Population-Based Metaheuristics 1

Ch. 3.1-3.2

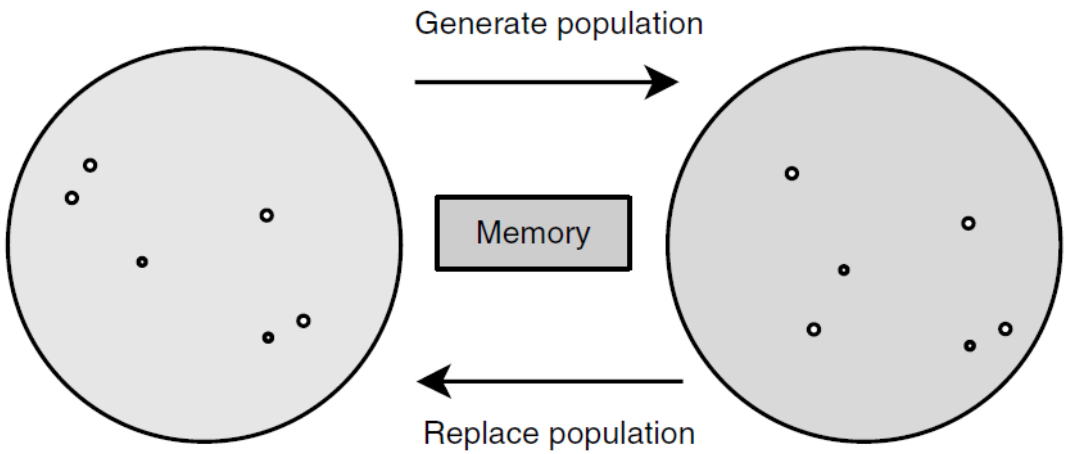
Marcus Posada

LINKÖPING UNIVERSITY

Common concepts

Ch. 3.1

Common concepts



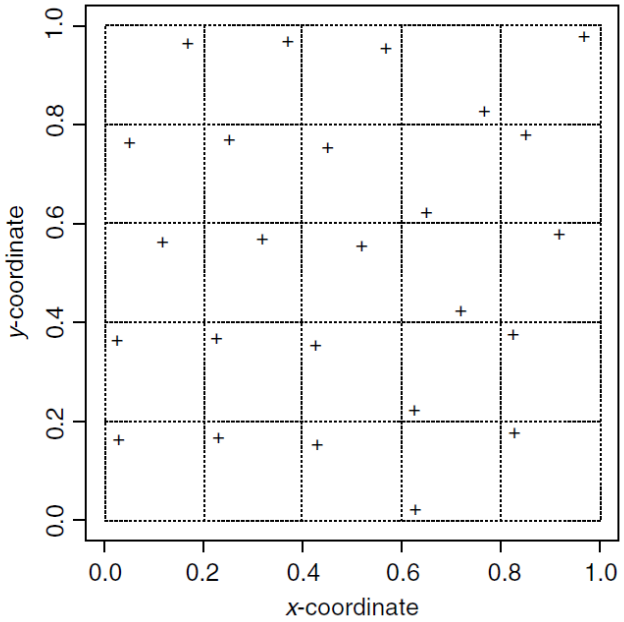
Algorithm 3.1 High-level template of P-metaheuristics.

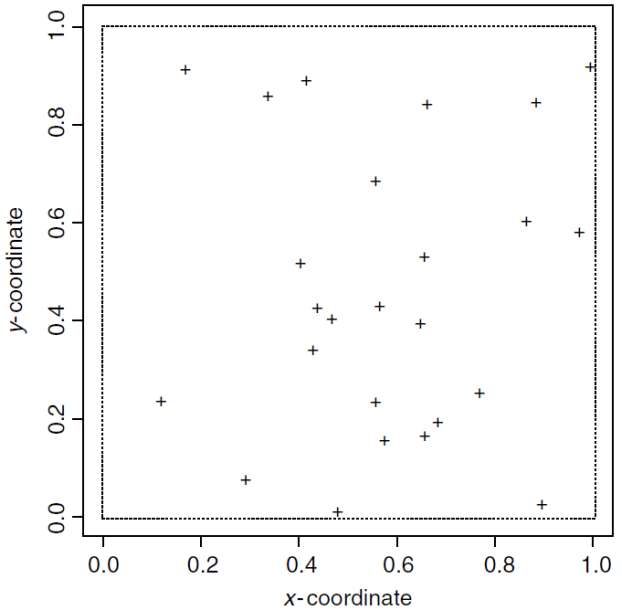
$P = P_0$; /* Generation of the initial population */
$t = 0$;
Repeat
  Generate(P'_t); /* Generation a new population */
  P_{t+1} = Select-Population($P_t \cup P'_t$); /* Select new population */
  $t = t + 1$;
Until Stopping criteria satisfied
Output: Best solution(s) found.

Initial population

- Pseudo- or Quasi-random
- Sequential or Parallel diversification
- Heuristic

Initial population: parallel diversification





Initial population

- Pseudo- or Quasi-random
- Sequential or Parallel diversification
- Heuristic

Initial population

Strategy	Diversity	Computational Cost	Quality of Initial Solutions
Pseudo-random	++	+++	+
Quasi-random	+++	+++	+
Sequential diversification	++++	++	+
Parallel diversification	++++	+++	+
Heuristic	+	+	+++

Stopping criteria

•Static: *a priori* knowledge of when the search will end. E.g. a maximum number of iterations.

•Adaptive: no *a priori* knowledge. E.g. number of iterations without improvement.

•There are also criteria which are specific to P-heuristics.

Evolutionary algorithms

Ch. 3.2

Evolutionary algorithms

- Genetic algorithms
- Evolution strategies
- Evolutionary programming
- Genetic programming

Evolutionary computation

- They are all inspired by the prinicple of heredity from parents to offspring in natural evolution.

Evolutionary algorithms

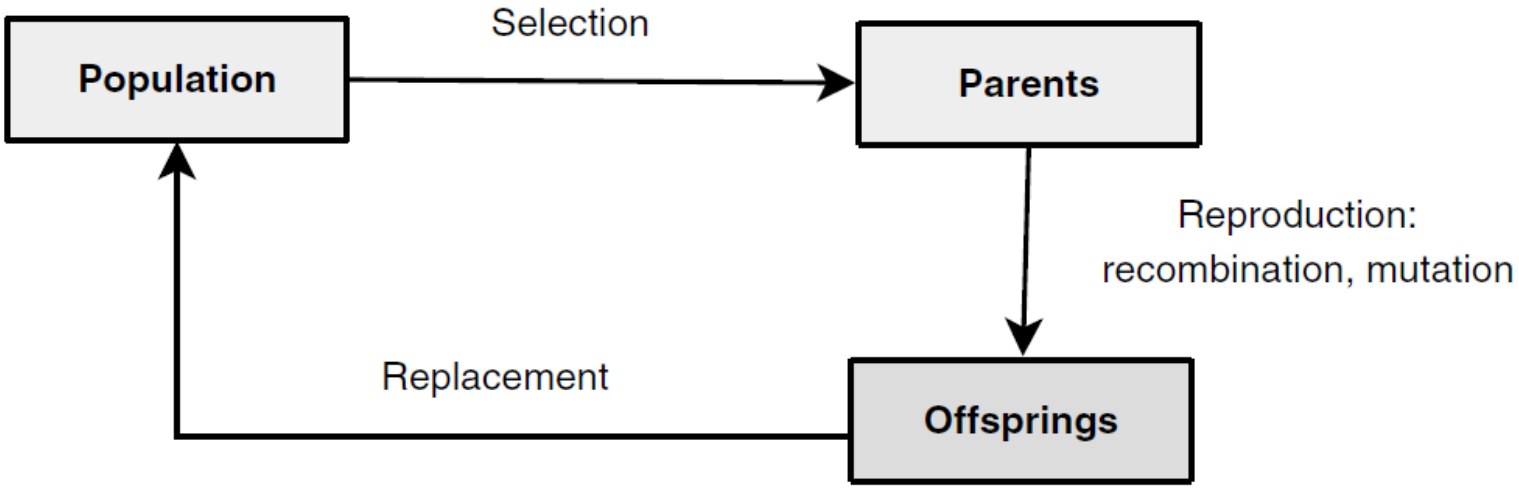


FIGURE 3.7 A generation in evolutionary algorithms.

Evolutionary algorithms

Metaphor	Optimization
Evolution	Problem solving
Individual	Solution
Fitness	Objective function
Environment	Optimization problem
Locus	Element of the solution
Allele	Value of the element (locus)

Algorithm 3.2 Template of an evolutionary algorithm.

```
Generate(P(0)) ; /* Initial population */
t = 0 ;
While not Termination_Criterion(P(t)) Do
   Evaluate(P(t)) ;
   P'(t)      = Selection(P(t)) ;
   P'(t)      = Reproduction(P'(t)); Evaluate(P'(t)) ;
   P(t + 1)   = Replace(P(t), P'(t)) ;
   t = t + 1 ;
End While
Output Best individual or best population found.
```

Genetic algorithms

- J. Holland 1970, USA. Applied to optimization in the 1980s.

- Classical characteristics: binary representation. Crossover + mutation. Parents are replaced by offspring.

Evolution strategies

•Rechenberg and Schewefel, 1964, Berlin. Used in continuous optimization.

•Classical characteristics: real-valued representation. Elitist replacement (offspring do not necessarily replace parents). Deterministic selection (fitness). No single step size in mutation.

•Efficient in terms of time complexity.

Evolution strategies

Algorithm 3.3 Template of evolution strategies.

Initialize a population of μ individuals ;
Evaluate the μ individuals ;
Repeat
  Generate λ offsprings from μ parents ;
  Evaluate the λ offsprings ;
  Replace the population with μ individuals from parents and offsprings ;
Until Stopping criteria
Output Best individual or population found.

Evolutionary programming

- L. Fogel, 1960s, USA.

- Classical characteristics: real-valued representation. No recombination. Normally distributed mutations. Deterministic parent selection, stochastic replacement selection (*n* species, tournament selection).

- Less used then the other families of EAs.

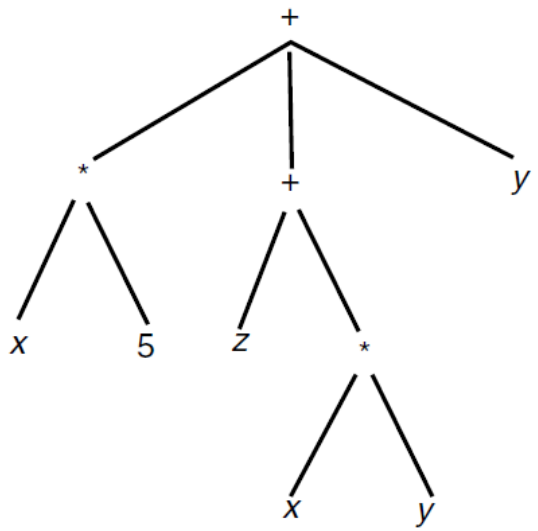
Genetic programming

- J Koza, 1980s. The individuals are themselves programs (not fixed length strings, as in the other methods).

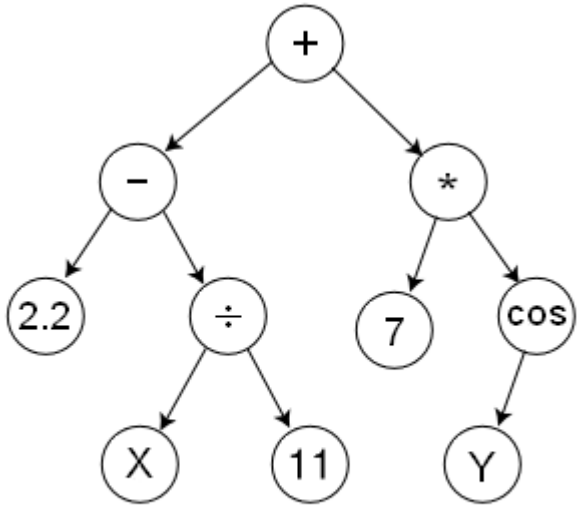
- Classical characteristics: deterministic parent selection, generational replacement. Crossover: sub-tree exchange. Mutation: random changes in the tree.

- Computationally intensive. Used for data mining.

Genetic programming: symbolic regression



$(x*5)+(z+(x*y))+y$



$$\left(2.2-\left(\frac{X}{11}\right)\right)+\left(7*\cos(Y)\right)$$

Reading advise

- Sörensen, Kenneth. "Metaheuristics—the metaphor exposed." *International Transactions in Operational Research* 22.1 (2015): 3-18.

Marcus Posada

www.liu.se

LINKÖPING UNIVERSITY