Introduction to Program Synthesis (WS 2024/25)

Chapter 4 - Advanced Methodologies

Dr. rer. nat. Roman Kalkreuth

Chair for AI Methodology (**AIM**), Department of Computer Science,
RWTH Aachen University, Germany

Advanced Methodologies
Evolutionary Algorithms

- Population-based approach to search heuristics $\rightarrow$ Inspired by **biological evolution**
  - $\rightsquigarrow$ Adaptation of **Darwinian evolution** $\rightarrow$ Survival of the fittest
  - $\rightsquigarrow$ Consider the cost of a **candidate** as **fitness**
  - $\rightsquigarrow$ Evolutionary mechanism are performed within an evolutionary algorithms
- Transforming populations of candidate programs to better ones
  - $\rightsquigarrow$ *Better* $\rightarrow$ Better fitness
  - $\rightsquigarrow$ Apply selection mechanism to mimic natural selection
- Adapting **genetic variation** for **combinatorial search** $\rightarrow[\rightsquigarrow]$ Consider **recombination** and **mutation** as search operators
  - *Evolving* candidate solutions towards an optimum

Advanced Methodologies
Evolutionary Algorithms

- Various methodologies have been established in the field of **evolutionary computation**:
  - $\rightsquigarrow$ **Genetic Algorithms** (GA) $\rightarrow$ bitstring representation
  - $\rightsquigarrow$ **Evolutionary Strategies** (ES) $\rightarrow$ real-valued representation
  - $\rightsquigarrow$ **Genetic Programming** (GP) $\rightarrow$ data-structures (trees, graphs, ...)

1954	Barricelli: Evolutionary simulations
1960s - early 1970s	Rechenberg, Schwefel: Evolutionary strategies
same period of time	Fogel: Evolutionary programming
same period of time	Holland: Genetic algorithms
1980s	Forsyth, Cramer, Hicklin: Genetic programming

Advanced Methodologies
Evolutionary Algorithms

population	—	a set of individuals
species	—	individuals, which share common characteristics
candidate solution	—	member of the population, part of the search space
individual	—	a candidate, potential solution
breeding	—	the genetic adaption, variation procedure
parent	—	an individual selected for breeding
offspring	—	a candidate solution produced by variation
genotype	—	representation model of an individual, set of genes, vector of numbers
phenotype	—	expression, behavior of the genotype
chromosome	—	a set of genotypes
gene	—	a region of the genotype that encodes functionality
crossover	—	genetic operator, which combines genetic information of two or more parents
	—	to produce new offspring
mutation	—	genetic operator, which varies information on the genome of a individual
	—	(mostly according to a given probability distribution)
selection	—	procedure which choses genomes from a population for later breeding
fitness function	—	an objective function to assess and compare individuals by their fitness
fitness	—	a measurement of the individual's phenotype against the ideal functionality
fitness evaluation	—	a procedure to evaluate the fitness of each individual

Table: List of the important terms which are commonly used in the field of evolutionary algorithms.

Advanced Methodologies
Evolutionary Algorithms

Algorithm Example of a simple evolutionary algorithm

```
procedure Evolutionary Algorithm
    initialize(P)                                ▷ Initialize set of candidate solutions
    repeat                                       ▷ Until termination criteria not triggered
        Q ← breed(P)                             ▷ Breed new individuals with crossover and mutation
        Evaluate(Q)                              ▷ Evaluate the fitness of each individual
        if Q meets termination criterion then
            return Q
        end if
        P ← select(P,Q)                          ▷ Select high-fitness individuals
    until P meets termination criterion
    return P
end procedure
```

Advanced Methodologies
Evolutionary Algorithms

- **Evolutionary HC** → Hill-climbing with mutation-based search operator(s)
  - ⇝ Mutation → structural change in a chromosome
  - ⇝ Mutations aggregate across the genome of an individual randomly
- Adaption of mutation for tree-structures → **Subtree mutation**
  - ⇝ Selection of a mutation points by chance
  - ⇝ Exchange of the subsequent subtree with a randomly generated one

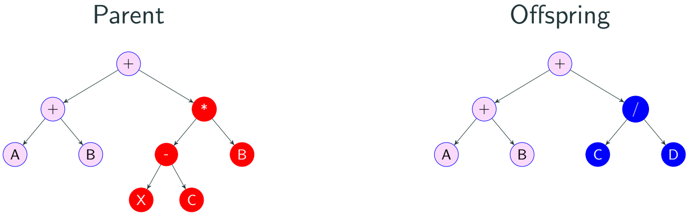

Figure: Subtree mutation

- $(1+\lambda)$ evolutionary strategy $\rightarrow$ Fundamental **search strategy** applied in the field of evolutionary strategies
  - $\leadsto$ Commonly used in numerical optimization approached with evolutionary computation
  - $\leadsto$ Breeds λ offspring from one parent in each generation
  - $\leadsto$ **Elitist** approach $\rightarrow$ **best partial solution** is **selected** from the **population** of $1+\lambda$ candidates
- Evolutionary algorithms that only uses mutation for **breeding** of new candidates
  - $\leadsto$ The $(1+1)$-ES is the most simple approach
- Adaption of HC to evolutionary computation

Advanced Methodologies
Evolutionary Algorithms

Algorithm $(1+\lambda)$ Evolutionary Strategy (ES)

Arguments

λ: Number of offspring

Return

$\mathcal{P}$: Parent individual with best fitness

initialize($\mathcal{P}$) ▷ *Initialize parent individual*
repeat ▷ *Until termination criteria not triggered*
  $\mathcal{Q} \leftarrow$ `breed`($\mathcal{P}$) ▷ *Breed λ offspring by mutation*
  `evaluate`($\mathcal{Q}$) ▷ *Evaluate the fitness of the offspring*
  $\mathcal{Q}^+ \leftarrow$ `better`($\mathcal{Q}$,$\mathcal{P}$) ▷ *Get individuals which have better fitness as the parent*
  ▷ *If there exist individuals with better fitness*
  if $|\mathcal{Q}^+| > 0$ **then**
    $\mathcal{P} \leftarrow$ `best`($\mathcal{Q}$) ▷ *Assign the best offspring as parent*
  end if
until $\mathcal{P}$ *meets termination criterion*
return $\mathcal{P}$ ▷

Advanced Methodologies
Evolutionary Algorithms

- Adaption of recombination to tree structures → **Subtree crossover**
- Exchanges subtree's between selected *parents* that have high fitness → Adaption of **propagation of traits** from generation to generation
- Creates one or two offspring → Recombination of *genetic material*
    - ⇝ Genetic material → Composition of nodes and terminals
    - ⇝ Genes → Non-terminal or terminal symbol

- $(\mu + \lambda)$ strategy $\rightarrow$ Extension of the $(1 + \lambda)$ strategy to recombination
- A set of best μ individuals is formed
  - Each parent is selected uniformly at random
  - λ offspring are bred by subtree crossover and mutation
- μ parents $+$ λ offspring then form the new population
  - $\rightsquigarrow$ Level of elitism can be controlled by setting of $\mu \rightarrow$ selection pressure

Advanced Methodologies
Evolutionary Algorithms

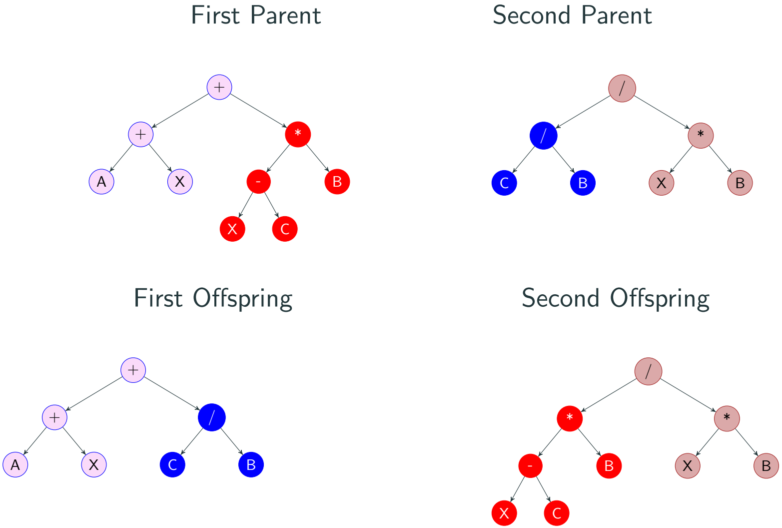

Figure: Subtree crossover

Advanced Methodologies
Evolutionary Algorithms

Algorithm $(\mu + \lambda)$ Evolutionary Strategy (ES)

Arguments
μ: Number of parents
λ: Number of offspring
p_c: Recombination probability
p_m: Mutation probability
Return
$\mathcal{B}$: Best individual

`initialize`$(\mathcal{P})$ ▷ *Initialize μ parents*
`evaluate`$(\mathcal{P})$ ▷ *Evaluate the fitness of the parents*
repeat ▷ *Until termination criteria not triggered*
  $\mathcal{Q} \leftarrow$ `selection`$(\mathcal{P}, \mu)$ ▷ *Select μ parents*
  $\mathcal{Q} \leftarrow$ `recombination`$(\mathcal{P}, \lambda, p_c)$ ▷ *Create λ offspring*
  $\mathcal{Q} \leftarrow$ `mutation`$(\mathcal{Q}, p_m)$ ▷ *Mutate the offsprings*
  `evaluate`$(\mathcal{Q})$ ▷ *Evaluate the fitness of the offspring*
  $\mathcal{P} \leftarrow \mathcal{P} + \mathcal{Q}$ ▷ *Form population of next generation*
  $\mathcal{B} \leftarrow$ `best`$(\mathcal{P})$ ▷ *Determine best individual*
until $\mathcal{B}$ *meets termination criterion*
return $\mathcal{B}$ ▷