IE552

Lecture Notes 5 Genetic Algorithm

Population-based metaheuristics

- Population-based metaheuristics (P-metaheuristics) share many common concepts.
- They could be viewed as an iterative improvement in a population of solutions.
- First, the population is initialized. Then, a new population of solutions is generated.
- Finally, this new population is integrated into the current one using some selection procedures.
- The search process is stopped when a given condition is satisfied (stopping criterion).
- Most of the P-metaheuristics are nature-inspired algorithms. Popular examples of P-metaheuristics are evolutionary algorithms, ant colony optimization, scatter search, particle swarm optimization, bee colony, and artificial immune systems.

Population-based metaheuristics

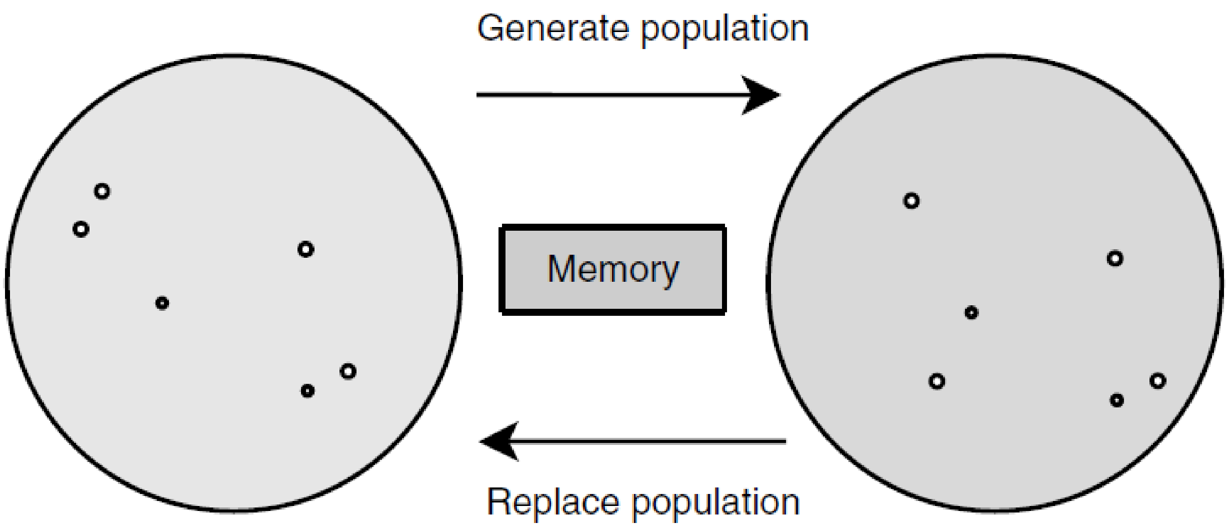

FIGURE 3.1 Main principles of P-metaheuristics.

Population-based metaheuristics

- Population-based metaheuristics start from an initial population of solutions. (some may start with empty or partial set)
- Then, they iteratively apply the generation of a new population and the replacement of the current population
- In the generation phase, a new population of solutions is created.
- In the replacement phase, a selection is carried out from the current and the new populations.
- This process iterates until a given stopping criteria.
- The generation and the replacement phases may be *memoryless*. In this case, the two procedures are based only on the current population. Otherwise, some history of the search stored in a memory can be used in the generation of the new population and the replacement of the old population.

Template of P-metaheuristics

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.

Population-based metaheuristics

- P-metaheuristics differ in the way they perform the generation and the selection procedures and the search memory they are using during the search.
- In evalutionary algorithms
  - The Search Memory: limited to the population of solutions.
  - Generation: the solutions composing the population are selected and reproduced using variation operators (e.g., mutation, recombination, also called crossover and merge)
  - Selection: The traditional strategy consists in selecting the generated population as the new population. Other strategies use some *elitism* in the selection phase where they provide the best solutions from the two sets.

Evolutionary Algorithms

- In the nineteenth century, J. Mendel was the first to state the baselines of heredity from parents to offsprings.
- Then in 1859, C. Darwin presented the theory of evolution in his famous book *On the Origin of Species*
- In the 1980s, these theories of creation of new species and their evolution have inspired computer scientists in designing evolutionary algorithms.
- Different main schools of evolutionary algorithms have evolved independently during the past 40 years
- *Genetic algorithms* (GA), mainly developed in Michigan, USA, by J. H. Holland.

Evolutionary Algorithms

- Evolutionary algorithms are stochastic P-metaheuristics that have been successfully applied to many real and complex problems.
- They are the most studied population-based algorithms.
- Their success in solving difficult optimization problems in various domains (continuous or combinatorial optimization, system modeling and identification, planning and control, engineering design, data mining and machine learning, artificial life) has promoted the field known as *evolutionary computation.*

Evolutionary Algorithms

- EAs are based on the notion of *competition*.
- They represent a class of iterative optimization algorithms that simulate the evolution of species.
- They are based on the evolution of a population of individuals.
- Initially, this population is usually generated randomly.
- Every individual in the population is the encoded version of a tentative solution.
- An objective function associates a fitness value with every individual indicating its suitability to the problem.

Evolutionary Algorithms

- At each step, individuals are selected to form the parents, following the selection paradigm in which individuals with better fitness are selected with a higher probability.
- Then, selected individuals are reproduced using variation operators (e.g., crossover, mutation) to generate new offsprings.
- Finally, a replacement scheme is applied to determine which individuals of the population will survive from the offsprings and the parents.
- This iteration represents a generation.

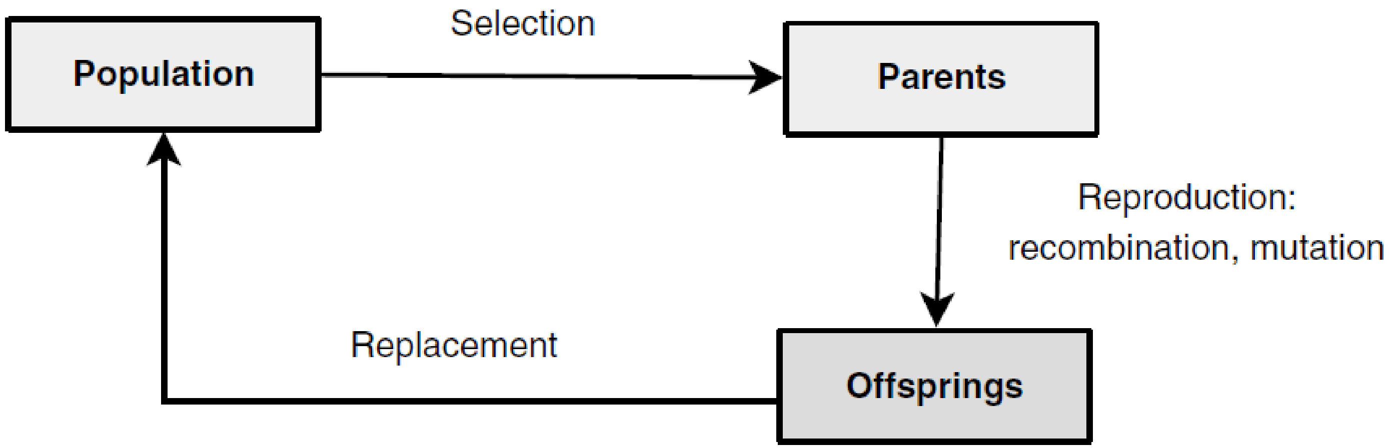

FIGURE 3.7 A generation in evolutionary algorithms.

Evolutionary Algorithms

TABLE 3.3 Evolution Process Versus Solving an Optimization Problem

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

Evolutionary Algorithms

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.

Evolutionary Algorithms

- In evolutionary algorithms, the genotype represents the encoding while the phenotype represents the solution.
- Hence, the genotype must be decoded to generate the phenotype.
- The variation operators act on the genotype level while the fitness function will use the phenotype of the associated individual
- The fitness of an individual measures the ability of the individual to survive in its environment.
- In the case where a direct encoding is used, the genotype is similar to the phenotype.
- Otherwise (i.e., if an indirect encoding is used), the genotype and the phenotype are different structures. Indeed, a decoding function is used to transform the genotype into a phenotype.

Genetic Algorithm

- Genetic algorithms have been developed by J. Holland in the 1970s (University of Michigan, USA) to understand the adaptive processes of natural systems.
- Then, they have been applied to optimization and machine learning in the 1980s.
- GAs are a very popular class of EAs.
- Traditionally, GAs are associated with the use of a binary representation but nowadays one can find GAs that use other types of representations.
- A GA usually applies a crossover operator to two solutions that plays a major role, plus a mutation operator that randomly modifies the individual contents to promote diversity.

Genetic Algorithm

- GAs use a probabilistic selection that is originally the proportional selection.
- The replacement (survivor selection) is generational, that is, the parents are replaced systematically by the offsprings.
- The crossover operator is based on the *n-point or uniform crossover* while the mutation is bit flipping.
- A fixed probability p_m (resp. p_c) is applied to the mutation (resp. crossover) operator.

Examples

- https://youtu.be/uQj5UNhCPuo
- https://www.youtube.com/watch?v=XP8R0yzAbdo&ab_channel=FullstackAcademy