GENETIC ALGORITHMS

Outline of the Algorithm

1. The algorithm begins by creating a *random initial population*.
2. The algorithm then creates a sequence of *new populations*. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
    a. *Scores* each member of the current population by *computing its fitness value.*
    b. *Scales* the raw fitness scores to convert them into a more usable range of values.
    c. *Selects* members, called *parents*, based on their fitness.
    d. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
    e. *Produces* children from the parents. Children are produced either by making random changes to a single parent—*mutation*—or by combining the vector entries of a pair of parents—*crossover*.
    f. *Replaces* the current population with the children to form the next generation.
3. The algorithm *stops* when one of the *stopping criteria* is met.

Crossover

- Parents are selected using of the selection schemes.
- It is, however, not a given that selected parents will mate.
- Each pair of parents has a probability, *pc*, of producing offspring.
- Usually, a high crossover rate is used.

Binary Representations

Most of the crossover operators for binary representations are applied to *two* selected parents as in the following Algorithm.

Algorithm 9.1 Generic Algorithm for Bitstring Crossover

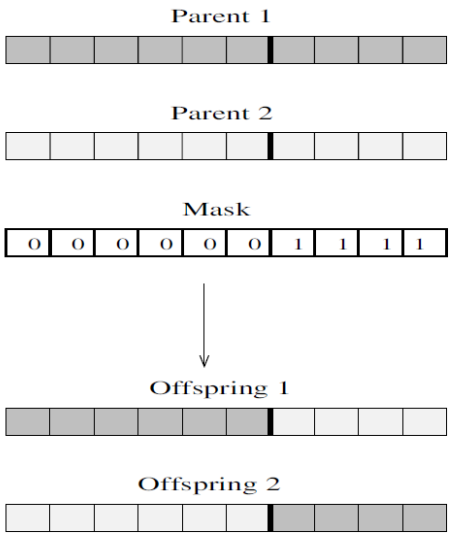
Let $\tilde{\mathbf{x}}_1(t) = \mathbf{x}_1(t)$ and $\tilde{\mathbf{x}}_2(t) = \mathbf{x}_2(t)$;
if $U(0,1) \leq p_c$ **then**
    Compute the binary mask, $\mathbf{m}(t)$;
    for $j = 1, \ldots, n_x$ **do**
        if $m_j = 1$ **then**
            //swap the bits
            $\tilde{\mathbf{x}}_{1j}(t) = \mathbf{x}_{2j}(t)$;
            $\tilde{\mathbf{x}}_{2j}(t) = \mathbf{x}_{1j}(t)$;
        end
    end
end

Several crossover operators have been developed:

1- **One-point crossover:**
   - Holland suggested that segments of genes be swapped between the parents to create their offspring.
   - A one-point crossover operator was developed that randomly selects a crossover point, and the bit strings after that point are swapped between the two parents.
   - The mask is computed using Algorithm 9.2.

Algorithm 9.2 One-Point Crossover Mask Calculation

Select the crossover point, $\xi \sim U(1, n_x - 1)$;
Initialize the mask: $m_j(t) = 0$, for all $j = 1, \ldots, n_x$;
for $j = \xi + 1$ *to* n_x **do**
    $m_j(t) = 1$;
end

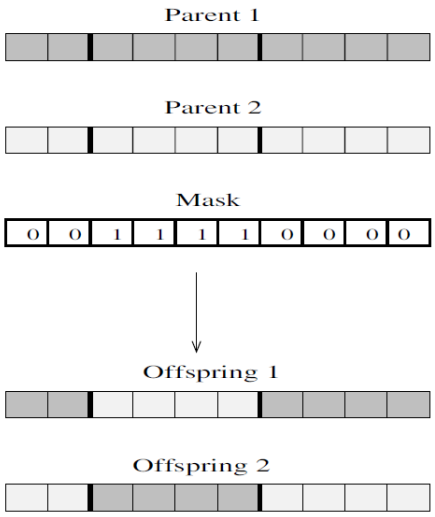


(b) One-point Crossover

2- **Two-point crossover:**
   - In this case two bit positions are randomly selected, and the bit strings between these points are swapped.
   - The mask is calculated using Algorithm 9.3.

Algorithm 9.3 Two-Point Crossover Mask Calculation

Select the two crossover points, $\xi_1, \xi_2 \sim U(1, n_x)$;
Initialize the mask: $m_j(t) = 0$, for all $j = 1, \ldots, n_x$;
for $j = \xi_1 + 1$ *to* ξ_2 **do**
    $m_j(t) = 1$;
end



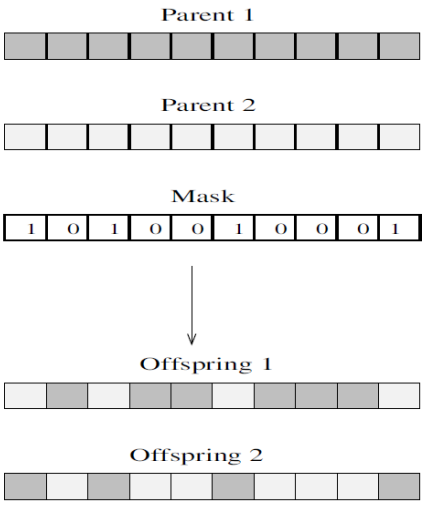
(c) Two-point Crossover

3- Uniform crossover:

- The *nx*-dimensional mask is created randomly as summarized in Algorithm 9.4.
- Here, *px*is the bit-swapping probability.
  - If *px*= 0.5, then each bit has an equal chance to be swapped.

Algorithm 9.4 Uniform Crossover Mask Calculation

Initialize the mask: $m_j(t) = 0$, for all $j = 1, \ldots, n_x$;
for $j = 1$ *to* n_x **do**
  if $U(0,1) \leq p_x$ **then**
    $m_j(t) = 1$;
  end
end



(a) Uniform Crossover

Floating-Point Representation

There are three operators:

- *Linear operator*: From the parents, $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$, three candidate offspring are generated as:

$$(\mathbf{x}_1(t) + \mathbf{x}_2(t)),\ (1.5\mathbf{x}_1(t) - 0.5\mathbf{x}_2(t)) \text{ and } (-0.5\mathbf{x}_1(t) + 1.5\mathbf{x}_2(t))$$

The two best solutions are selected as the offspring.

- *Arithmetic operator*: takes a weighted average over two or more parents.

$$\tilde{x}_{ij}(t) = (1-\gamma)x_{1j}(t) + \gamma x_{2j}(t)$$

with $\gamma \in [0, 1]$.

- *Geometrical crossover:* to produce a single offspring as follows:

$$\tilde{x}_{ij}(t) = (x_{1j}x_{2j})^{0.5}$$

Mutation

- The aim of mutation is to introduce new genetic material into an existing individual; that is, to add diversity to the genetic characteristics of the population.
- Mutation is applied at a certain probability, *pm*, to each gene of the offspring.
- Usually pm is a small value, $pm \in [0, 1]$, to ensure that good solutions are not distorted too much.

Binary Representations

For binary representations, the following mutation operators have been developed:

1- **Uniform (random) mutation**, where bit positions are chosen randomly and the corresponding bit values negated. Uniform mutation is summarized in Algorithm 9.6.

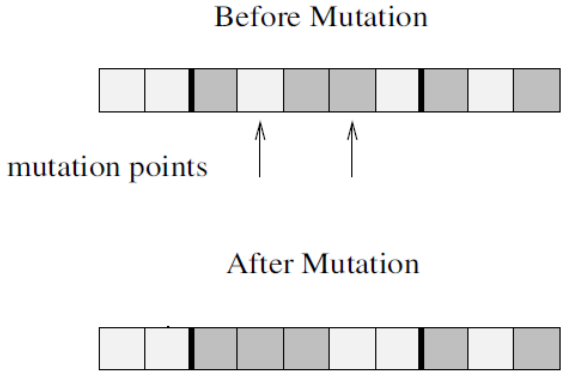
Algorithm 9.6 Uniform/Random Mutation

```
for j = 1, ..., n_x do
    if U(0,1) ≤ p_m then
        x'_ij(t) = ¬x̃_ij(t), where ¬ denotes the boolean NOT operator;
    end
end
```

2- **Inorder mutation**, where two mutation points are randomly selected and only the bits between these mutation points undergo random mutation. Inorder mutation is illustrated in Algorithm 9.7.

Algorithm 9.7 Inorder Mutation

Select mutation points, $\xi_1, \xi_2 \sim U(1, \ldots, n_x)$;
for $j = \xi_1, \ldots, \xi_2$ **do**
  if $U(0,1) \leq p_m$ **then**
    $x'_{ij}(t) = \neg \tilde{x}_{ij}(t)$;
  end
end



(b) Inorder Mutate

Floating-Point Representations

Uniform mutation:

$$x'_{ij}(t) = \begin{cases} \tilde{x}_{ij}(t) + \Delta(t, x_{max,j} - \tilde{x}_{ij}(t)) & \text{if a random digit is 0} \\ \tilde{x}_{ij}(t) + \Delta(t, \tilde{x}_{ij}(t) - x_{min,j}(t)) & \text{if a random digit is 1} \end{cases}$$

where $\Delta(t, x)$ returns random values from the range $[0, x]$.

Replacement Algorithms

There are two main classes of GAs are identified based on the replacement strategy used:

1- ***Generational genetic algorithms (GGA):*** the replacement strategy replaces all parents with their offspring after all offspring have been created and mutated. This results in no overlap between the current population and the new population (assuming that elitism is not used)

2- **Steady state genetic algorithms (SSGA)**: a decision is made immediately after an offspring is created and mutated as to whether the parent or the offspring survives to the next generation. Thus, there exists an overlap between the current and new populations.

- The amount of overlap between the current and new populations is referred to as the ***generation gap***. GGAs have a zero generation gap, while SSGAs generally have large generation gaps.

- A number of replacement strategies have been developed for SSGAs:
  - **Replace worst**, where the offspring replaces the *worst* individual of the current population.
  - **Replace random**, where the offspring replaces a *randomly* selected individual of the current population.

- **Kill tournament**, where a *group* of individuals is randomly selected, and the *worst* individual of this group is replaced with the offspring.
- **Replace oldest**, where a first-in-first-out strategy is followed by replacing the *oldest* individual of the current population.
- **Elitist** strategies of the above replacement strategies have also been developed, where the best individual is *excluded* from selection.
- **Parent-offspring** competition, where a selection strategy is used to decide if an offspring replaces one of its own parents.

Assignment

1- Suppose a genetic algorithm uses chromosomes of the form x = *abcdefgh*with a fixed length of eight genes. Each gene can be any digit between 0and 9. Let the fitness of individual x be calculated as:

$$f(x) = (a+b) - (c+d) + (e+f) - (g+h) \ ,$$

And let the initial population consist of four individuals with the followingchromosomes:

$$\begin{aligned} x_1 &= 65413532 \\ x_2 &= 87126601 \\ x_3 &= 23921285 \\ x_4 &= 41852094 \end{aligned}$$

a- Evaluate the fitness of each individual, showing all your workings, andarrange them in order with the fittest first and the least fit last.

b- Perform the following crossover operations:
   i) Cross the fittest two individuals using one–point crossover at the middle point.
   ii) Cross the second and third fittest individuals using a two–pointcrossover (points b and f).
   iii) Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.

c- Suppose the new population consists of the six offspring individualsreceived by the crossover operations in the above question. Evaluatethe fitness of the new population, showing all your workings. Has theoverall fitness improved?

d- By looking at the fitness function and considering that genes can onlybe digits between 0 and 9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of themaximum fitness.