Genetic Algorithm

The genetic algorithm applies concepts from evolutionary biology and attempts to find a good solution by mimicking the process of natural selection and the "survival of the fittest". The algorithm can be outlined as follows.

- The potential solutions are viewed as members of a population. Each such member can be described, for example, by a vector. Initially we can start with a random population, i.e., a set of random vectors.
- In each iteration certain operations are executed that randomly change the population. Two fundamental operations are:
  - *Mutation.* This means that some randomly chosen individuals (vectors) undergo random changes. For example, some of their bits (in case of binary vectors) are reversed.
  - *Crossover.* Two randomly selected individuals (vectors) "mate" and create offsprings that inherit the combined properties of the parents. For example, the components of the offspring vectors are obtained by swapping some components of the parent vectors.
- After each round of random changes the fitness of each individual is evaluated by a *fitness function* that can measure how good is each solution vector. Those that are not fit enough, die, while the fit ones survive.
- The population evolves through such iterations. After a large number of iterations one can hope that the arising population already contains good solutions with high fitness value.

Advantages of Genetic Algorithms

- Offers a chance that the initial solutions evolve into good ones that are close to optimal.
- The evolving population tends to have more and more fit individuals, so the algorithm may find many good solutions, not just one.

Disadvantages of Genetic Algorithms

- Nothing is guaranteed formally (usual problem with heuristic algorithms).
- There is no general way of estimating the number of iterations needed for a problem, so we cannot easily decide how close is the result to a global optimum.

Exercise

Assume that in a Genetic Algorithm the fitness of each individual member of the population, in each iteration, can be modeled as a random variable that is uniformly distributed in the interval $[0, 1]$. Further asssume that for each individual this random variable is independent from all the others, in each iteration. We are looking for a solution (an individual) with as high fitness value as possible. The initial population has size n. After each iteration of modifying the population via mutation and crossover operations we only keep the n fittest individuals, the rest "die". Which of the following is correct?

1. The maximum fitness value found in the population will converge to 1, as the number of iterations grow. The reason is that the fittest individual in the current population always survives when we keep only the n fittest. Since the fitness in our case is modeled as an independent, uniformly distributed random variable in $[0, 1]$, therefore, the probability that we have an individual with fitness in $[1 - \epsilon, 1]$ tends to 1 for every fixed $0 < \epsilon < 1$, if the number of iterations grow.

2. Since there is no general performance guarantee for the Genetic Algorithm, therefore, we can never gurantee any kind of convergence.

3. The convergence of the maximum occuring fittness to 1 under the given assumptions is only guaranteed if n also grows with the iterations.

Solution (with explanation): Let p denote the probability that in a given iteration there is an individual with fitness in $[1 - \epsilon, 1]$. Under the given assumptions, in any iteration, the value of p can be computed this way:

$$p = 1 - (1 - \epsilon)^n. \tag{1}$$

The reason is that $(1 - \epsilon)^n$ is the probability that *all* the n individuals have their fitness in the interval $[0, 1 - \epsilon)$. Then $1 - (1 - \epsilon)^n$ is the probability of the complement event, which is the event that *not all* have their fitness in $[0, 1-\epsilon)$, that is, *at least one* must fall in $[1-\epsilon, 1]$. Answer 1 claims that for every fixed $\epsilon > 0$ we have $p \to 1$. From formula (1), we see that this would require $(1-\epsilon)^n \to 0$. However, for a constant n this is not satisfied. On the other hand, if $n \to \infty$, then $(1-\epsilon)^n \to 0$ indeed holds. Therefore, the correct choice is Answer 3.