
Chapter 3: Genetic Algorithm

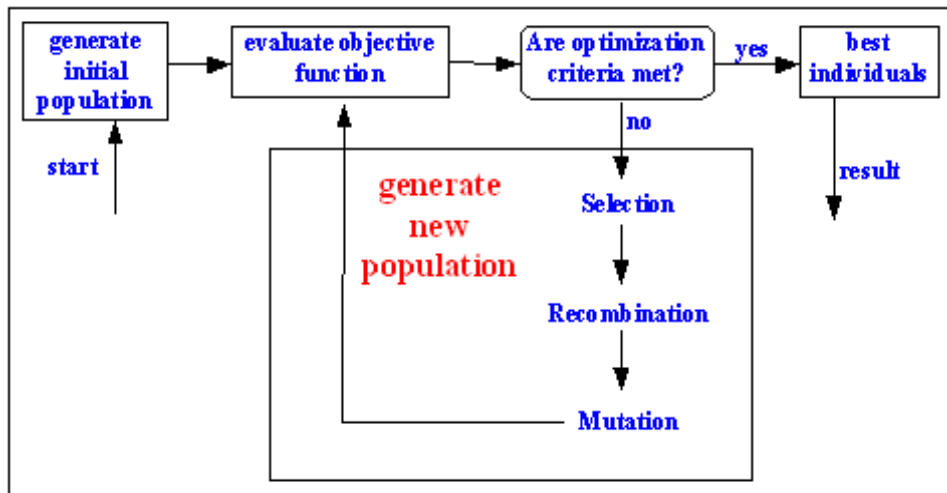
16-28

3.1 Introduction

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

Evolutionary algorithms[11] model natural processes, such as selection, recombination, mutation, migration, locality and neighbourhood. Figure shows the structure of a simple evolutionary algorithm. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner.

Fig. 3-1: Structure of a single population evolutionary algorithm

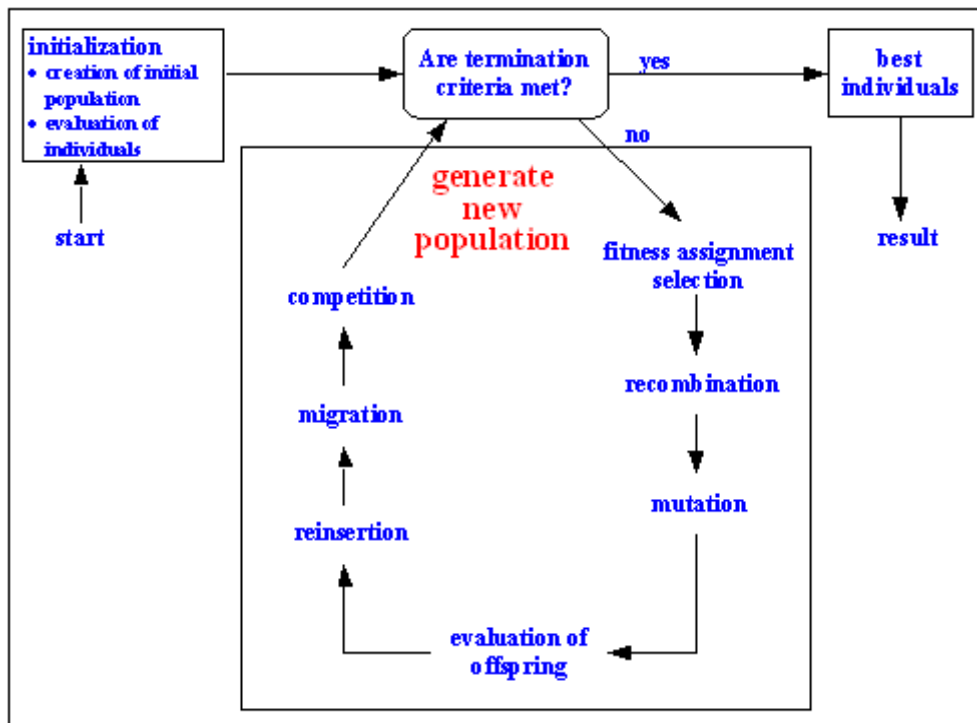


At the beginning of the computation a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals. The first/initial generation is produced.

If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

Such a single population evolutionary algorithm is powerful and performs well on a wide variety of problems. However, better results can be obtained by introducing multiple subpopulations. Every subpopulation evolves over a few generations isolated (like the single population evolutionary algorithm) before one or more individuals are exchanged between the subpopulation. The multi-population evolutionary algorithm[12] models the evolution of a species in a way more similar to nature than the single population evolutionary algorithm. Figure shows the structure of such an extended multi-population evolutionary algorithm.

Fig. 3-2: Structure of an extended multipopulation evolutionary algorithm



From the above discussion, it can be seen that evolutionary algorithms differ substantially from more traditional search and optimization methods[13]. The most significant differences are:

- Evolutionary algorithms search a population of points in parallel, not just a single point.
- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms are generally more straightforward to apply, because no restrictions for the definition of the objective function exist.
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user. (Thus, in cases where the particular problem does not have one individual solution, for example a family of pareto-optimal[14] solutions, as in the case of multi-objective optimization and scheduling problems, then the evolutionary algorithm is potentially useful for identifying these alternative solutions simultaneously.)

The following sections list some methods and operators of the main parts of Evolutionary Algorithms. A thorough explanation of the operators will be given in the following chapters.

3.2 Selection

Selection determines which individuals are chosen for mating (recombination) and how many offspring each selected individual produces. The first step is fitness assignment by:

- proportional fitness assignment or
- rank-based fitness assignment,
- multi-objective ranking,

The actual selection is performed in the next step. Parents are selected according to their fitness by means of one of the following algorithms:

- roulette-wheel selection,
- stochastic universal sampling,
- local selection, ,
- truncation selection,
- tournament selection

3.3 Recombination

Recombination produces new individuals in combining the information contained in the parents (parents - mating population). Depending on the representation of the variables of the individuals the following algorithms can be applied:

- All presentation:
 - discrete recombination (known from recombination of real valued variables), corresponds with uniform crossover (known from recombination of binary valued variables),
- Real valued recombination,
 - intermediate recombination,
 - line recombination,
- extended line recombination,
 - Binary valued recombination,
 - single-point / double-point /multi-point crossover,
 - uniform crossover,
 - shuffle crossover,
 - crossover with reduced surrogate, For the recombination of binary valued variables the name 'crossover' is established. This has mainly historical reasons. Genetic algorithms mostly used binary variables and the name 'crossover'. Both notions (recombination and crossover) are equivalent in

the area of Evolutionary Algorithms. For consistency, throughout this study the notion 'recombination' will be used (except when referring to specially named methods or operators

3.4 Mutation

After recombination every offspring undergoes mutation. Offspring variables are mutated by small perturbations (size of the mutation step)[15], with low probability. The representation of the variables determines the used algorithm. Two operators are explained:

- mutation operator for real valued variables,
- mutation for binary valued variables,

3.5 Reinsertion

After producing offspring they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original population. Another case is, when not all offspring are to be used at each generation or if more offspring are generated than needed. By a reinsertion[16] scheme is determined which individuals should be inserted into the new population and which individuals of the population will be replaced by offspring.

The used selection algorithm determines the reinsertion scheme:

- global reinsertion for all population based selection algorithm (roulette-wheel selection, stochastic universal sampling, truncation selection),
- local reinsertion for local selection.

3.6 Population models - parallel implementation of evolutionary algorithms

The extended management of populations (population models) allows the definition of extensions of Evolutionary Algorithms. These extensions can contribute to an increased performance of Evolutionary Algorithms.

The following extensions can be distinguished:

- global model[17],
- local model (diffusion model, neighborhood model, fine grained model),
- regional model (migration model, island model, coarse grained model),

3.7 Application of multiple/different strategies and competition between subpopulations

Based on the regional population model the application of multiple different strategies at the same time is possible. This is done by applying different operators and parameters for each subpopulation. For an efficient distribution of resources during an optimization competing subpopulations are used.

- application of multiple strategies,
- competition between subpopulations,
- These extensions of the regional population model contribute to an increased performance of Evolutionary Algorithms, especially for large and complex real-world applications.

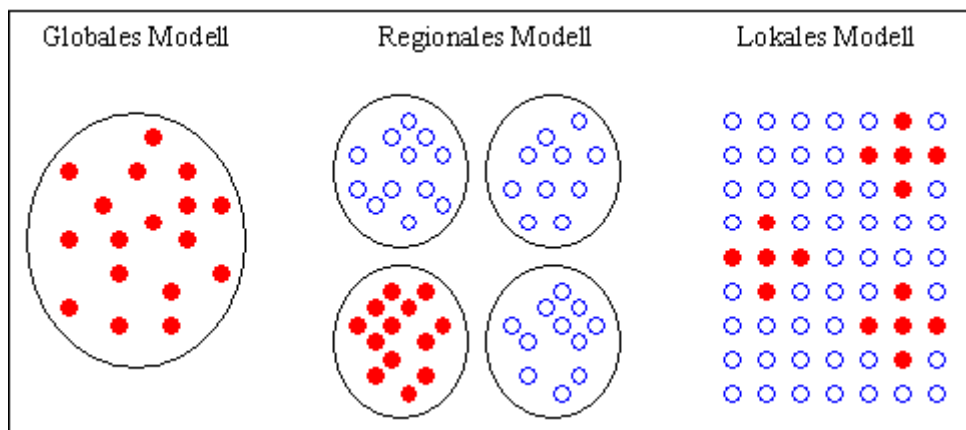
3.9 Population models - Parallel implementations

The population models may be distinguished from each other by looking at the range of the selection strategies of the parents and the definition of the selection pool. Three population models can be defined:

- global model,
- In the global model the selection takes place inside the whole population. That means, any two or more individuals may be selected together for the production of offspring. No restrictions exist.
- local model, The local model constrains the selection of parents to a local neighborhood.
- regional model[18], The regional model constrains the selection of parents to parts of the population isolated from each other, called subpopulation. Inside the subpopulation the selection is unrestricted (similar to the global model).

Figure 3-3 presents the corresponding selection pool.

Classification of population models by range of selection (selection pool)

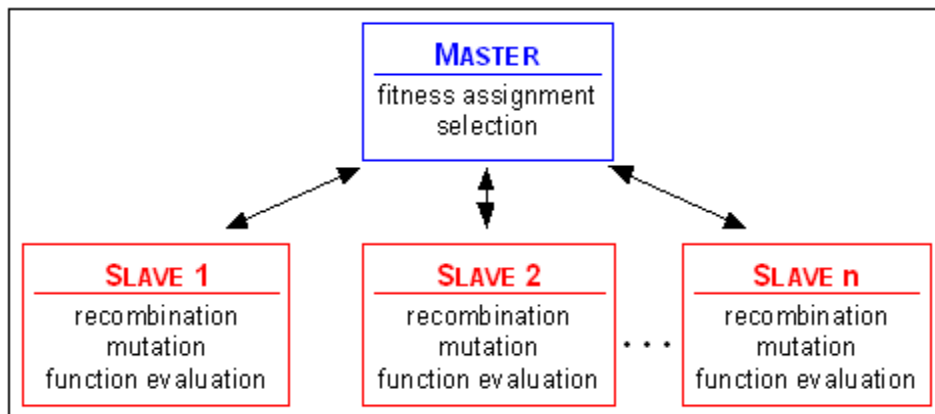


3.9 Global model - worker/farmer

The global model does not divide the population. Instead, the global model employs the inherent parallelism of evolutionary algorithms (population of individuals)[19]. The global model corresponds to the classical evolutionary algorithm.

The calculations where the whole population is needed - fitness assignment and selection - are performed by the master. All remaining calculations which are performed for one or two individuals each can be distributed to a number of slaves. The slaves perform recombination, mutation and the evaluation of the objective function separately. This is known as synchronous master-slave-structure, see figure 3-4

Fig. 3-4: Global population model (master-slave-structure)

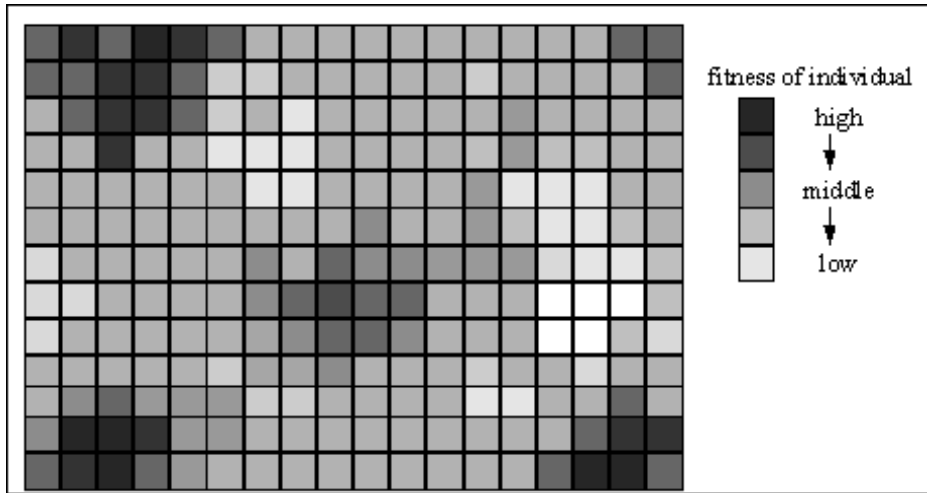


The slave calculations can be done in parallel. For most problems the evaluation of the objective function is the most time consuming part. In this case, the whole evolutionary algorithm is calculated by the master and only the objective function evaluation is distributed to the slaves. A nearly linear acceleration of the calculation time may be achieved (as long as the evaluation time of the objective function is higher than the communication time between master and slaves).

The global model is a simple way (and inherent to every evolutionary algorithm) to reduce very long computation times. Additionally, the distribution of objective function evaluation can be employed for any other population model as well.

3.10 Local model - Diffusion model

Fig. 3-5: Local model (diffusion evolutionary algorithm)



The local model (diffusion model) handles every individual separately and selects the mating partner in a local neighbourhood by local selection. Thus, a diffusion of information through the population takes place. During the search virtual islands, see figure will evolve.

3.11 Regional model – Migration

The regional model (migration model) divides the population into multiple subpopulations. These subpopulations evolve independently of each other for a certain number of generations (isolation time). After the isolation time a number of individuals is distributed between the subpopulation (migration). The number of exchanged individuals (migration rate), the selection method of the individuals for migration and the scheme of migration determines how much genetic diversity can occur in the subpopulation and the exchange of information between subpopulation.

The parallel implementation of the regional model showed not only a speed up in computation time, but it also needed less objective function evaluations when compared to a single population algorithm[20]. So, even for a single processor computer, implementing the parallel algorithm in a serial manner (pseudo-parallel) delivers better results (the algorithm finds the global optimum more often or with less function evaluations).

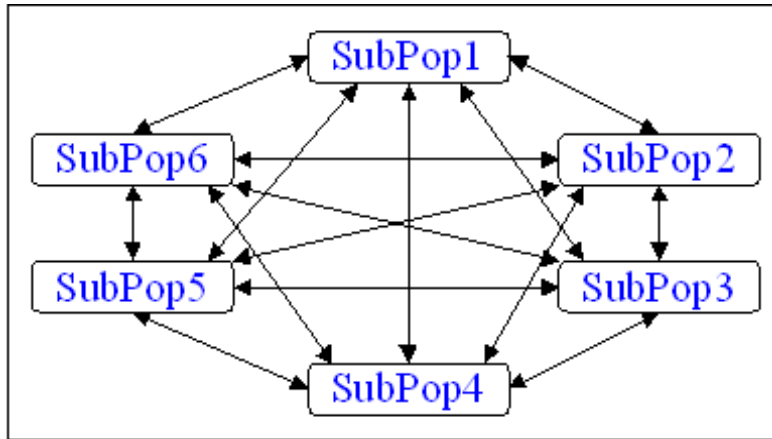
The selection of the individuals for migration can take place:

- uniformly at random (pick individuals for migration in a random manner),
- fitness-based (select the best individuals for migration).

Many possibilities exist for the structure of the migration of individuals between subpopulation. For example, migration may take place:

- between all subpopulations (complete net topology - unrestricted), see figure ,
- in a ring topology, see figure ,
- in a neighborhood topology, see figure .

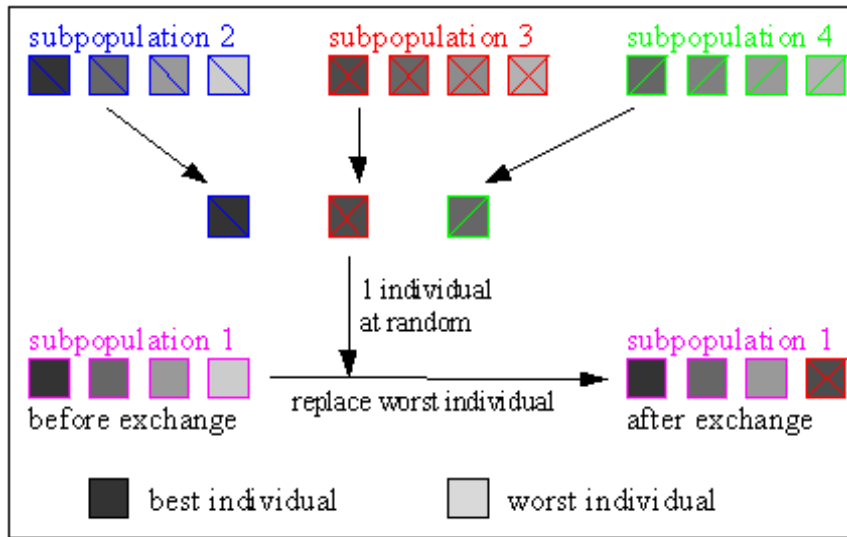
Fig. 3-6: Unrestricted migration topology (Complete net topology)



The most general migration strategy is that of unrestricted migration (complete net topology). Here, individuals may migrate from any subpopulation to another. For each subpopulation, a pool of potential immigrants is constructed from the other subpopulation. The individual migrants are then uniformly at random determined from this pool.

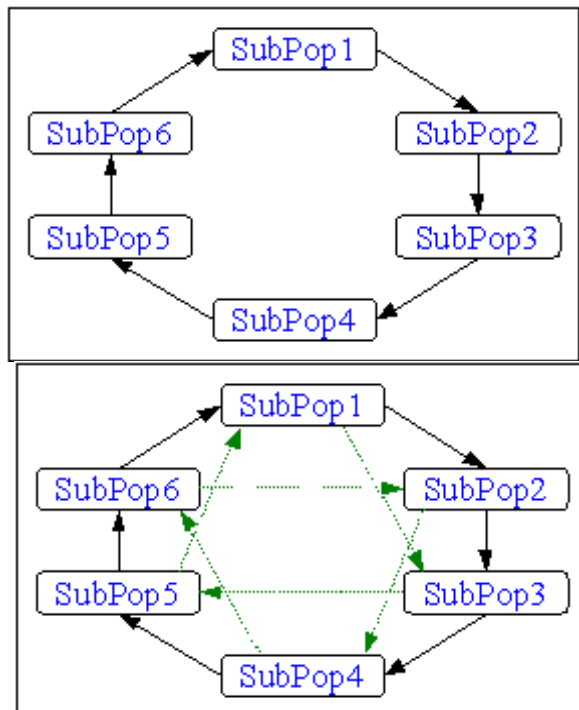
Figure gives a detailed description of the unrestricted migration scheme for 4 subpopulations with fitness-based selection. Subpopulation[22] 2, 3 and 4 construct a pool of their best individuals (fitness-based migration). 1 individual is uniformly at random chosen from this pool and replaces the worst individual in subpopulation 1. This cycle is performed for every subpopulation. Thus, it is ensured that no subpopulation will receive individuals from itself.

Fig. 3-7: Scheme for migration of individuals between subpopulation



The most basic migration scheme is the ring topology. Here, individuals are transferred between directionally adjacent subpopulations. For example, individuals from subpopulation 6 migrate only to subpopulation 1 and individuals from subpopulation 1 only migrate to subpopulation 2.

Fig.3-8: Ring migration topology; left: distance 1, right: distance 1 and 2



A similar strategy to the ring topology is the neighbourhood migration of figure . Like the ring topology, migration is made only between the nearest neighbours. However, migration may occur in either direction between subpopulations. For each subpopulation, the possible immigrants are determined, according to the desired selection method, from the adjacent subpopulations and a final selection is made from this pool of individuals (similar to figure).

Fig. 3-9: Neighbourhood migration topology (2-D grid)

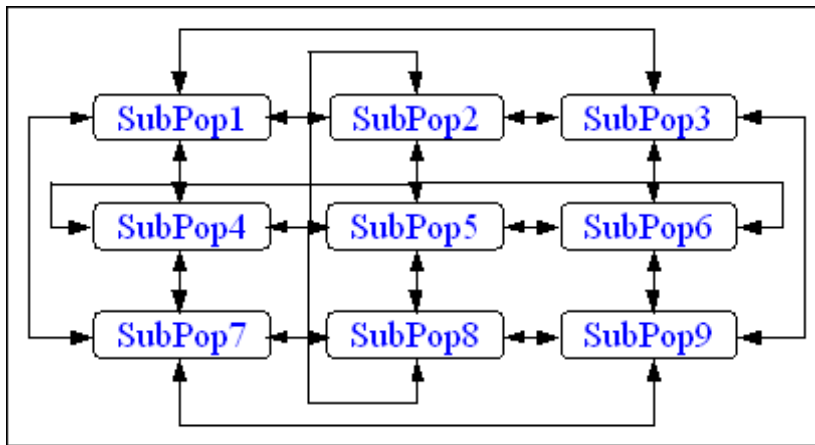


Figure shows a possible scheme for a 2-D implementation of the neighbourhood topology. Sometimes this structure is called a torus.