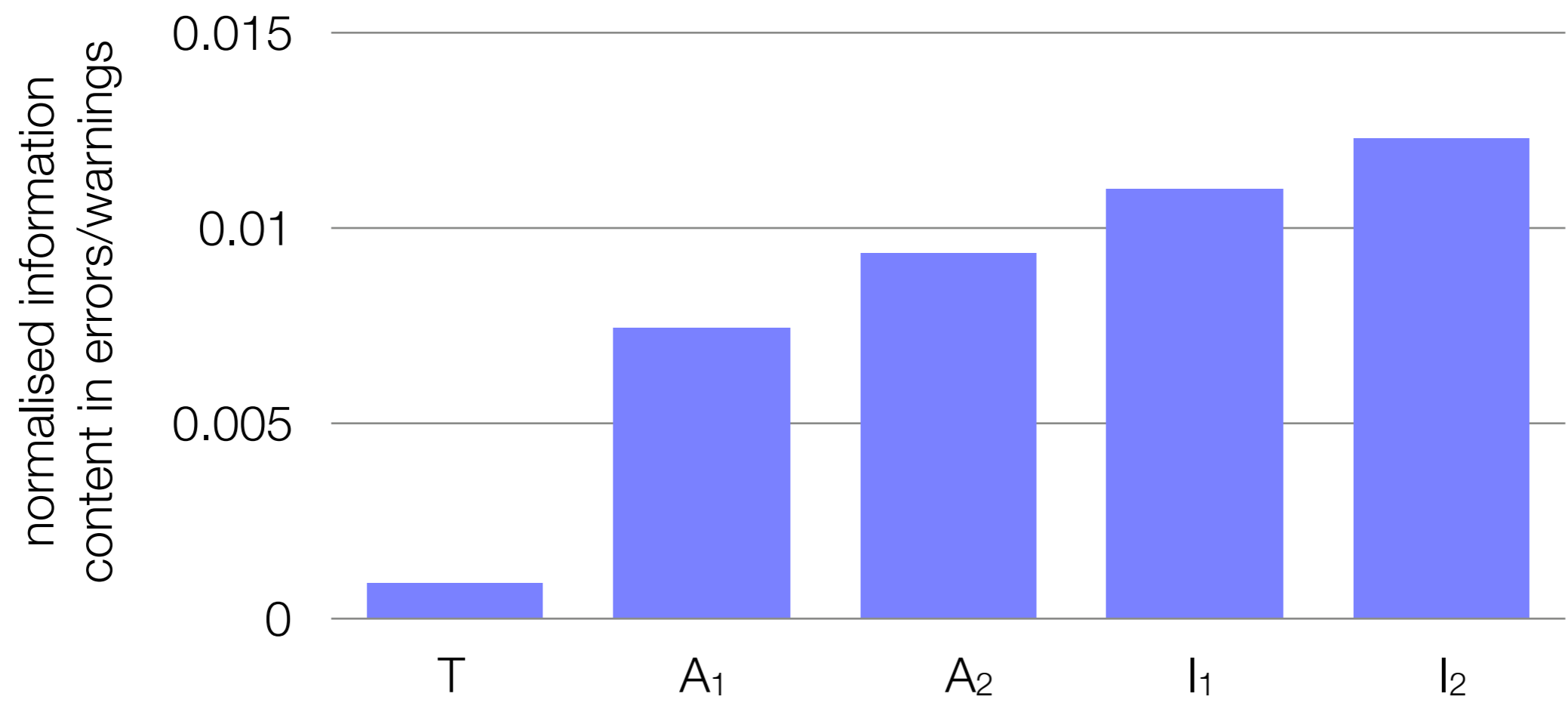


typical test set



mutating  
choice model





## Context

software testing: generating test data for software for which inputs are highly structured

## Problem

how to enable the test engineer to generate test data that is both valid and has desirable properties?

## Solution

- (1) allow the test engineer to specify the construction of valid test inputs using generators written in a general purpose language
- (2) apply metaheuristic and Monte-Carlo tree search algorithms to optimise the generation process for the desirable properties