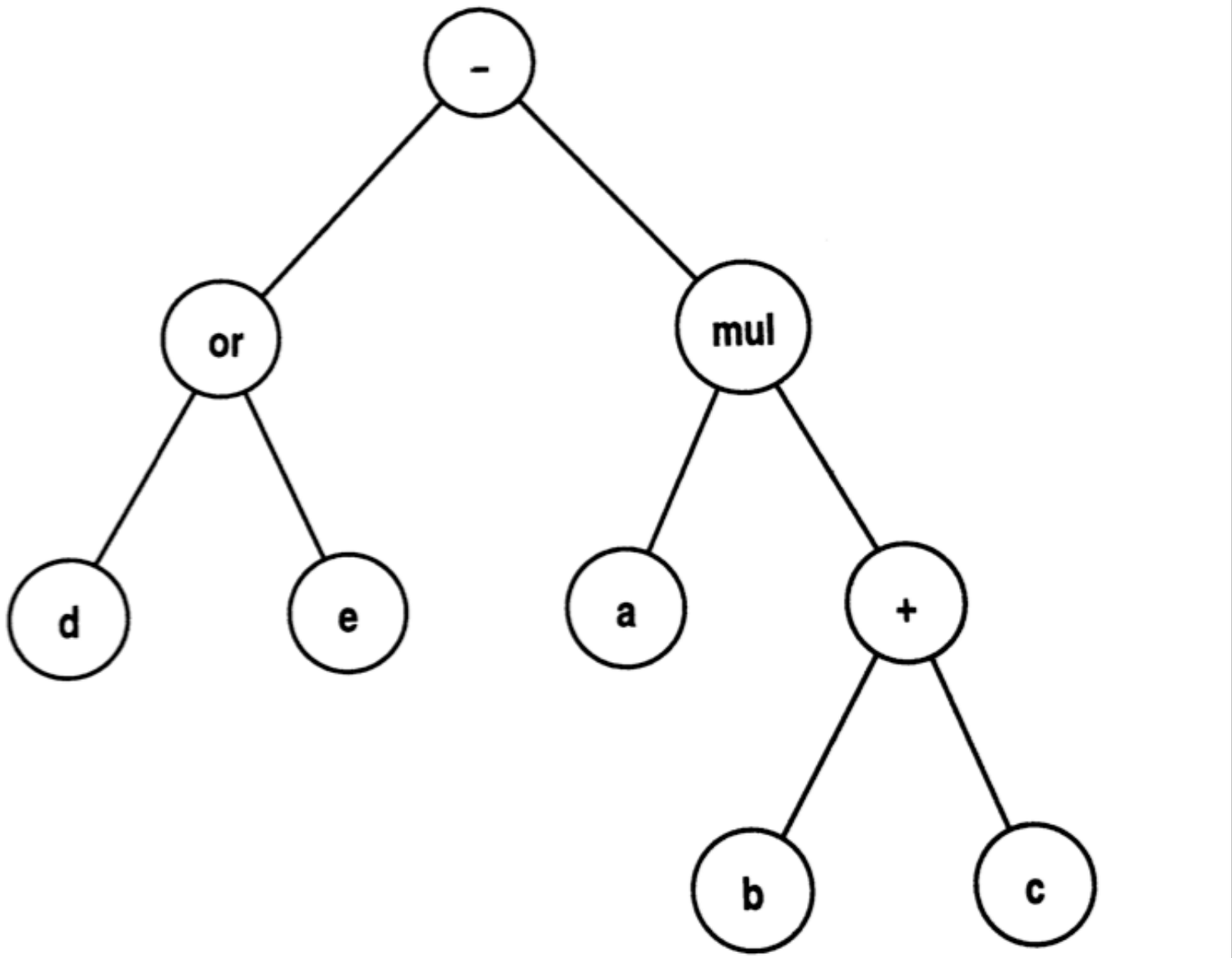
16: Genetic Programming 2

- Initialization of trees
- Mutation and crossover
- Fitness evaluation
- Population management
- Textbook chapter 6.4
- Reference book chapter 5
- Reference book: *Genetic Programming - An Introduction, Banzhaf et al, Morgan Kaufmann*

GP technical summary table

Representation	Tree structures
Recombination	Exchange of subtrees
Mutation	Random change in trees
Parent selection	Fitness proportional
Survivor selection	Generational replacement

An example GP tree



Choosing terminals and functions

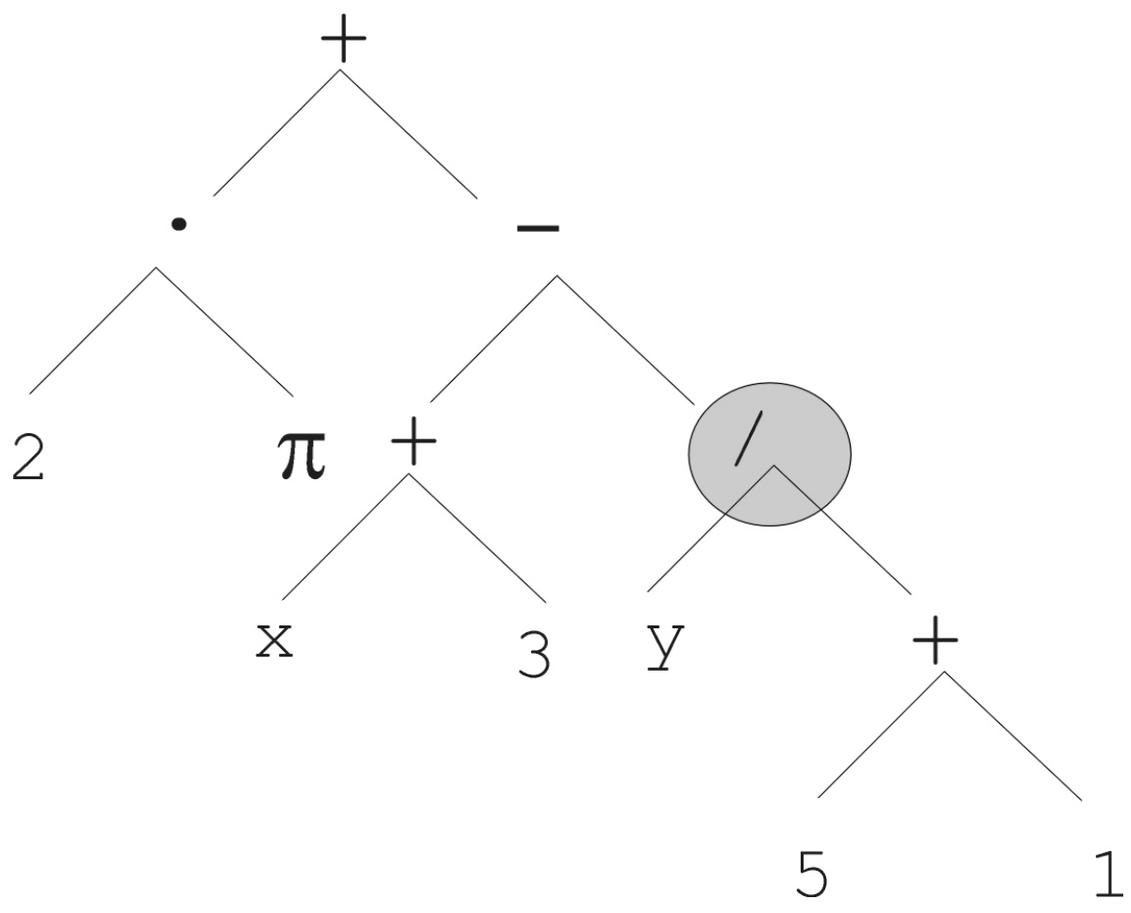
- Sufficiency and parsimony
    - for choosing both functions and constants
- Adopting the following general recursive definition:
    - every $t \in T$ is a correct expression
    - $f(e_1, \ldots, e_n)$ is a correct expression if $f \in F$, arity(F) = n and $e_1, \ldots e_n$ are correct expressions
    - There are no other forms of correct expressions
- *Closure* property: any function should be able to handle all values it might receive as inputs
    - e.g. division operator -> protected division

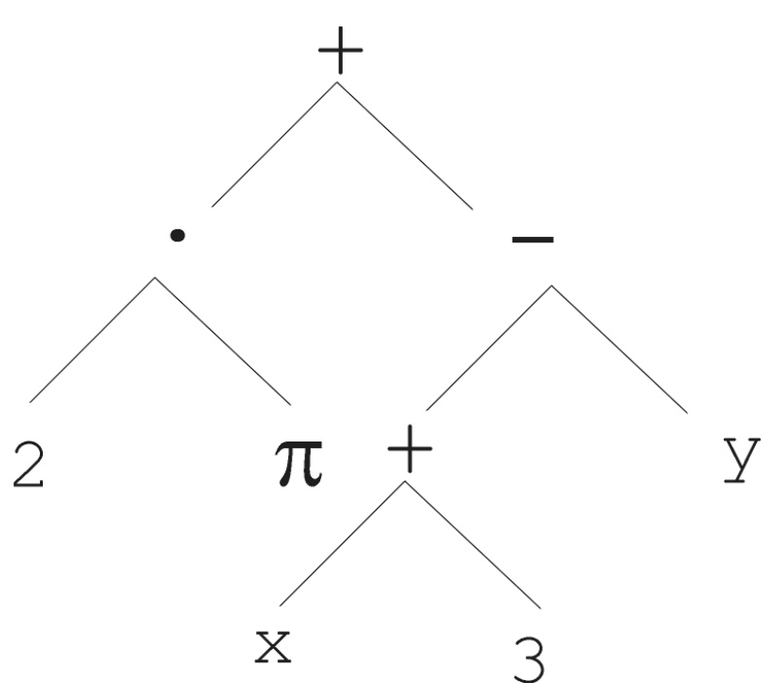
Initialization of GP trees

- Maximum initial depth of trees D_{max} is usually determined
- ***Full*** method (each branch has depth = D_{max}):
  - nodes at depth $d < D_{max}$ randomly chosen from function set F
  - nodes at depth $d = D_{max}$ randomly chosen from terminal set T
- ***Grow*** method (each branch has depth $\leq D_{max}$):
  - nodes at depth $d < D_{max}$ randomly chosen from $F \cup T$
  - nodes at depth $d = D_{max}$ randomly chosen from T
- Common GP initialization: *ramped half-and-half*
  - population is divided equally with trees having depths 2, 3, 4, …., and D_{max}
  - grow & full methods each deliver half of a depth group

Variation operator - mutation

- Most common mutation: replace randomly chosen subtree by randomly generated tree





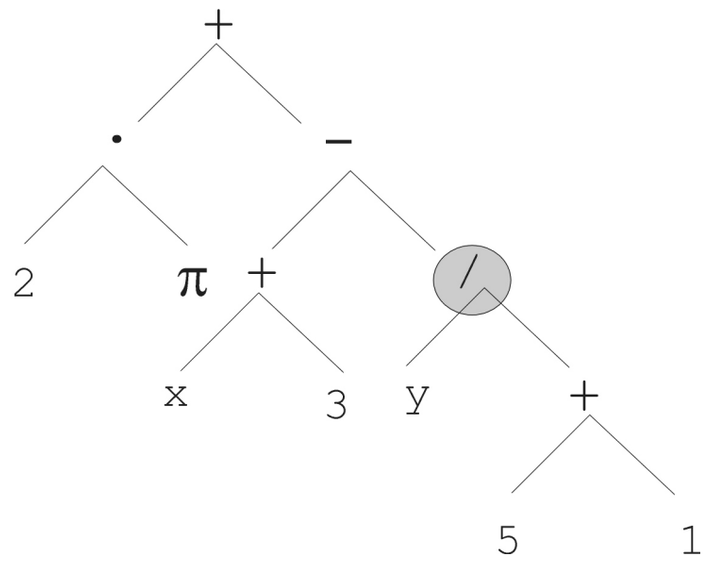
Mutation (cont'd)

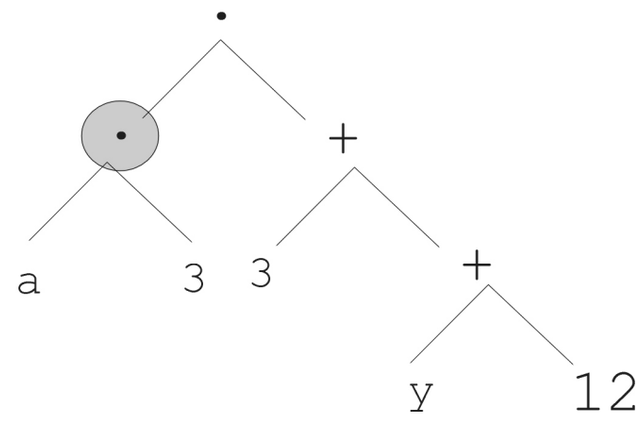
- Mutation has two parameters:
  - probability p_m to perform mutation
  - probability to chose an internal point as the root of the subtree to be replaced
- p_m is advised to be very small like 0.05
- The size of the child can exceed the size of the parent

Variation operator - recombination

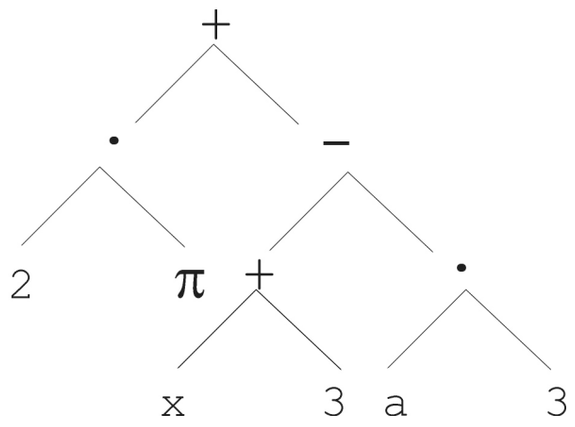
- Most common recombination: exchange two randomly chosen subtrees among the parents
- Recombination has two parameters:
  - probability p_c to perform recombination
  - probability to choose an internal point within each parent as crossover point
- The size of offspring can exceed that of the parents

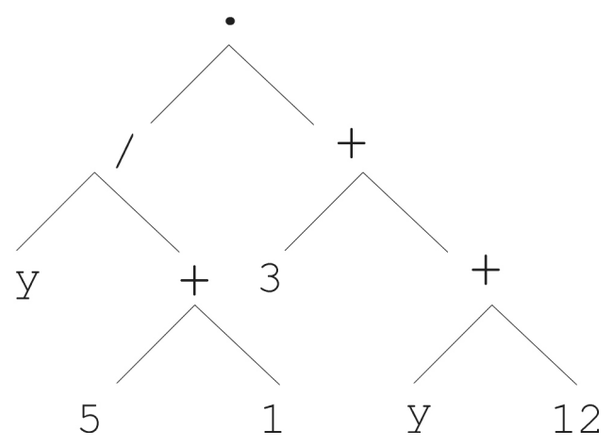
parents





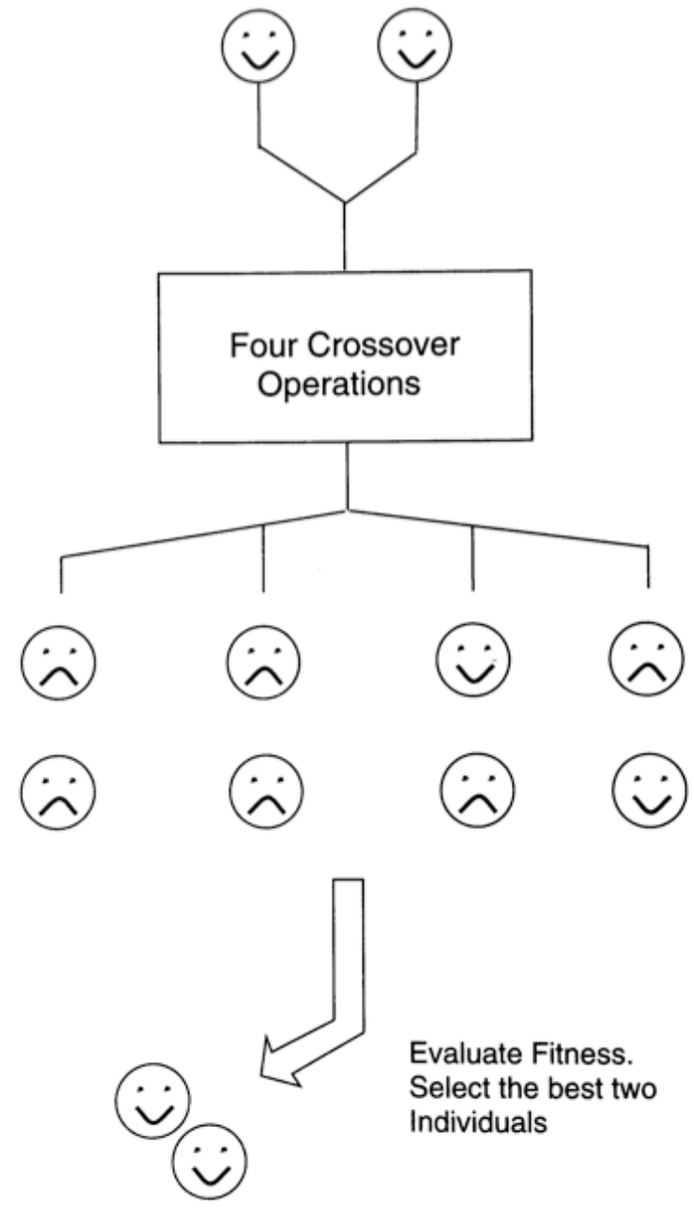
offspring





Improving crossover

- Brood recombination [Altenberg, 1994][Tackett, 1994]



Improving crossover

- Building block hypothesis
- Intelligent crossover
  - evolve the strategy on how to select crossover point [Teller and Veloso, 1995]
  - use heuristics to guide crossover by quantifying performance of subtrees [Iba and de Garis, 1996]
- Biological homologous crossover
  - context-sensitive crossover [D'haeseleer, 1994]
  - measure structural and functional similarities [Sankoff and Kruskal, 1983]

Fitness evaluation

- The measure used by GP during simulated evolution of how well a program has learned to predict the outputs from the inputs
- Designed to give graded and continuous feedback about how well a program performs on the training set
- Error-based fitness functions, squared errors or square root errors
- Standardized or normalized

An example

	Input	Output	Model Out	Error	Squared error
Training case 1	1	2	1	1	1
Training case 2	2	6	4	2	4
Training case 3	4	20	16	4	16
Training case 4	7	56	49	7	49
Training case 5	9	90	81	9	81
Fitness				23	151

Other fitness examples

- An image matching application
- A robot controlled by GP that learns obstacle avoidance
- In a classification task
- A GP-controlled agent in a betting game
- An artificial agent in an artificial life application
- In a game-playing application (cooperative or competing)

Selection and population mode

- Parent selection
  - traditionally fitness proportional
  - recently tournament selection (with steady-state population mode)

- Survivor selection and population mode
  - traditionally generational scheme
  - recently ES truncation selection (μ, λ) or $(\mu + \lambda)$
  - or EP $(\mu + \mu)$ with round-robin tournament with parameter *q*

Generational GP

1. Initialize the population
2. Evaluate the individual programs in the existing population
3. Until the new population is fully populated, repeat the following
    1. Select parent individuals
    2. Perform crossover and mutation
    3. Place offspring in the new population
4. If the termination criterion is fulfilled, the continue to the next step.

   Otherwise, replace the existing population with the new population and

   repeat step 2 to 4
5. Present the best individual in the population as the output from the algorithm

Steady-state GP

1. Initialize the population
2. Randomly choose a subset of individuals for tournament selection
3. Evaluate the fitness value of each competitor in the tournament
4. Select the winner(s) from the competitors
5. Apply crossover and mutation to the copies of the the winner(s)
6. Replace the losers in the tournament with the offspring
7. Repeat step 2 to 7 until the termination criterion is met
8. Choose the best individual in the population as the output from the algorithm

GP example: symbolic regression

- Target function:

$$y = f(x) = \frac{x^2}{2}$$

- GP configuration:
  - Terminal set: variable x, integer constants between -5 and +5
  - Function set: arithmetic functions +, -, *, %(protected)
  - Fitness function: root mean square error over 10 training cases
  - Parameter design: population size, initialization, crossover rate, mutation rate, selection methods

GP example: symbolic regression

- Target function:

$$y = f(x) = \frac{x^2}{2}$$

- Training set:

	Input	Output
Fitness Case 1	0.000	0.000
Fitness Case 2	0.100	0.005
Fitness Case 3	0.200	0.020
Fitness Case 4	0.300	0.045
Fitness Case 5	0.400	0.080
Fitness Case 6	0.500	0.125
Fitness Case 7	0.600	0.180
Fitness Case 8	0.700	0.245
Fitness Case 9	0.800	0.320
Fitness Case 10	0.900	0.405

GP example: symbolic regression

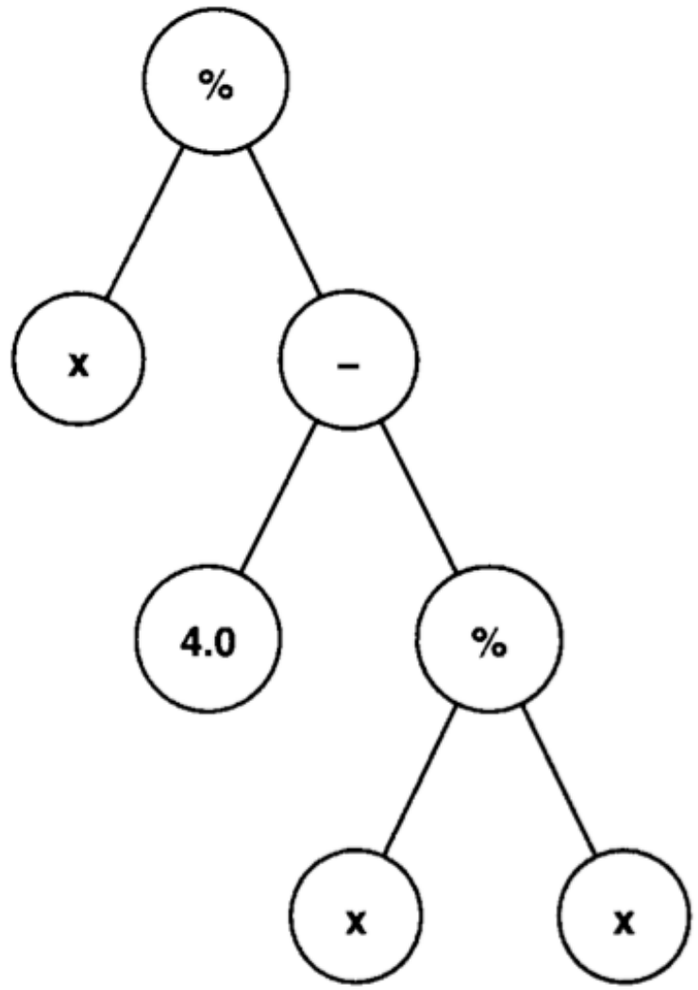
- Target function:

$$y = f(x) = \frac{x^2}{2}$$

Parameters	Values
objective:	evolve function fitting the values of the fitness case table
terminal set:	x, integers from -5 to +5
function set:	ADD, SUB, MUL, DIV
population size:	600
crossover probability:	90 percent
mutation probability:	5 percent
selection:	tournament selection, size 4
termination criterion:	none
maximum number of generations:	100
maximum depth of tree after crossover:	200
maximum mutant depth:	4
initialization method:	grow

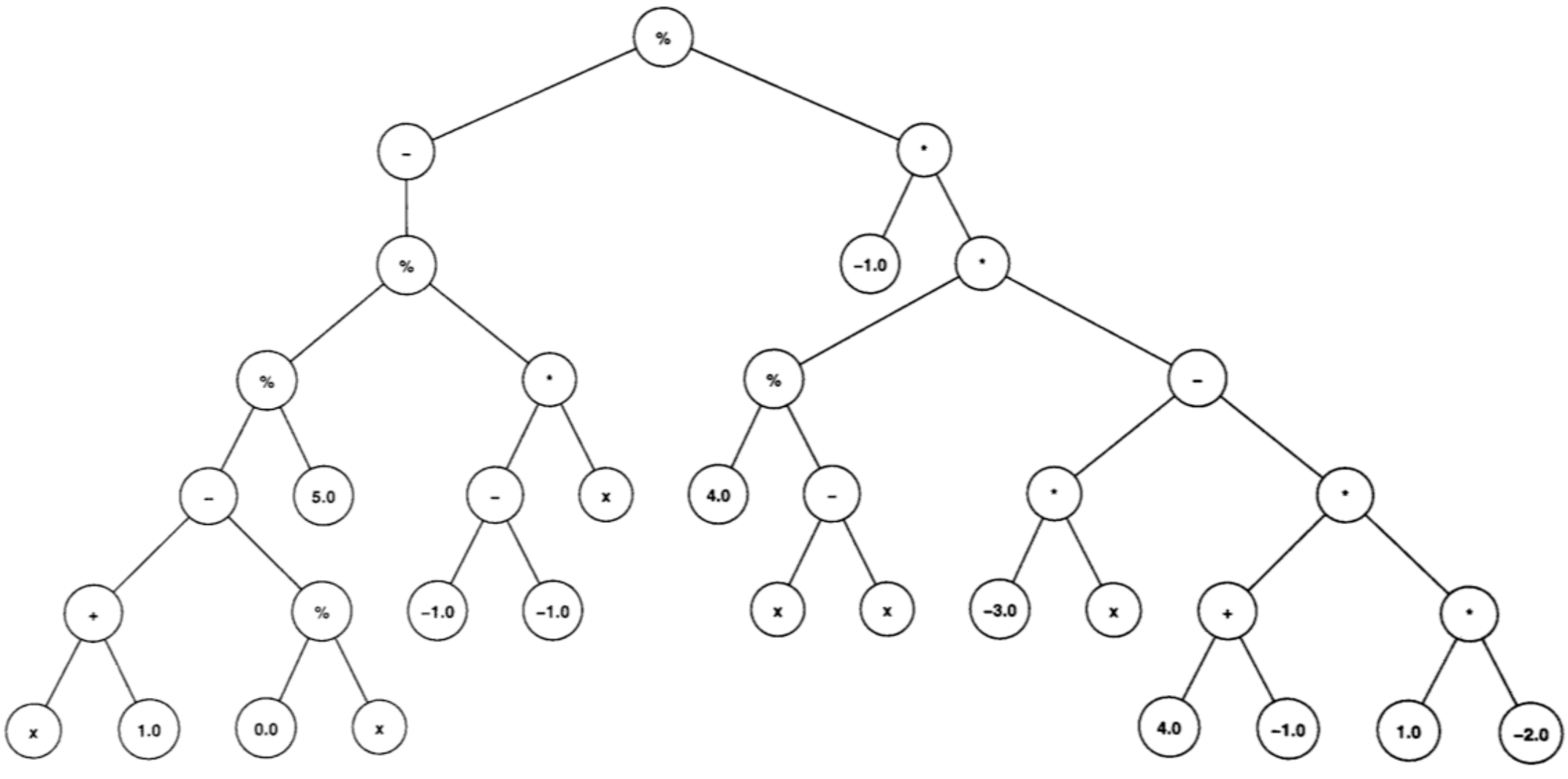
GP example: symbolic regression

- Best individual in generation 0



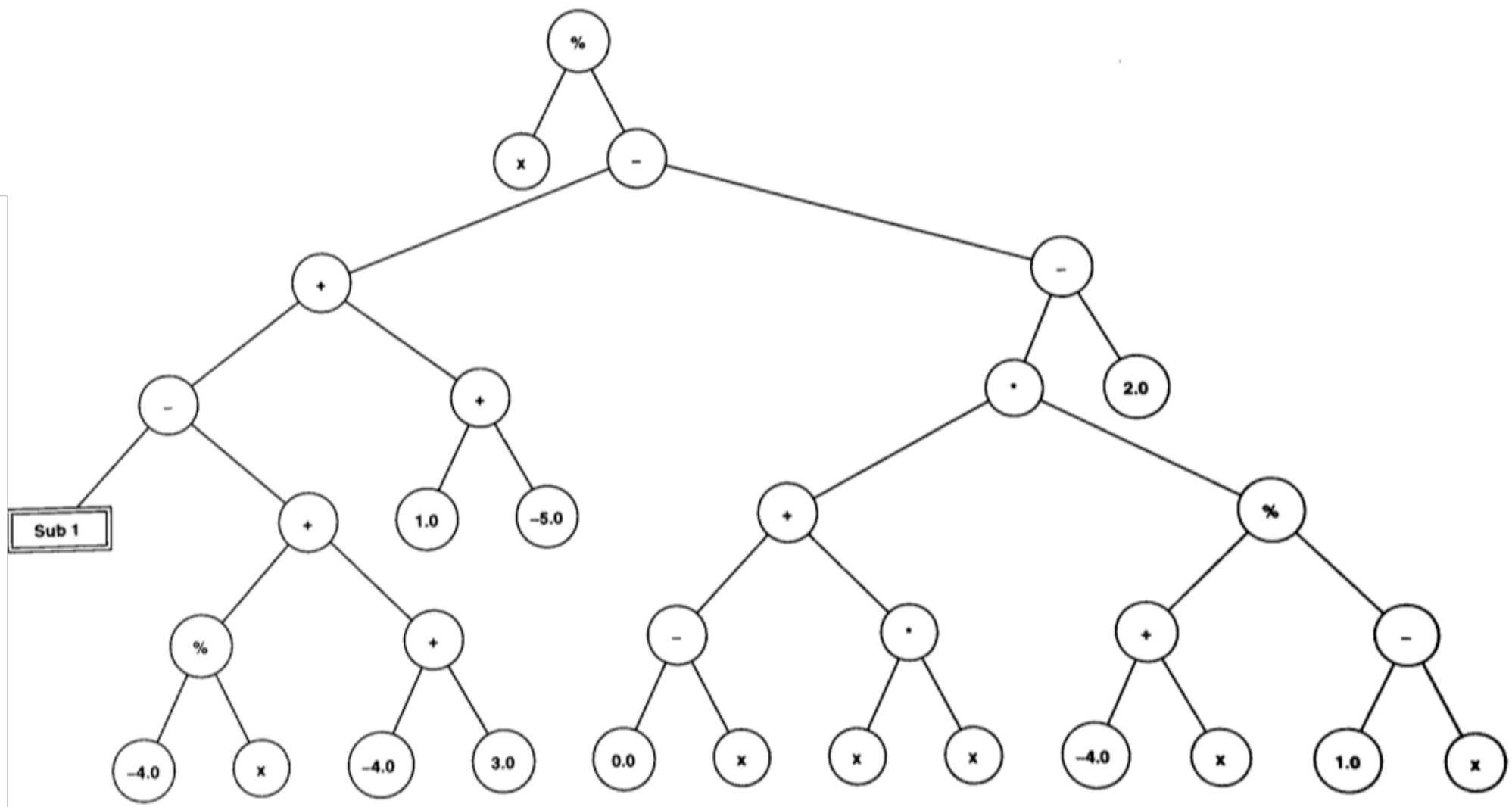
GP example: symbolic regression

- Best individual in generation 1



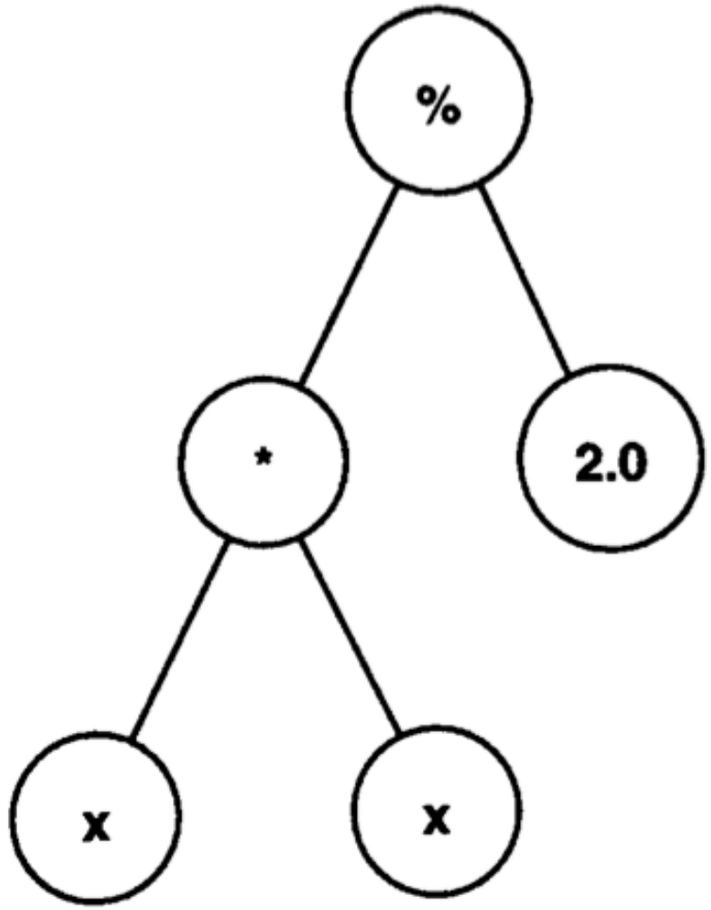
GP example: symbolic regression

- Best individual in generation 2



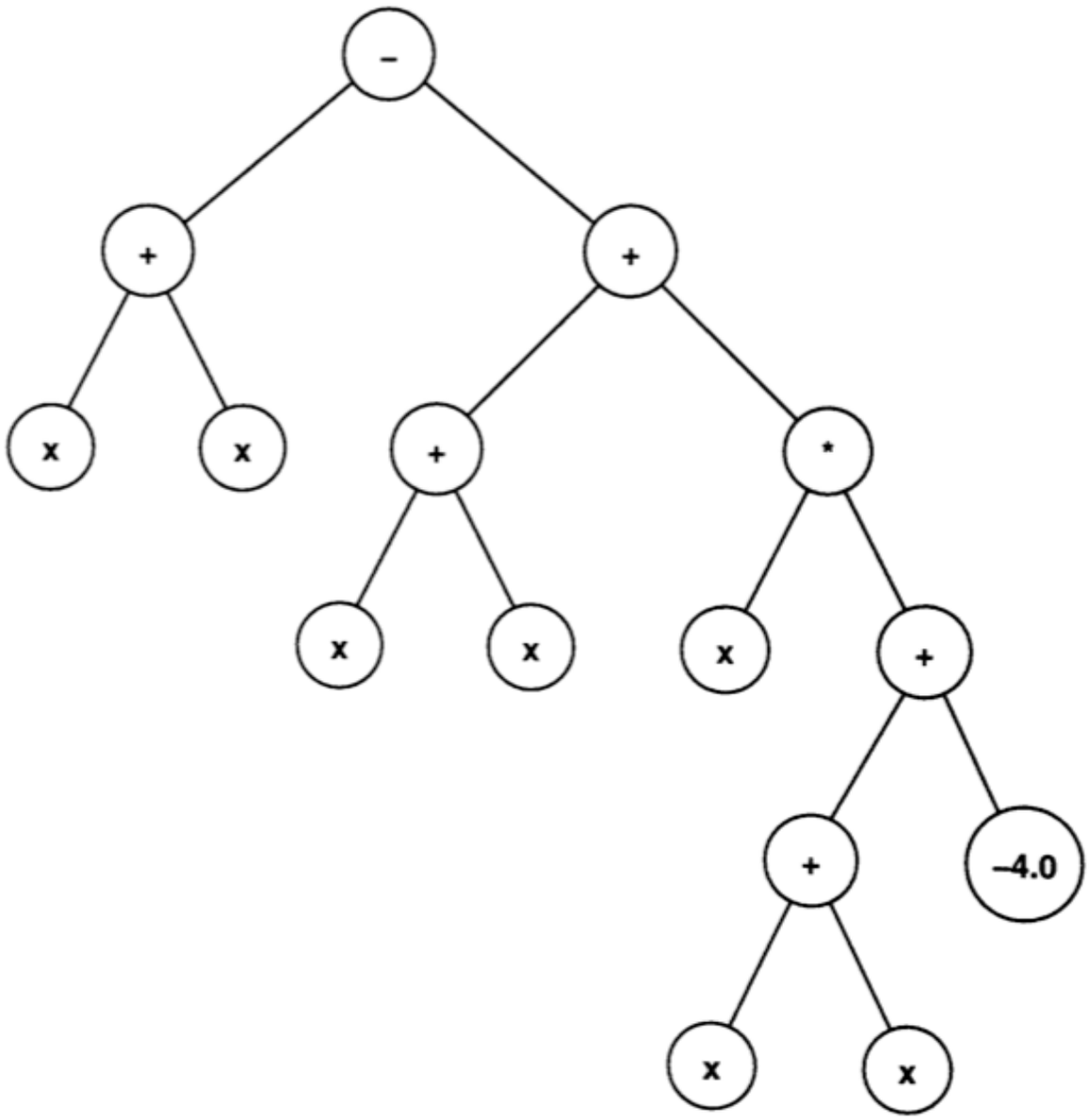
GP example: symbolic regression

- Best individual in generation 3



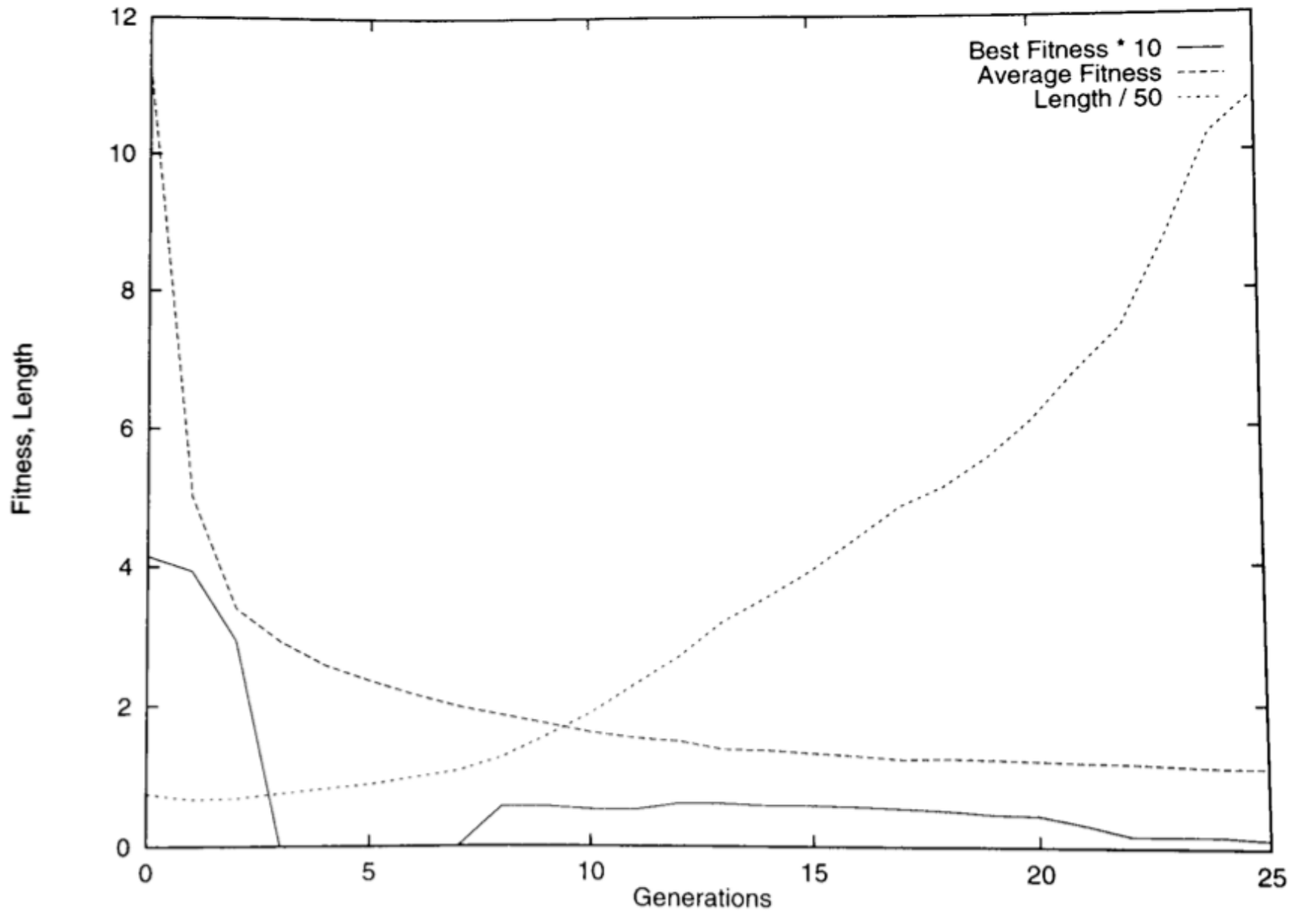
GP example: symbolic regression

- Best individual in generation 5



GP example: symbolic regression

- Fitness progression and tree depth growth



Bloat

- The tree sizes in the population are increasing over time
- Ongoing research and debate about the reasons
- Need countermeasures, e.g.
  - prohibiting variation operators that would deliver “too big” children
  - parsimony pressure: penalty for being oversized