Crossover Techniques in GAs

Credits:

Soft Computing Applications, Dr. Debasis Samanta, IIT Kharagpur

RC Chakraborty, www.myreaders.info

Reproduction in Genetic Algorithm

Reproduction:

- **Crossover**
- Mutation
- Inversion

These genetic operators varies from one encoding scheme to another.

- Binary coded GAs
- Real-coded GAs
- Tree-coded GAs

Mating Pool: Prior to crossover operation

- A mating pair (each pair consists of two strings) are selected at random. Thus, if the size of mating pool is N, then $\frac{N}{2}$ mating pairs are formed.[Random Mating]
- The pairs are checked, whether they will participate in reproduction or not by tossing a coin, whose probability being p_c. If p_c is head, then the parent will participate in reproduction. Otherwise, they will remain intact in the population.

Note :
Generally, $p_c = 1.0$, so that almost all the parents can participate in production.

Crossover operation

Once, a pool of mating pair are selected, they undergo through crossover operations.

1. In crossover, there is an exchange of properties between two parents and as a result of which **two** offspring solutions are produced.
2. The crossover point(s) (also called k-point(s)) **is(are)** decided using a random number generator generating integer(s) in between 1 and L, where L is the length of the chromosome.
3. Then we perform exchange of gene values with respect to the k-point(s)

There are many exchange mechanisms and hence crossover strategies.

Crossover Techniques in Binary Coded GA

Crossover operations in Binary-coded GAs

- There exists a large number of crossover schemes, few important of them are listed in the following.

  1. Single point crossover
  2. Two-point crossover
  3. Multi-point crossover (also called n-point crossover)
  4. Uniform crossover (UX)
  5. Half-uniform crossover (HUX)
  6. Shuffle crossover
  7. Matrix crossover (Tow-dimensional crossover)
  8. Three parent crossover

Single point crossover

1. Here, we select the K-point lying between 1 and L. Let it be k.
2. A single crossover point at k on both parent's strings is selected.
3. All data beyond that point in either string is swapped between the two parents.
4. The resulting strings are the chromosomes of the offsprings produced.

Single point crossover: Illustration

Before Crossover

Parent 1 :	0	1	1	0	0	0	1	0
Parent 2 :	1	0	1	0	1	1	0	0

Crossover Point -k

Select crossover points randomly

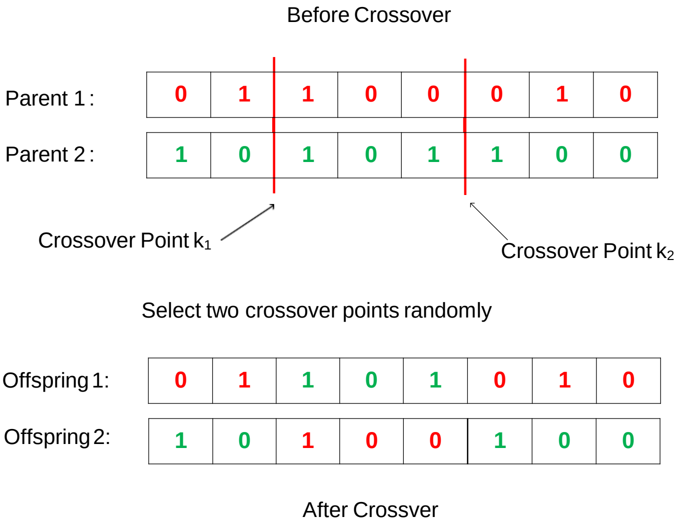
Offspring 1:	0	1	1	0	1	1	0	0
Offspring 2:	1	0	1	0	0	0	1	0

After Crossver

Two-point crossover

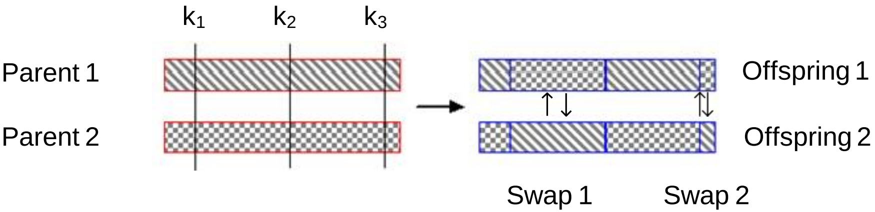
1. In this scheme, we select two different crossover points k_1 and k_2 lying between 1 and L at random such that $k_1 \neq k_2$.
2. The middle parts are swapped between the two strings.
3. Alternatively, left and right parts also can be swapped.

Two-point crossover: Illustration



Multi-point crossover

1. In case of multi-point crossover, a number of crossover points are selected along the length of the string, at random.
2. The bits lying between alternate pairs of sites are then swapped.



Uniform Crossover (UX)

- Uniform crossover is a more general version of the multi-point crossover.
- In this scheme, at each bit position of the parent string, we toss a coin (with a certain probability p_s) to determine whether there will be swap of the bits or not.
- The two bits are then swapped or remain unaltered, accordingly.

Uniform crossover (UX): Illustration

Before crossover

Parent 1 :	1	1	0	0	0	1	0	1	1	0	0	1
Parent 2 :	0	1	1	0	0	1	1	1	0	1	0	1
Coin tossing:	1	0	0	1	1	1	0	1	1	0	0	1

After crossover

Offspring 1:	1	1	1	0	0	1	1	1	1	1	0	1
Offspring 2:	0	1	0	0	0	1	0	1	0	0	0	1

Rule: If the toss is 0 than swap the bits between P1 and P2

Uniform crossover with crossover mask

- Here, each gene is created in the offspring by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosome.
- Where there is a 1 in the mask, the gene is copied from the first parent
- Where there is a 0 in the mask, the gene is copied from the second parent.
- The reverse is followed to create another offsprings.

Uniform crossover with crossover mask: Illustration

Before Crossover

Parent 1 :	1	1	0	0	0	1	0	1
Parent 2 :	0	1	1	0	0	1	1	1

Mask	1	0	0	1	1	1	0	1

Offspring 1:	1	1	1	0	0	1	1	1
Offspring 2:	0	1	0	0	0	1	0	1

When there is a 1 in the mask, the gene is copied from Parent 1 else from Parent 2.

When there is a 1 in the mask, the gene is copied from Parent 2 else from Parent 1.

After Crossver

Half-uniform crossover (HUX)

- In the half uniform crossover scheme, exactly half of the non-matching bits are swapped.

    1. Calculate the Hamming distance (the number of differing bits) between the given parents.
    2. This number is then divided by two.
    3. The resulting number is how many of the bits that do not match between the two parents will be swapped but probabilistically.
    4. Choose the locations of these half numbers (with some strategies, say coin tossing) and swap them.

Half-uniform crossover: Illustration

Before crossover

Parent 1 :	1	1	0	0	0	0	1	0
Parent 2 :	1	0	0	1	1	0	1	1

Here, Hamming distance is 4

Tossing:		1		0	1			1

If toss is 1, then swap the bits else remain as it is

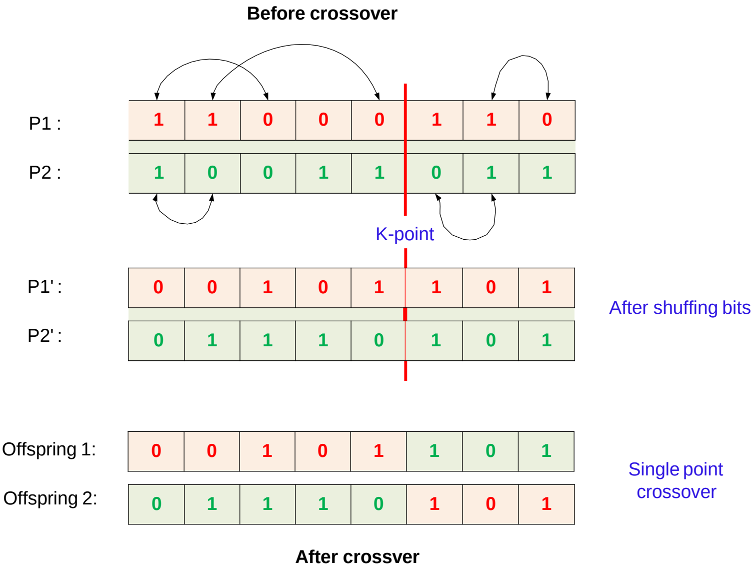
Offspring 1:	1	0	0	0	1	0	1	1
Offspring 2:	1	1	0	1	0	0	1	0

After crossver

Shuffle crossover

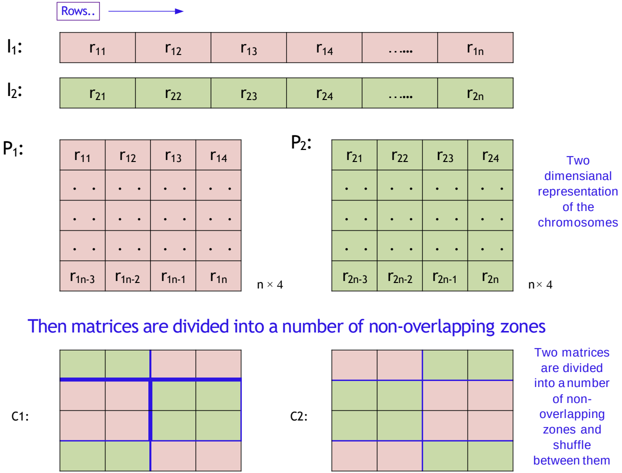
- A single crossover point is selected. It divides a chromosome into two parts called schema.
- In both parents, genes are shuffled in each schema. Follow some strategy for shuflling bits
- Schemas are exchanged to create offspring (as in single crossover)

Shuffle crossover: Illustration



Matrix crossover

The matrix crossover strategy is expained with the following illustration.



Three parent crossover

- In this techniques, three parents are randomly chosen.
- Each bit of the first parent is compared with the bit of the second parent.
- If both are the same, the bit is taken for the offspring.
- Otherwise, the bit from the third parent is taken for the offspring.

Three parent crossover: Illustration

P1:	1	1	0	1	0	0	0	1
P2:	0	1	1	0	1	0	0	1
P3:	0	1	1	0	1	1	0	1
C1:	0	1	1	0	1	0	0	1

Note: Sometime, the third parent can be taken as the crossover mask.

Comments on the binary crossover techniques

1. **Non-uniform variation:**

   It can not combine all possible schemas (i.e. building blocks)

   For example : it can not in general combine instances of
   1 1 * * * * * 1
   and
   *** * * * 1 1 * ***
   to form an instance of
   1 1 * * 1 1 * 1.

2. **Positional bias:**

   The schemas that can be created or destroyed by a crossover depends strongly on the location of the bits in the chromosomes.

Comments on the binary crossover techniques

3. **End-point bias:**

   It is also observed that single-point crossover treats some loci preferentially, that is, the segments exchanged between the two parents always contain the end points of the strings.

4. **Hamming cliff problem:**

   A one-bit change can make a large (or a small) jump.
   A multi-bits can make a small (or a large gap).

   For example, **1000** $\Longrightarrow$ **0111**
   (Here, Hamming distance = 4, but distance between phenotype is 1)

   Similarly, **0000** $\Longrightarrow$ **1000**
   (Here, Hamming distance = 1, but distance between phenotype is 8)

Comments on the binary crossover techniques

- To reduce the positional bias and end-point bias, two-point crossover and multi-point crossover schemes have been evolved.
- In contrast, UX and HUX distribute the patterns in parent chromosomes largely resulting too much deflections in the offspring.
- In summary, binary coding is the simplest encoding and its crossover techniques are fastest compared to the crossover techniques in other GA encoding schemes.

Thank you.