Course Outcome

1. Find weights required to perform classification of pattern using perceptron networks) . The patterne

$(1,1,-1)$ and $(1,-1,-1)$ belonging to target class -1.

$(-1,1,1)$ and $(-1,-1,1)$ belonging to target class $+1$.

ans. Let $\omega_1 = \omega_2 = \omega_3 = 0 \quad b = 0 \quad \alpha = 1$.

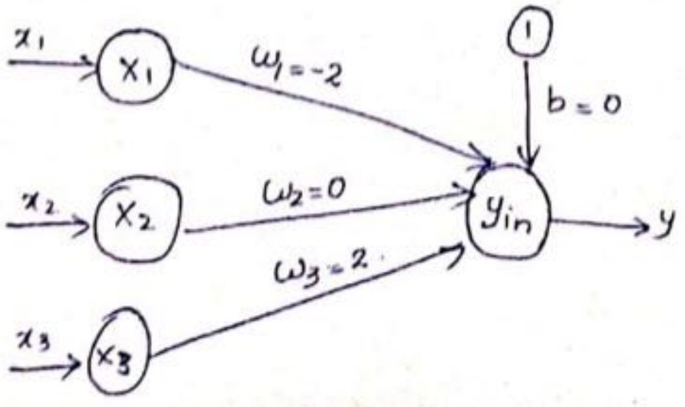
$$y_{in} = b + x_1\omega_1 + x_2\omega_2 + x_3\omega_3 \qquad y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > 0 \\ 0 & \text{if } y_{in} = 0 \\ -1 & \text{if } y_{in} < 0 \end{cases}$$

$$W_{new} = W_{old} + \alpha t x_i \qquad b_{new} = b_{old} + \alpha t$$

	x_1	x_2	x_3	t	y_{in}	y	$\overset{0}{\omega_1}$	$\overset{0}{\omega_2}$	$\overset{0}{\omega_3}$	$\overset{0}{b}$
	1	1	-1	-1	0	0	-1	-1	1	-1
Epoch 1.	1	-1	-1	-1	-2	-1	-1	-1	1	-1
	-1	1	1	1	0	0	-2	0	2	0
	-1	-1	1	1	4	1	-2	0	2	0
	1	1	-1	-1	-4	-1	-2	0	2	0
epoch2.	1	-1	-1	-1	-4	-1	-2	0	2	0
	-1	1	1	1	4	1	-2	0	2	0
	-1	-1	1	1	4	1	-2	0	2	0

Since the desired & calculated o/p are same we can stop.

$\omega_1 = -2 \quad \omega_2 = 0 \quad \omega_3 = 2 \quad b = 0$



Q) Explain the architecture and training algorithm of Adaline network . Use Adaline network to train NOR logic function with bipolar inputs and targets. Perform 2 epochs of training.

Ans: Adaline is a single unit neuron, which receives the input from several units and also from units called bias

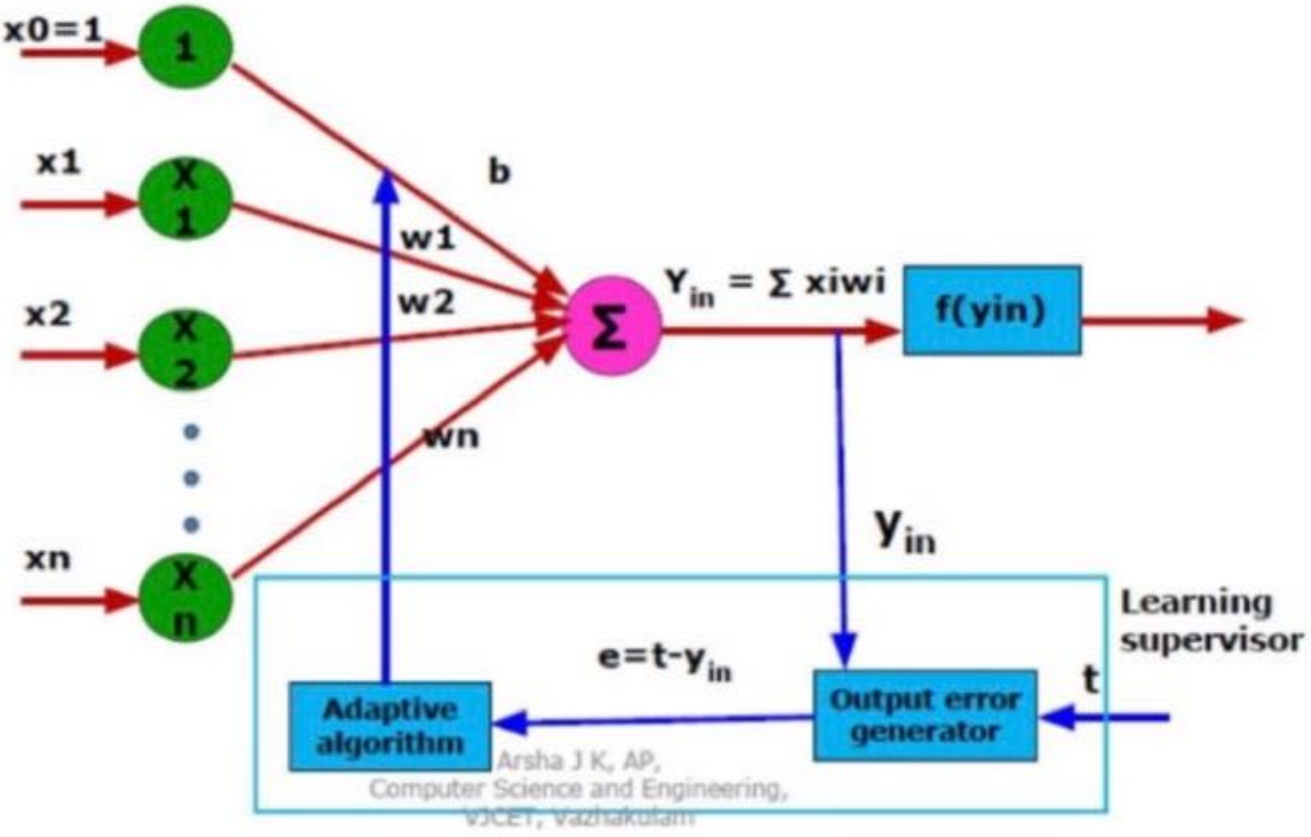
✔ Basic Adaline model consist of trainable weights

✔ Input are either of 2 values(+1 or -1) and the weight have signs(positive or negative)

✔ Initially random weights are assigned

✔ The net input is calculated is applied to a quantizer transfer function that restore the output to +1 or -1

✔ Adaline model compare the actual output with the target output and on the basis of training algorithm, the weights are adjusted



ADALINE TRAINING ALGORITHM

Step 0: Weights and bias are set to some random values other than zero. Learning rate α is set

Step 1: Perform Steps 2-6 when stopping condition is false.

Step 2: Perform steps 3-5 for each bipolar training pair s:t

Step 3: Set activations for input units i=1 to n xi=si

Step 4: Calculate the net input to the output unit

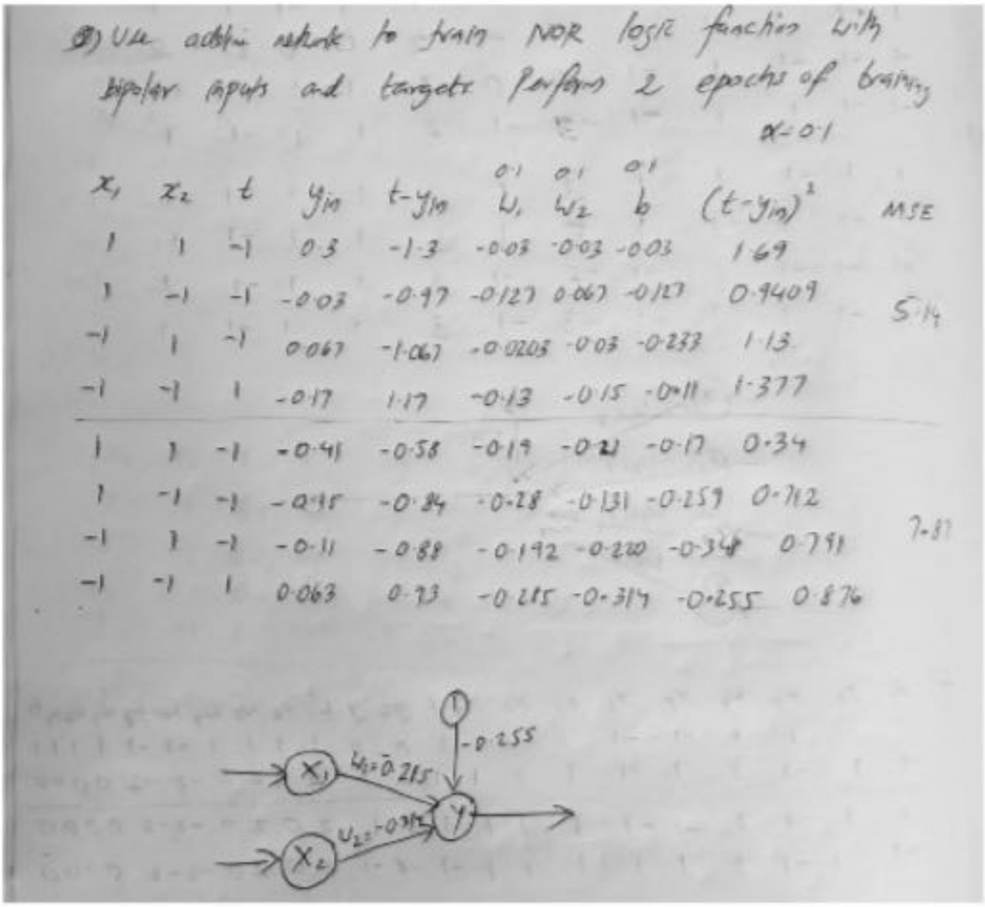
$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

Step 5: Update the weights and bias for i=1 to n

$$wi(new) = wi(old) + \alpha(t - yin)xi$$

$$b(new) = b(old) + \alpha(t - y_{in})$$

Step 6: If highest weight change that occurred during training is smaller than a specified tolerance then stop the training else continue. (Stopping condition)



Q) Use adaline network to train NOR logic function with bipolar inputs and targets Perform 2 epochs of training

$\alpha = 0.1$

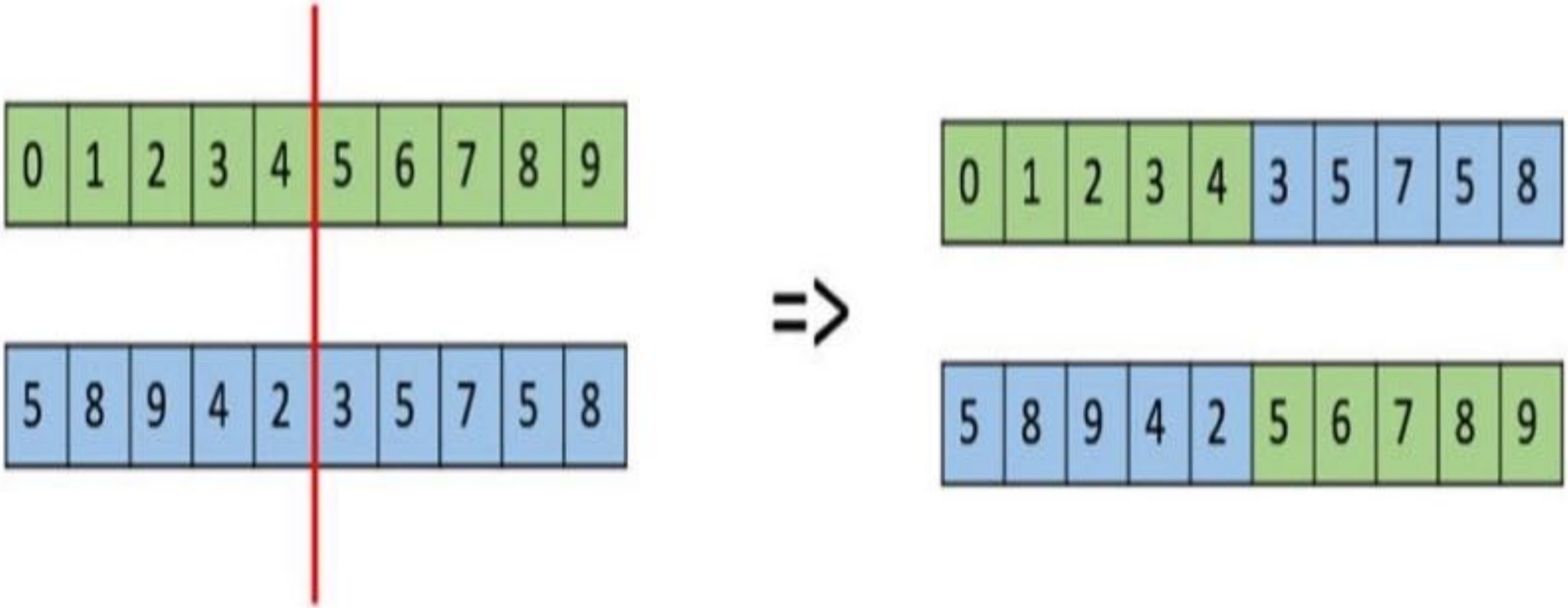
x_1	x_2	t	y_{in}	$t-y_{in}$	w_1	w_2	b	$(t-y_{in})^2$	MSE
					0.1	0.1	0.1		
1	1	-1	0.3	-1.3	-0.03	-0.03	-0.03	1.69	
1	-1	-1	-0.03	-0.97	-0.127	0.067	-0.127	0.9409	5.14
-1	1	-1	0.067	-1.067	-0.0203	-0.03	-0.233	1.13	
-1	-1	1	-0.17	1.17	-0.13	-0.15	-0.11	1.377	
1	1	-1	-0.41	-0.58	-0.19	-0.21	-0.17	0.34	
1	-1	-1	-0.15	-0.84	-0.28	-0.131	-0.259	0.712	
-1	1	-1	-0.11	-0.88	-0.192	-0.220	-0.348	0.791	7.81
-1	-1	1	0.063	0.73	-0.215	-0.314	-0.255	0.876	

CROSSOVER

- Various crossover techniques are
  - Single- point Crossover
  - Two- point crossover
  - Multipoint cross over/N point cross over
  - Uniform Crossover
  - Three parent Crossover
  - Crossover with reduced Surrogate
  - Shuffle crossover
  - Precedence Preservative Crossover
  - Ordered crossover
  - Partially matched crossover

Single-Point Crossover

- In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.



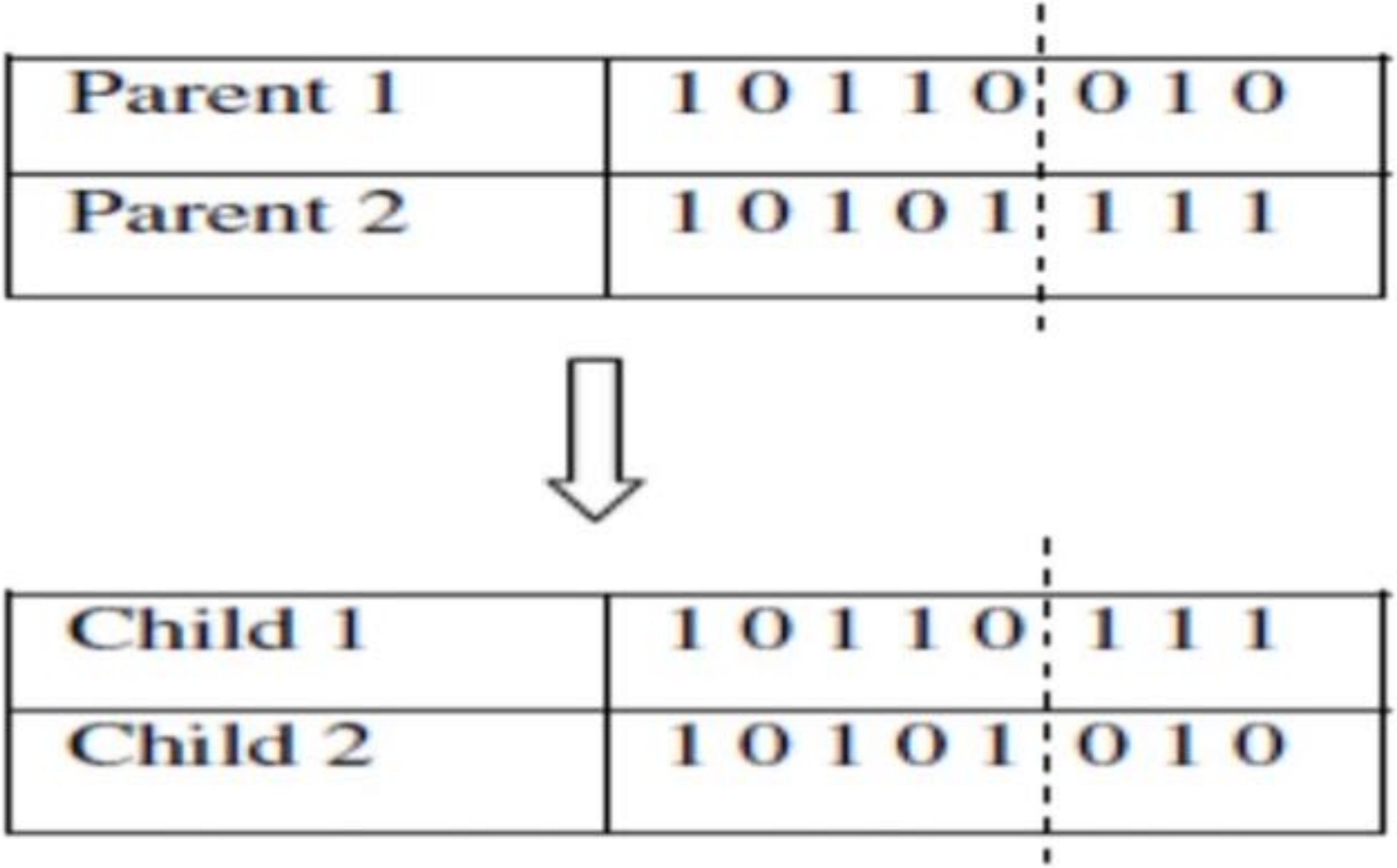
0 1 2 3 4 5 6 7 8 9

5 8 9 4 2 3 5 7 5 8

=>

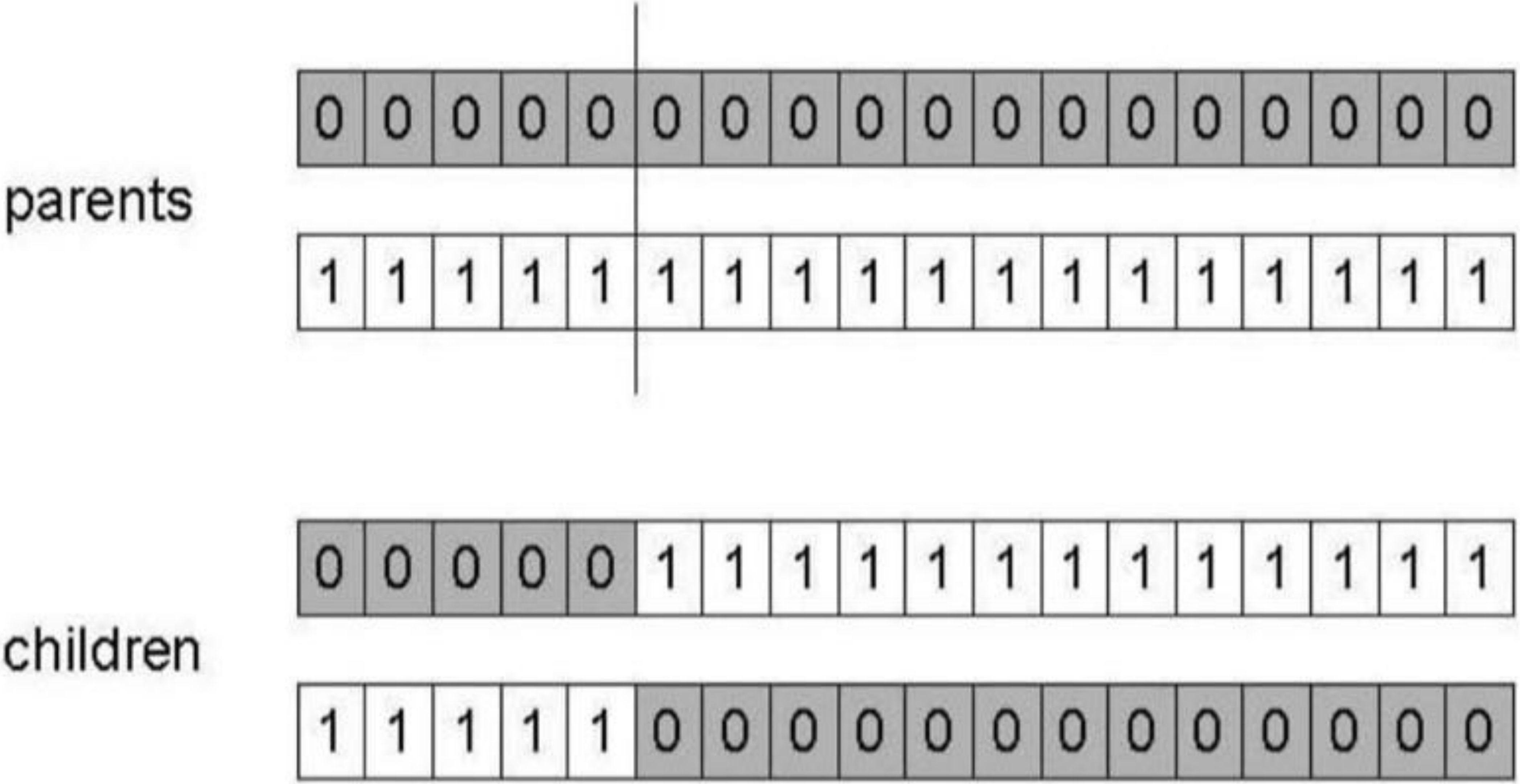
0 1 2 3 4 3 5 7 5 8

5 8 9 4 2 5 6 7 8 9



Parent 1	1 0 1 1 0 ¦ 0 1 0
Parent 2	1 0 1 0 1 ¦ 1 1 1

Child 1	1 0 1 1 0 ¦ 1 1 1
Child 2	1 0 1 0 1 ¦ 0 1 0



parents

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

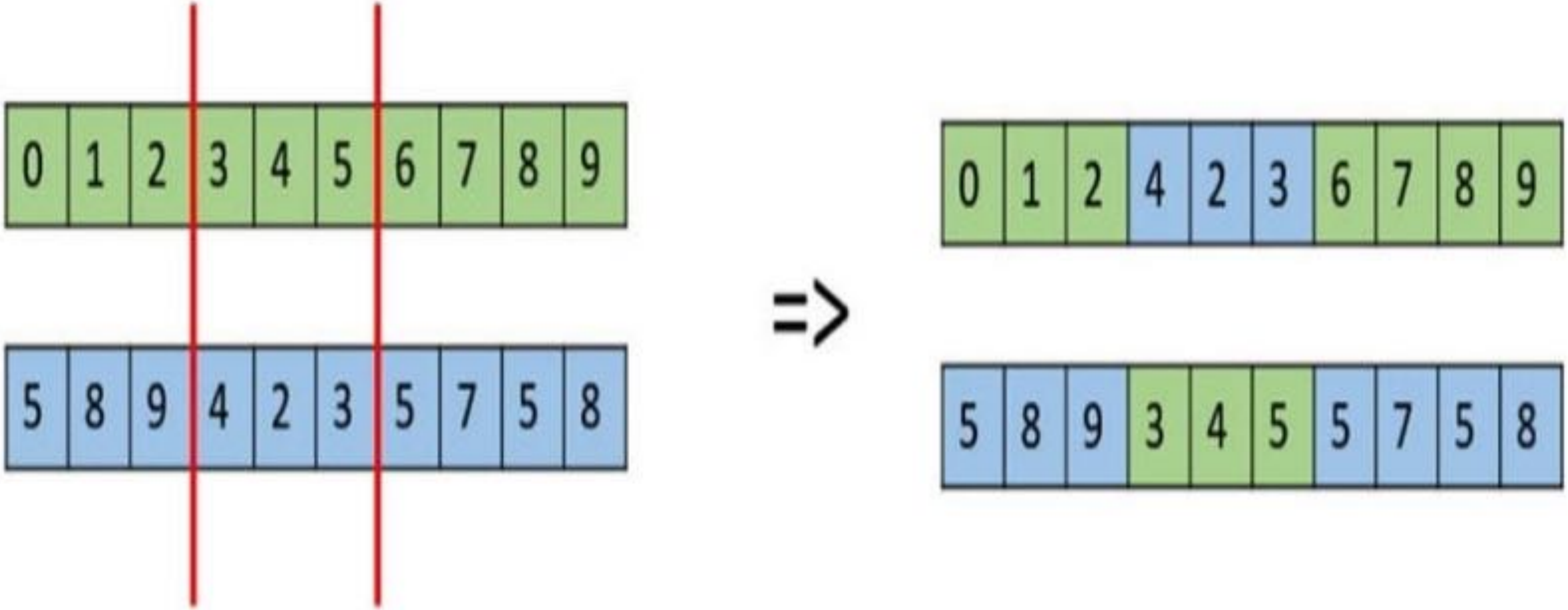
children

0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

Two-Point Crossover

- Two crossover points are chosen and the content between these points are exchanged between two matted parents



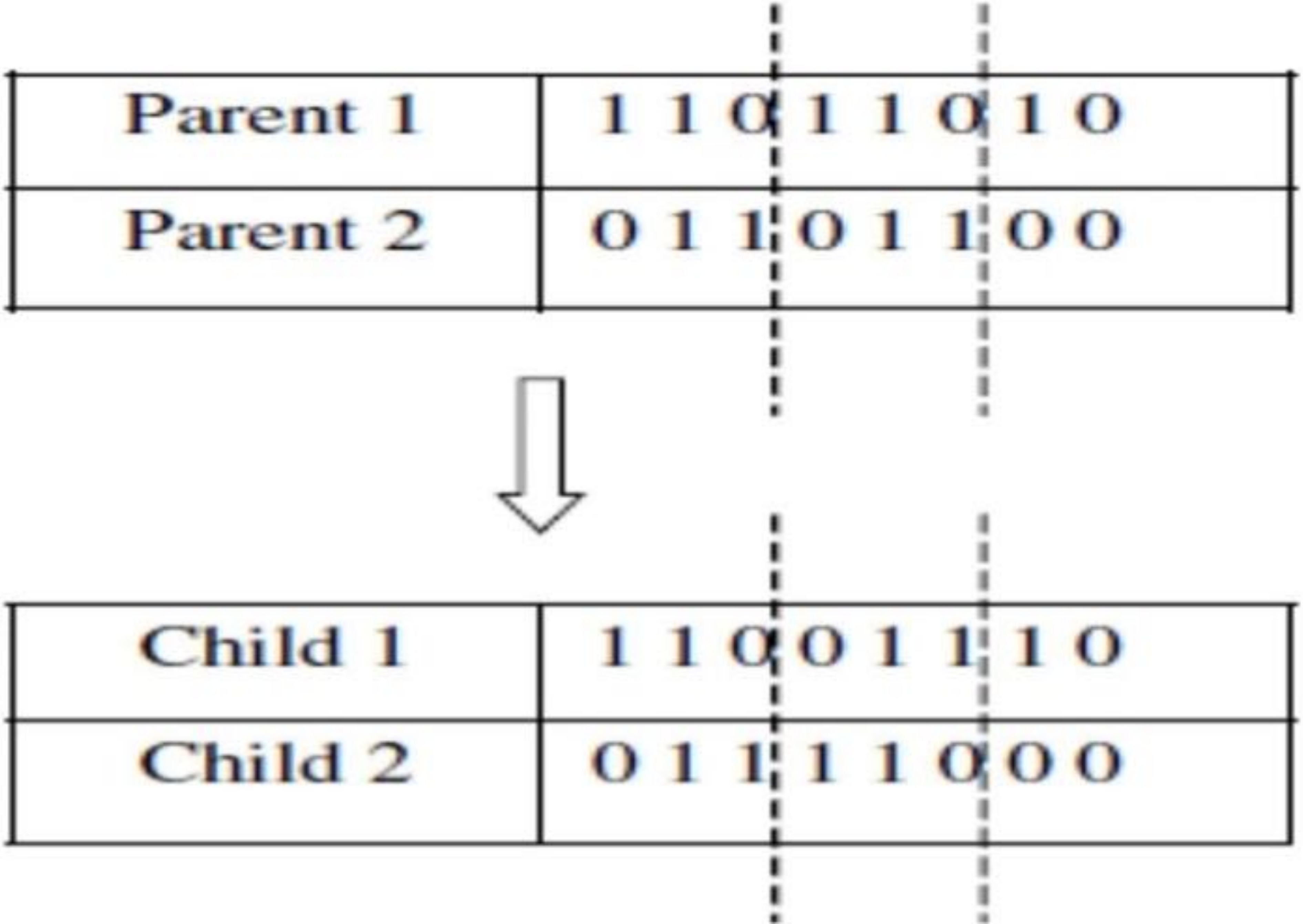
0 1 2 3 4 5 6 7 8 9

5 8 9 4 2 3 5 7 5 8

=>

0 1 2 4 2 3 6 7 8 9

5 8 9 3 4 5 5 7 5 8



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

Child 1	1 1 0 0 1 1 1 0
Child 2	0 1 1 1 1 0 0 0

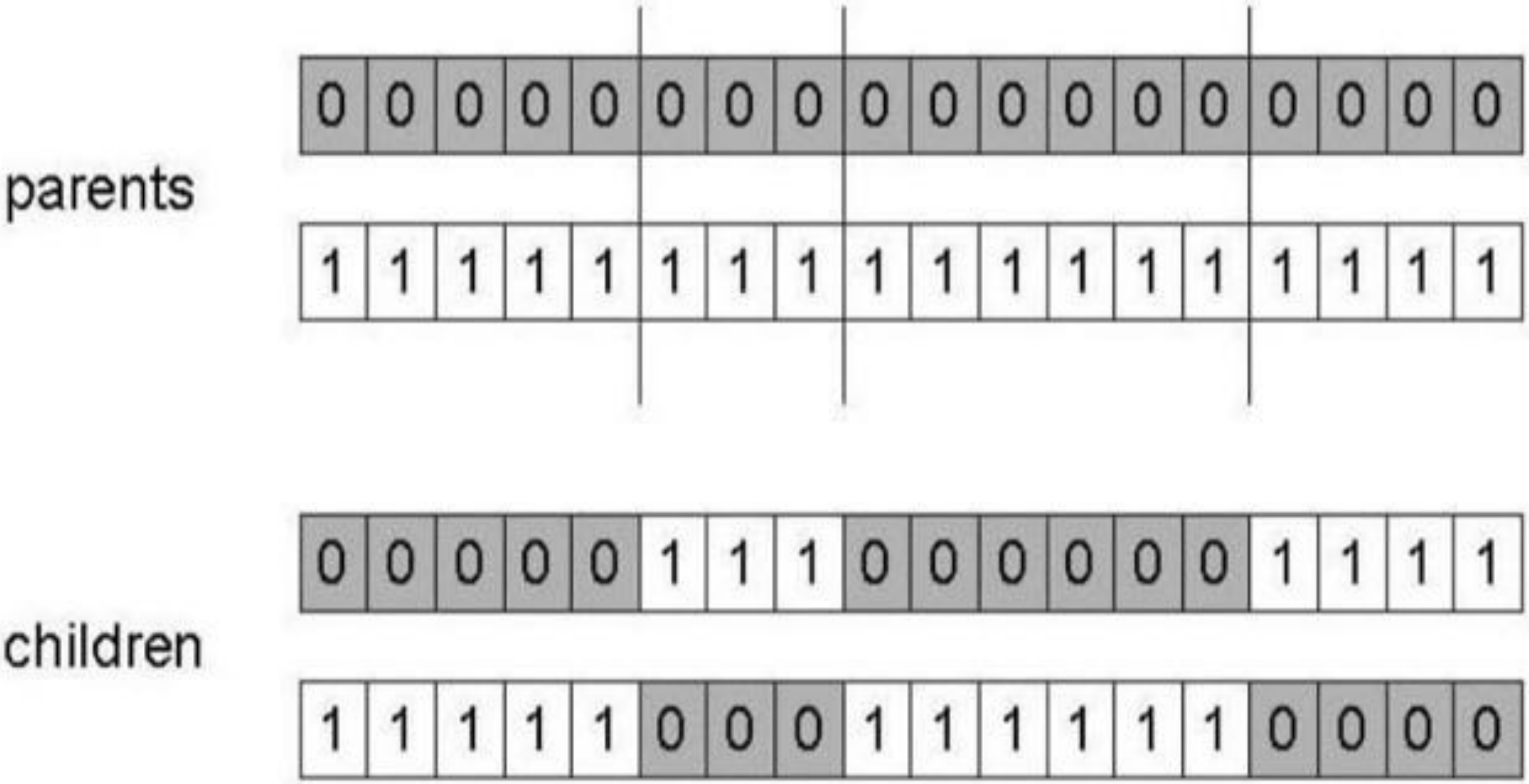
Two-Point Crossover

- But with one point cross over , the head and tail of one chromosome cannot be passed together to the offspring
- If both the head and tail of chromosome contain good genetic information, none of the offspring obtain directly with the one point crossover
- Using two point crossover one can overcome these drawbacks so it is better than one point cross over

Multi-Point Crossover/

(N-Point crossover)

- Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



parents

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

children

0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1

1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0

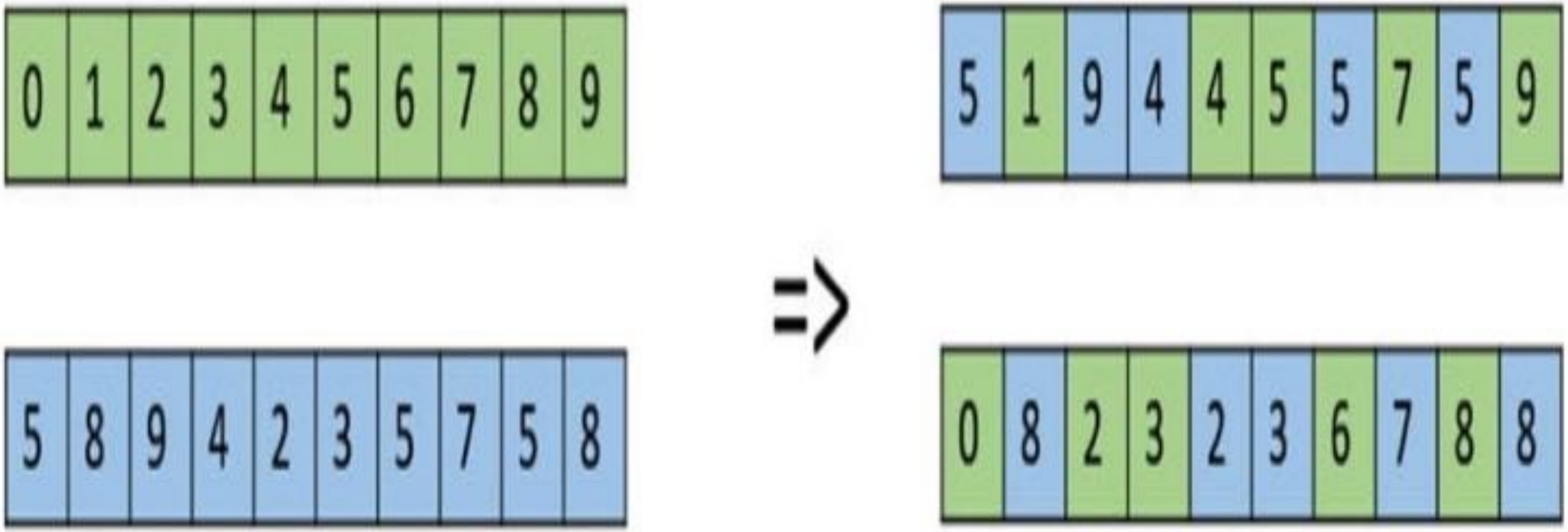
Uniform Crossover

PARENT 1	1 0 1 1 0 0 1 1
PARENT 2	0 0 0 1 1 0 1 0
MASK	1 1 0 1 0 1 1 0
CHILD 1	1 0 0 1 1 0 1 0
CHILD 2	0 0 1 1 0 0 1 1

Uniform Crossover

- Each gene in the offspring is created by copying the corresponding gene from the one or other parent chosen according to the random generated binary crossover mask of same length as the chromosome
- Where there is a 1 in the crossover point, the gene is copied from the first parent
- Where there is a 0 in the mask , the gene is copied from the second parent
- A new crossover mask is generated for each parent pair
- Offspring therefore contain a mixture of gene from each parent
- The number of effective crossing point is not fixed , but will be the average of L/2 where L is the chromosome length

Uniform Crossover



0 1 2 3 4 5 6 7 8 9

5 8 9 4 2 3 5 7 5 8

=>

5 1 9 4 4 5 5 7 5 9

0 8 2 3 2 3 6 7 8 8

Genetic Algorithm – Uniform Crossover

Generate uniformly random number.

X1 = **0** 1 1 0 0 0 **1** 0 1 0

X2 = **1** 1 0 0 0 0 **0** 1 1 1

Uniformly generated = **1** 0 0 0 0 0 **1** 0 0 0

As a result, the new population becomes,

X1 = **1** 1 1 0 0 0 **0** 0 1 0

X2 = **0** 1 0 0 0 0 **1** 1 1 1

Three- Parent Crossover

PARENT 1	1 1 0 1 0 0 0 1
PARENT 2	0 1 1 0 1 0 0 1
PARENT 3	0 1 1 0 1 1 0 0
CHILD	0 1 1 0 1 0 0 1

Three- Parent Crossover

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

Crossover with Reduced Surrogate

The reduced surrogate operator constrains crossover to always produce new individuals wherever possible.

This is implemented by restricting the location of crossover points such that crossover points only occur where gene values differ.

Shuffle Crossover

Shuffle crossover is related to uniform crossover.

A single crossover position (as in single-point crossover) is selected.

But before the variables are exchanged, they are randomly shuffled in both parents.

After recombination, the variables in the offspring are un shuffled.

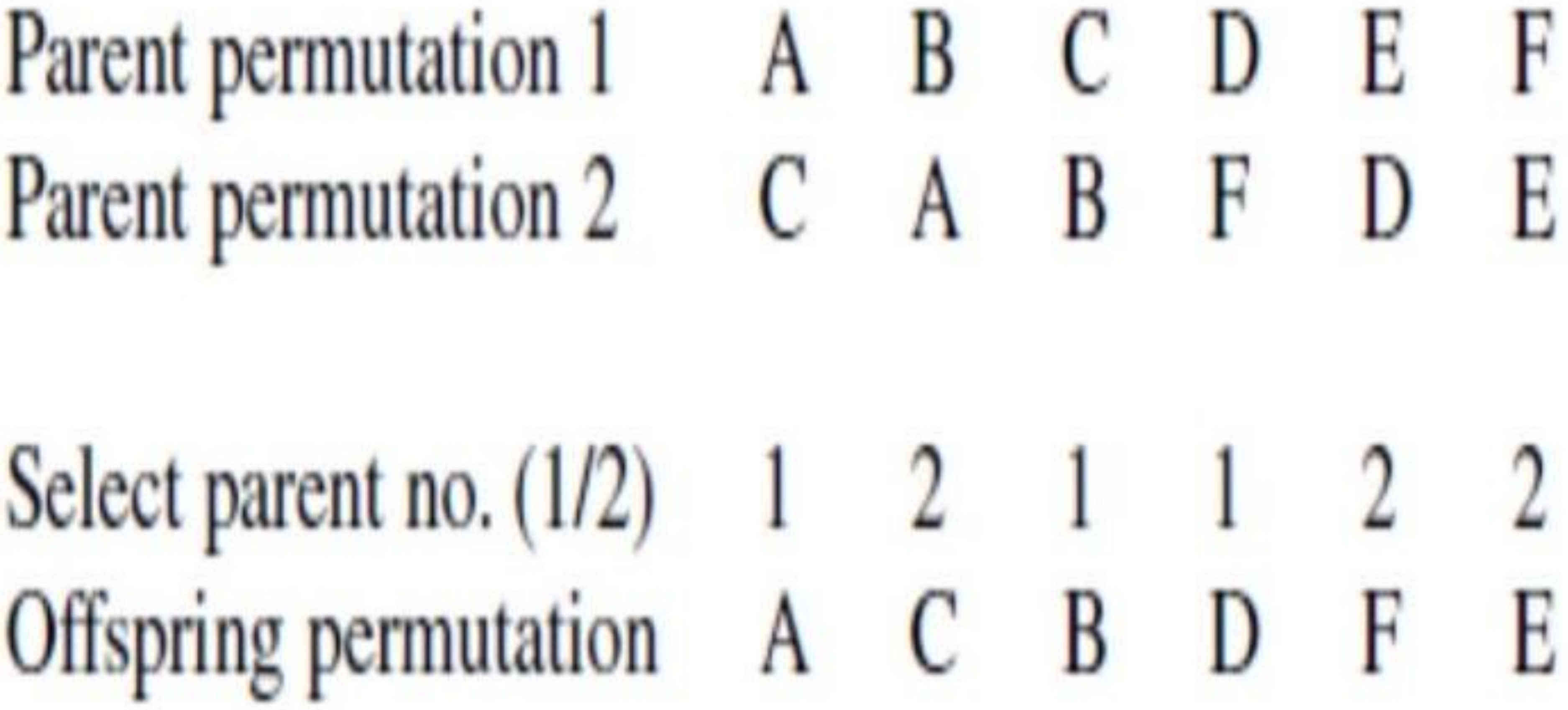
This removes positional bias as the variables are randomly reassigned each time crossover is performed.

Precedence Preservative Crossover (PPX)

- Developed for vehicle routing problems
- The operator passes on precedence relation of operations given in two parental permutations to one offspring at the same rate, while no new precedence relations are introduced

1. A vector of length Sigma, sub i=1to mi, representing the number of operations involved in the problem, is randomly filled with elements of the set {1, 2}.
2. This vector defines the order in which the operations are successively drawn from parent 1 and parent 2.
3. We can also consider the parent and offspring permutations as lists, for which the operations ‘append’ and ‘delete’ are defined.
4. First we start by initializing an empty offspring.
5. The leftmost operation in one of the two parents is selected in accordance with the order of parents given in the vector.
6. After an operation is selected it is deleted in both parents.
7. Finally the selected operation is appended to the offspring.
8. This step is repeated until both parents are empty and the offspring contains all operations involved.

Precedence Preservative Crossover (PPX)



Parent permutation 1	A	B	C	D	E	F
Parent permutation 2	C	A	B	F	D	E
Select parent no. (1/2)	1	2	1	1	2	2
Offspring permutation	A	C	B	D	F	E

Ordered Crossover

- Ordered two-point crossover is used when the problem is of order based, for example in U-shaped assembly line balancing etc.
- Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and right portion.
- The ordered two-point crossover behaves in the following way:

  child 1 inherits its left and right section from parent 1, and its middle section is determined by the genes in the middle section of parent 1 in the order in which the values appear in parent 2.
- A similar process is applied to determine child 2.

$$\begin{array}{ll} \text{Parent } 1: 4\ \ 2\,|\,1\ \ 3\,|\,6\ \ 5 & \text{Child } 1: 4\ \ 2\,|\,3\ \ 1\,|\,6\ \ 5 \\ \text{Parent } 2: 2\ \ 3\,|\,1\ \ 4\,|\,5\ \ 6 & \text{Child } 2: 2\ \ 3\,|\,4\ \ 1\,|\,5\ \ 6 \end{array}$$

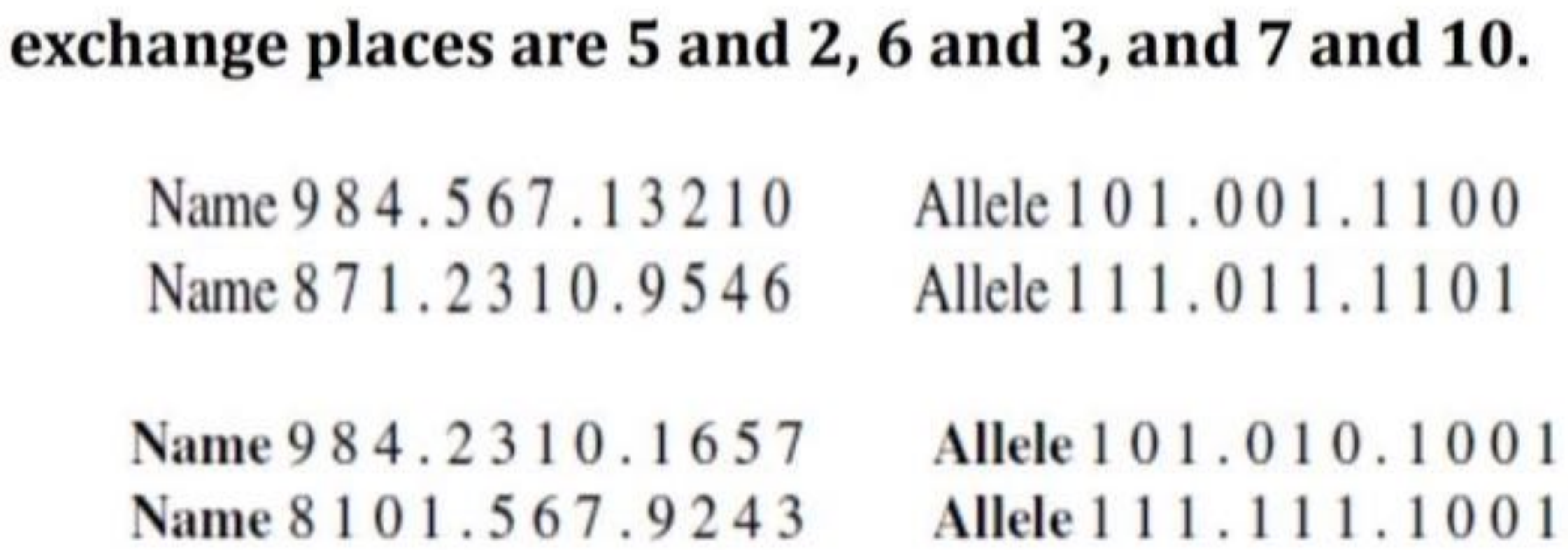
Partially Matched Crossover (PMX)

- PMX can be applied usefully in the TSP.
- Indeed, TSP chromosomes are simply sequences of integers, where each integer represents a different city and the order represents the time at which a city is visited.
- Under this representation, known as permutation encoding, we are only interested in labels .
- It may be viewed as a crossover of permutations that guarantees that all positions are found exactly once in each offspring, i.e. both 3 offspring receive a full complement of genes, followed by the corresponding filling in of alleles from their parents.

Partially Matched Crossover (PMX)

- PMX proceeds as follows:
  1. The two chromosomes are aligned.
  2. Two crossing sites are selected uniformly at random along the strings, defining a matching section
  3. The matching section is used to effect a cross through position by-position exchange operation
  4. Alleles are moved to their new positions in the offspring

- Consider the two strings shown in Fig
- Where, the dots mark the selected cross points.
- The matching section defines the position-wise exchanges that must take place in both parents to produce the offspring.
- The exchanges are read from the matching section of one chromosome to that of the other.
- In the example, the numbers that exchange places are 5 and 2, 6 and 3, and 7 and 10.
- The resulting offspring are as shown



exchange places are 5 and 2, 6 and 3, and 7 and 10.

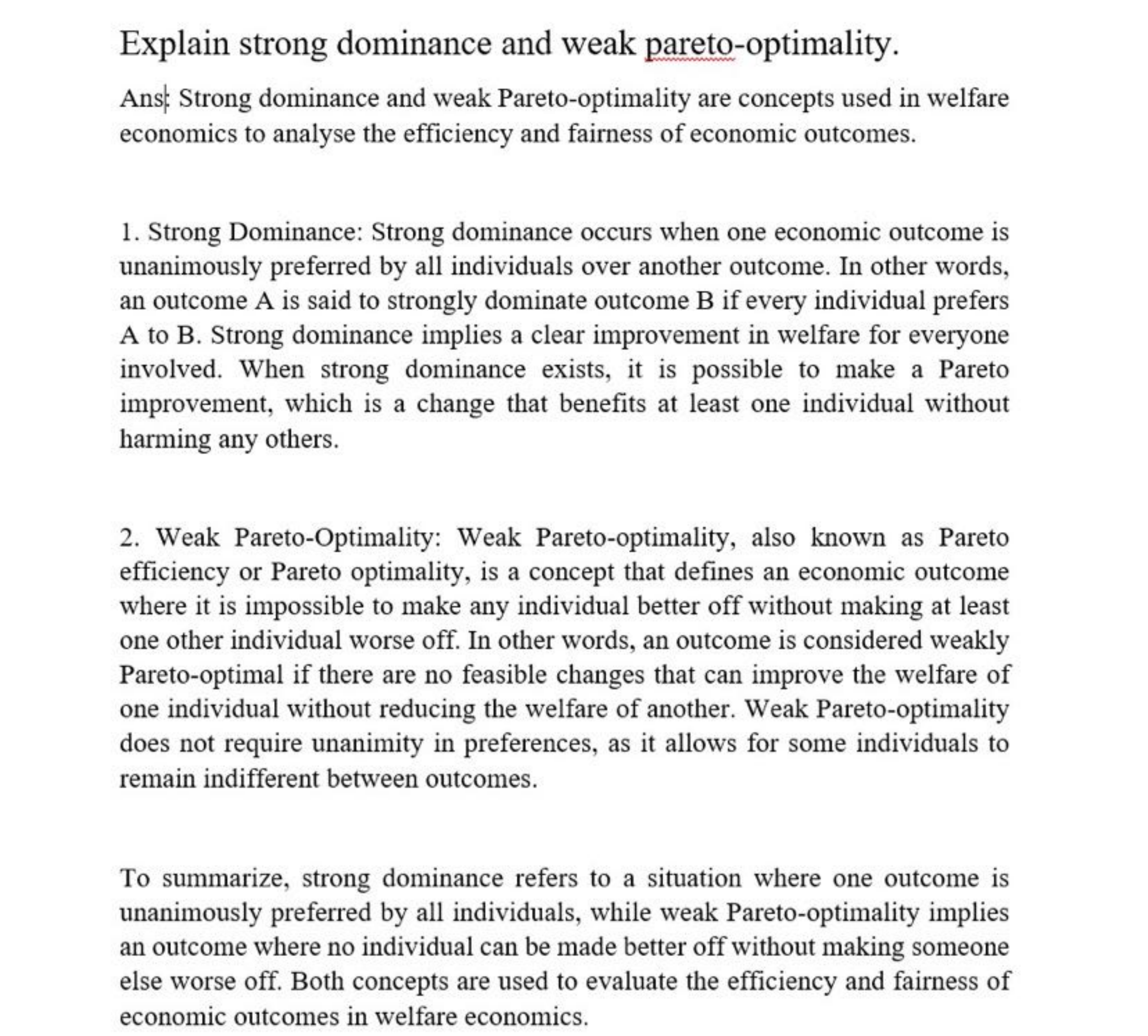
Name 9 8 4 . 5 6 7 . 1 3 2 1 0 Allele 1 0 1 . 0 0 1 . 1 1 0 0
Name 8 7 1 . 2 3 1 0 . 9 5 4 6 Allele 1 1 1 . 0 1 1 . 1 1 0 1

Name 9 8 4 . 2 3 1 0 . 1 6 5 7 **Allele** 1 0 1 . 0 1 0 . 1 0 0 1
Name 8 1 0 1 . 5 6 7 . 9 2 4 3 **Allele** 1 1 1 . 1 1 1 . 1 0 0 1

Parent A	4 8 7 | 3 6 5 | 1 10 9 2
Parent B	3 1 4 | 2 7 9 | 10 8 6 5

<table>
<tr><td>Parent A</td><td>4 8 7 | 3 6 5 | 1 10 9 2</td></tr>
<tr><td>Parent B</td><td>3 1 4 | 2 7 9 | 10 8 6 5</td></tr>
</table>

<table>
<tr><td>Child A</td><td>4 8 6 2 7 9 1 10 5 3</td></tr>
<tr><td>Child B</td><td>2 1 4 3 6 5 10 8 7 9</td></tr>
</table>



Explain strong dominance and weak pareto-optimality.

Ans: Strong dominance and weak Pareto-optimality are concepts used in welfare economics to analyse the efficiency and fairness of economic outcomes.

1. Strong Dominance: Strong dominance occurs when one economic outcome is unanimously preferred by all individuals over another outcome. In other words, an outcome A is said to strongly dominate outcome B if every individual prefers A to B. Strong dominance implies a clear improvement in welfare for everyone involved. When strong dominance exists, it is possible to make a Pareto improvement, which is a change that benefits at least one individual without harming any others.

2. Weak Pareto-Optimality: Weak Pareto-optimality, also known as Pareto efficiency or Pareto optimality, is a concept that defines an economic outcome where it is impossible to make any individual better off without making at least one other individual worse off. In other words, an outcome is considered weakly Pareto-optimal if there are no feasible changes that can improve the welfare of one individual without reducing the welfare of another. Weak Pareto-optimality does not require unanimity in preferences, as it allows for some individuals to remain indifferent between outcomes.

To summarize, strong dominance refers to a situation where one outcome is unanimously preferred by all individuals, while weak Pareto-optimality implies an outcome where no individual can be made better off without making someone else worse off. Both concepts are used to evaluate the efficiency and fairness of economic outcomes in welfare economics.

Course Outcome 4(CO4)

Q3. using genetic algorithm with Roulette wheel selection method maximize the function $f(x) = x2$ over $\{0,1,2, \ldots 3\}$ with initial x values of $(13,24,8,19)$. Show one crossover & mutation

→ Maximize $f(x) = x^2$; where x is permitted to vary between 0 and 31

1. coding of decision variables as some finite length string

   X as binary unsigned integer of length 5

   $[0, 31] = [00000; 11111]$

2. Constant setting

   $P_{mutation} = 0.0333$

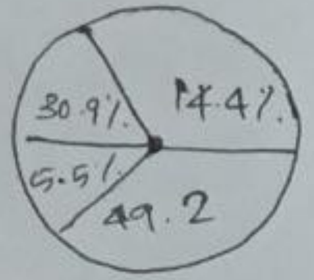
   $P_{cross} = 0.6$

   Population size = 30

3. Select initial population at random:

String number	Initial population	Xvalue	$f(x)$	pselect $\frac{f_i}{\Sigma f}$	Expected Count $\frac{f_i}{f}$	Actual Count (Roulette Wheel)
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1

4. Reproduction: Select mating pool by spinning roulette wheel 4 times



weighted Roulette wheel

	pselect
01101	0.14
11000	0.49
01000	0.06
10011	0.31

5. Crossover - strings are mated randomly using coin tosses to pair the couples.

- mated string couples crossover using coin tosses to select the crossing site

String number	Mating pool after reproduction	Mate (randomly selected)	Crossover site (random)	New population	X-value	$f(x)$ $=x^2$
1	0110|1	2	4	01100	12	144
2	1100|0	1	4	11001	25	625
3	11|000	4	2	11011	27	729
4	10|011	3	2	100000	16	256

CO3 Question 2(i)

7. Given 2 universes $X = \{x1, x2, x3, x4, x5)$ and $Y = \{y1, y2, y3, y4, y5\}$, the fuzzy set A defined on Y are given below.

$$A = \left\{ \frac{0.4}{x_1} + \frac{0.7}{x_2} + \frac{1}{x_3} + \frac{0.8}{x_4} + \frac{0.6}{x_5} \right\}$$

$$B = \left\{ \frac{0.2}{y_1} + \frac{0.6}{y_2} + \frac{1}{y_3} + \frac{0.9}{y_4} + \frac{0.7}{y_5} \right\}$$

(i) Find the relation $R = A \times B$

→

$$R = \begin{array}{c|ccccc} & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline x_1 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 \\ x_2 & 0.2 & 0.6 & 0.7 & 0.7 & 0.7 \\ x_3 & 0.2 & 0.6 & 1 & 0.9 & 0.7 \\ x_4 & 0.2 & 0.6 & 0.8 & 0.8 & 0.7 \\ x_5 & 0.2 & 0.6 & 0.6 & 0.6 & 0.6 \end{array}$$

Procedure for finding a non-Dominated set

Finding the non-dominated set of solutions from a given set of solutions is similar in principle to finding the minimum of a set of real numbers. In the case of finding the non-dominated set, the dominance relation can be used to identify the better of two given solutions. Like the existence of different algorithms for finding the minimum number from a finite set (Cormen et al., 1990), many approaches have also been suggested for finding the non-dominated set from a given population of solutions. Although many approaches are possible, they would usually have different computational complexities. To make matters simple, we will discuss here different procedures.

Approach 1: Naive and Slow

In this approach, each solution is *i* compared with every other solution in the population to check if it is dominated by any solution in the population. If the solution *i* is found to be dominated by any solution, this means that there exists at least one solution in the population which is better than *i* in all objectives. Hence the solution *i* cannot belong to the non-dominated set. We mark a flag against the solution *i* to denote that it does not belong to the non-dominated set. However, if no solution is found to dominate solution *i* it is a member of the non-dominated set. This is how any other solution in the population can be checked to see if it belongs to the non-dominated set. The following approach describes a step-by-step procedure for finding the non-dominated set in a given set P of size N.

Identifying the Non-Dominated Set: Approach I

Step 1 Set solution counter *i*=1 and create an empty non-dominated set P'.

Step 2 For a solution j ∈ P (but j!=i), check if solution j dominates solution i. If yes, go to Step 4.

Step 3 If more solutions are left in P, increment j by one and go to Step 2; otherwise, get P=P' U{i}.

Step 4 Increment i by one. If i<=N, go to Step 2;otherwise stop and declare P' as the non-dominated set.

Approach 2: Continuously Updated

This approach is similar in principle to Approach 1, except that better bookkeeping is used to make the algorithm faster. In this approach, every solution from the population is checked with a partially filled population for domination. To start with, the first solution from the population is kept in an empty set P’. Thereafter, each solution i (the second solution onwards) is compared with all members of the set P i, one by one. If the solution i dominates any member of P’, then that solution is removed from P’. In this way non-members of the non-dominated solutions get deleted from P’ Otherwise If solution i is dominated by any member of P’, the solution i is ignored.If solution i is not dominated by any member of P’, it is entered in P’. This is how the set P’ grows with non-dominated solutions. When all solutions Of the population are checked, the remaining members of P’ constitute the non-dominated set.

Identifying the Non-Dominated Set: Approach 2

Step 1 Initialize P’ = {1} Set solution counter i = 2.

Step 2 Set j = 1.

Step 3 Compare solution i with j from P’ for domination.

Step 4 If i dominates j, delete the j-th member from P’ or update P’ = {P’\{p’^(j)}. If j <|P’|, increment j by one and then go to Step 3. Otherwise, go to Step 5. Alternatively, if the j-th member of P’ dominates i, increment i by one and then go to Step 2.

Step 5 Insert i in P’ or update P’ = P’ U{i}. If i < N, increment i by one and go to Step 2. Otherwise, stop and declare P’ as the non-dominated set.

GENETIC NEURO HYBRID SYSTEMS

A neuro-genetic hybrid or a genetic-neuro-hybrid system is one in which a neural network employs a genetic algorithm to optimize its structural parameters that define its architecture. Neural networks learn and execute different tasks using several example, classify phenomena, and model nonlinear relationships; that is neural networks solve problems by self learning and self-organizing, On the other hand, genetic algorithms present themselves as a potential solution for the optimization of parameters of neural networks.

Certain properties of genetic neuro-hybrid systems are as follows:

1. The parameters of neural networks are encoded by genetic algorithms as a string of properties of the network, that is, chromosomes. A large population of chromosomes is generated, which represent the many possible parameter sets for the given neural network.

2. Genetic Algorithm- Neural Network, or GANN, has the ability to locate the neighbourhood of the optimal solution quickly, compared to other conventional search strategies.

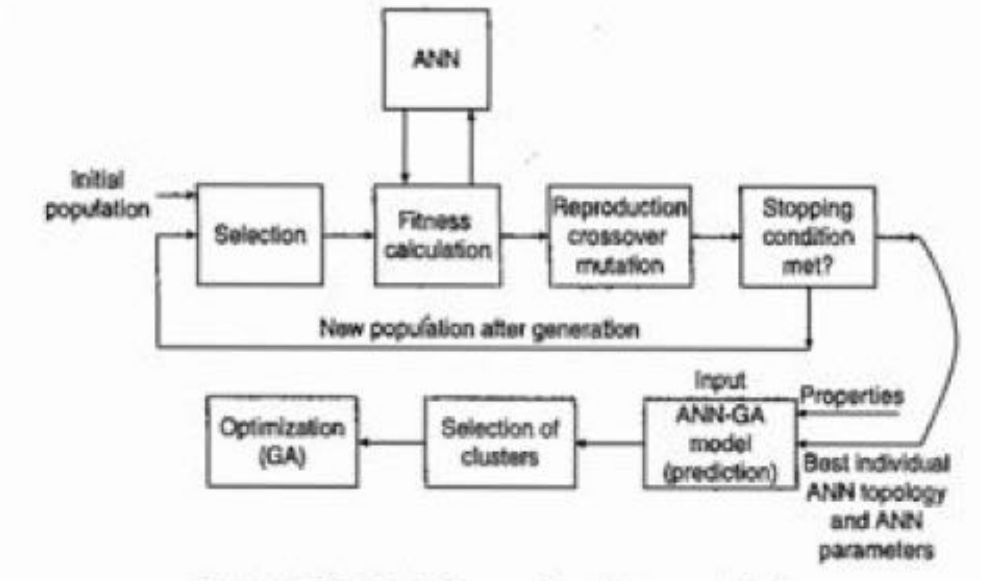

ANN
Initial population
Selection
Fitness calculation
Reproduction crossover mutation
Stopping condition met?
New population after generation
Input
Properties
ANN-GA model (prediction)
Best individual ANN topology and ANN parameters
Selection of clusters
Optimization (GA)

Figure 16-5 Block diagram of genetic–neuro hybrids.

Genetic Algorithm Based Back-Propagation Network (BPN)

BPN is a method of reaching multi-layer neural networks how to perform a given task. Here learning occurs during this training phase.

The limitations of BPN are as follows:

1. BPN do not have the ability to recognize new patterns; they can recognize patterns similar to those they have learnt.

2. They must be sufficiently trained so that enough general features applicable to both seen and unseen instances can be extracted; there may be undesirable effects due to over training the network.

Also, it may be noted that the BPN determines its weight based on gradient search technique and hence it may encounter a local minima problem. Though genetic algorithms do not guarantee to find global optimum solution, they are good in quickly finding good acceptable solutions. Thus, hybridization of BPN with genetic algorithm is expected to provide many advantages compared to what they alone can.

However, before a genetic algorithm is executed,

I. A suitable coding for the problem has to be devised.

2. A fitness function has to be formulated.

3. Parents have to be selected for reproduction and then crossed over to generate offspring.

Coding

Assume a BPN configuration n-l-m where n is the number of neurons in the input layer, l is the number of neurons in the hidden layer and m is the number of output layer neurons. The number of weights to be determined is given by

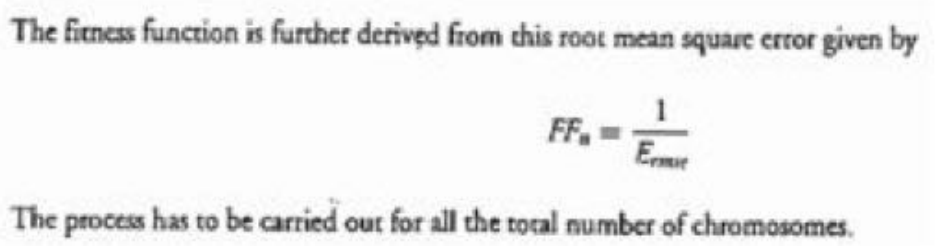
$$(n+m)\ l$$

Weight Extraction

In order to determine the fitness values, weights are extracted from each chromosome.

Fitness Function

A fitness has to be formulated for each and every problem to be solved.



The fitness function is further derived from this root mean square error given by

$$FF_n = \frac{1}{E_{rmse}}$$

The process has to be carried out for all the total number of chromosomes.

Reproduction of Offspring

In this process, before the parents produce the offspring with better fitness, the mating pool has to be formulated. This is accomplished by neglecting the chromosome with minimum fitness and replacing it with a chromosome having maximum fitness. In other words, the fittest individuals among the chromosomes will be given more chances to participate in the generations and the worst individuals will be eliminated. Once the mating pool is formulated, parent pairs are selected randomly and the chromosomes of respective pairs are combined using crossover technique to reproduce offspring. The selection operator is suitably used to select the best parent to participate in the reproduction process.

Convergence

The convergence for genetic algorithm is the number of generations with which the fitness value increases towards the global optimum. Convergence is the progression towards increasing uniformity. When about 95% of the individuals in the population share the same fitness value then we say that a population has converged.

ADVANTAGES

The various advantages of neuro-genetic hybrid are as follows:

- GA performs optimization of neural network parameters with simplicity, ease of operation, minimal requirements and global perspective.
- GA helps to find out complex structure of ANN for given input and the output data set by using its learning rule as a fitness function.
- Hybrid approach ensembles a powerful model that could significantly improve the predictability of the system under construction.

The hybrid approach can be applied to several applications, which include: load forecasting, stock forecasting, cost optimization in textile industries, medical diagnosis, face recognition, multi-processor scheduling, job shop scheduling, and so on.

11)Illustrate the various types of cross over with suitable examples?

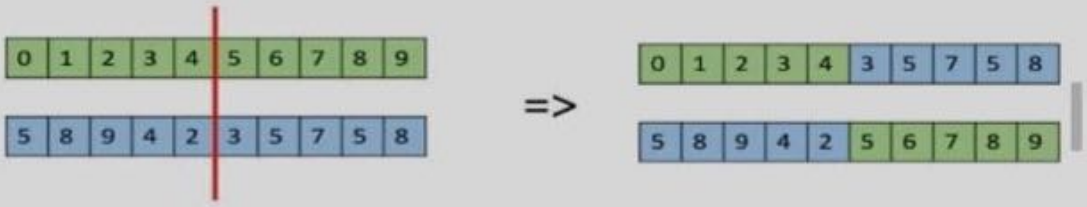
Answer:

Various crossover techniques are:

1. Single- point Crossover
2. Two- point crossover
3. Multipoint cross over/N point cross over
4. Uniform Crossover
5. Three parent Crossover
6. Crossover with reduced Surrogate
7. Shuffle crossover
8. Precedence Preservative Crossover
9. Ordered crossover
10. Partially matched crossover

1)Single-Point Crossover

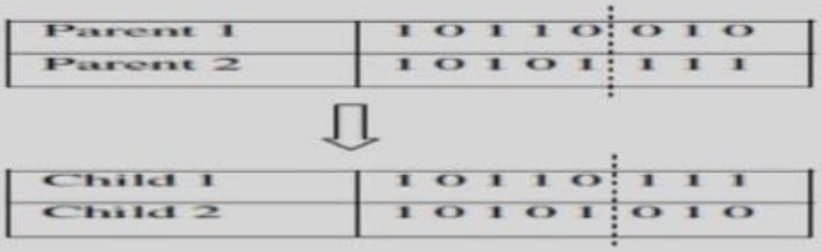
- In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.



0 1 2 3 4 5 6 7 8 9 => 0 1 2 3 4 3 5 7 5 8

5 8 9 4 2 3 5 7 5 8 => 5 8 9 4 2 5 6 7 8 9

Arsha J K, AP,Computer Science and Engineering, VJCET, Vazhakulam



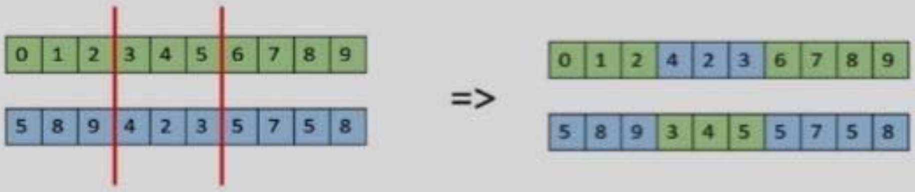
Parent 1	1 0 1 1 0 ⋮ 0 1 0
Parent 2	1 0 1 0 1 ⋮ 1 1 1

⇩

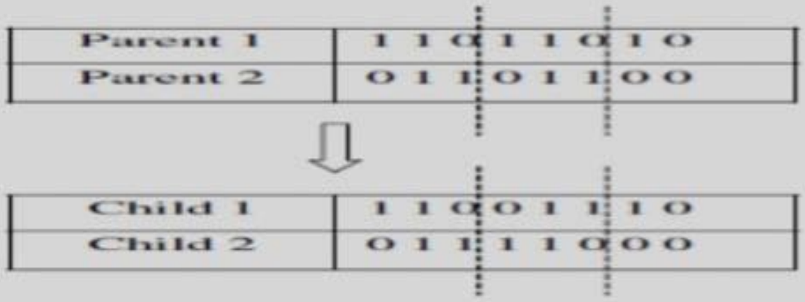
Child 1	1 0 1 1 0 ⋮ 1 1 1
Child 2	1 0 1 0 1 ⋮ 0 1 0

2)Two-Point Crossover

• Two crossover points are chosen and the content between these points are exchanged between two matted parents.



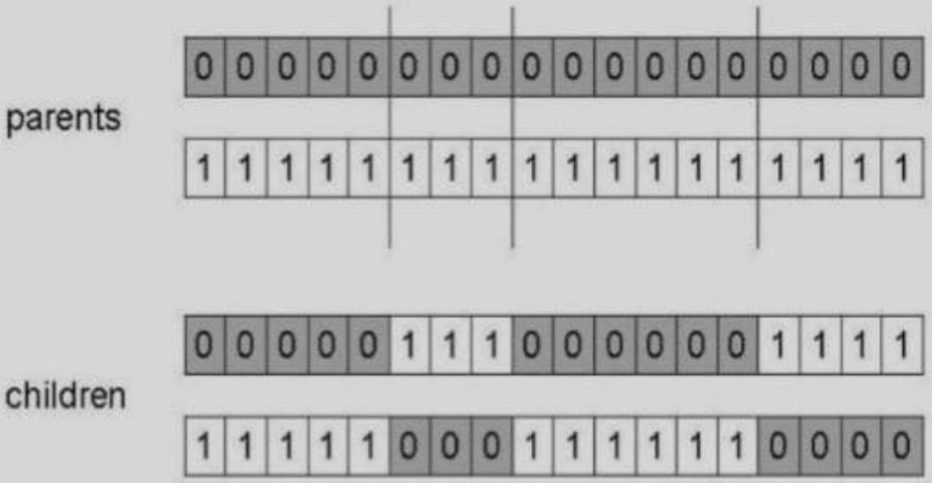
Arsha J K, AP/Computer Science and Engineering, VJCET, Vazhakulam



- But with one point cross over , the head and tail of one chromosome cannot be passed together to the offspring
- If both the head and tail of chromosome contain good genetic information, none of the offspring obtain directly with the one point crossover
- Using two point crossover one can overcome these drawbacks so it is better than one point cross over

3)Multi-Point Crossover/ (N-Point crossover)

• Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



4)Uniform Crossover

- Each gene in the offspring is created by copying the corresponding gene from the one or other parent chosen according to the random generated binary crossover mask of same length as the chromosome

- Where there is a 1 in the crossover point, the gene is copied from the first parent

- Where there is a 0 in the mask , the gene is copied from the second parent

- A new crossover mask is generated for each parent pair

- Offspring therefore contain a mixture of gene from each parent

- The number of effective crossing point is not fixed , but will be the average of L/2 where L is the chromosome length

PARENT 1	10110011
PARENT 2	00011010
MASK	11010110
CHILD 1	10011010
CHILD 2	00110011

5)Three- Parent Crossover

- Three parents are randomly chosen

- Each bit of the first parent is compared with the bit of second parent.

- If both are same, the bit is taken for the offspring

- Otherwise bit from the third parent is taken for the offspring.

PARENT 1	11010001
PARENT 2	01101001
PARENT 3	01101100
CHILD	01101001

6) Crossover with Reduced Surrogate

The reduced surrogate operator constrains crossover to always produce new individuals wherever possible. This is implemented by restricting the location of crossover points such that crossover points only occur where gene values differ

7) Shuffle Crossover

Shuffle crossover is related to uniform crossover. A single crossover position (as in single-point crossover) is selected. But before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are un shuffled. This removes positional bias as the variables are randomly reassigned each time crossover is performed.

8) Precedence Preservative Crossover

Developed for vehicle routing problems

The operator passes on precedence relation of operations given in two parental permutations to one offspring at the same rate, while no new precedence relations are introduced.

9) Ordered Crossover

Ordered two-point crossover is used when the problem is of order based, for example in U-shaped assembly line balancing etc. • Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and right portion. • The ordered two-point crossover behaves in the following way: child 1 inherits its left and right section from parent 1, and its middle section is determined by the genes in the middle section of parent 1 in the order in which the values appear in parent 2. • A similar process is applied to determine child 2.

10) Partially Matched Crossover

PMX proceeds as follows:

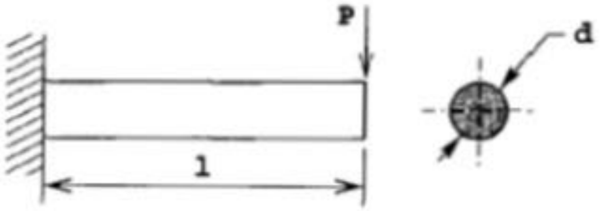
1. The two chromosomes are aligned.

2. Two crossing sites are selected uniformly at random along the strings, defining a matching section 3. The matching section is used to effect a cross through position by-position exchange operation

4. Alleles are moved to their new positions in the offspring.

Q) Illustrate Pareto-Optimal Solutions

To illustrate the concept of Pareto-optimal solutions.
Consider a cantilever design problem with 2 decision variables, i.e, diameter(d) and length(l).The beam has to carry an end load P. Let us also consider two conflicting objectives of design, i.e, minimization of weight f1 and minimization of end deflection f2. The first objective will resort to an optimum solution having the smaller dimensions of d and l, so that the overall weight of the beam is minimum. Since the dimensions are small, the beam will not be adequately rigid and the nd deflection of the beam will be large. On the other hand, if the beam is minimised for the end deflection, the dimensions of the beams are expected to be large, thereby making the weight of the beam large.2 constraints are considered:



two constraints: the developed maximum stress $\sigma_{\max}$ is less than the allowable strength S_y and the end deflection δ is smaller than a specified limit $\delta_{\max}$. With all of the above considerations, the following two-objective optimization problem is formulated as follows:

$$\left.\begin{array}{ll} \text{Minimize} & f_1(d,l) = \rho\frac{\pi d^2}{4}l, \\ \text{Minimize} & f_2(d,l) = \delta = \frac{64Pl^3}{3E\pi d^4}, \\ \text{subject to} & \sigma_{\max} \le S_y, \\ & \delta \le \delta_{\max}, \end{array}\right\} \tag{2.3}$$

where the maximum stress is calculated as follows:

$$\sigma_{\max} = \frac{32Pl}{\pi d^3}. \tag{2.4}$$

The following parameter values are used:

$$\begin{array}{lll} \rho = 7800\ \text{kg/m}^3, & P = 1\ \text{kN}, & E = 207\ \text{GPa}, \\ S_y = 300\ \text{MPa}, & \delta_{\max} = 5\ \text{mm}. & \end{array}$$

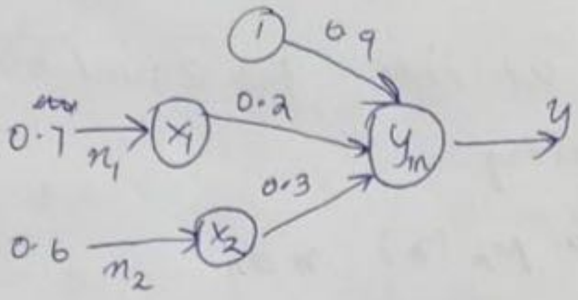
24/5/23

Soft computing ① Team 1

Diya Anna Varghese
Mehanoor Bashir
Tiny Tomy
Veena Regikumar

course outcome 1 (CO1):

1) Describe the necessity of activation function? Examine the various aspects of sigmoidal activation Function List the drawbacks calculate the net output of the following neural network using the bipolar and binary sigmoidal activation Function.



Ans:

The activation Function is applied over the net input to calculate the output of an ANN.

The information processing of a processing element can be viewed as consists of two major part input and output.

An Activation function decides whether a neuron should be activated or not.. it is necessary to prevent linearity. it is introducing non-linearity. This will allows neural n/w to develop complex representation and functions based on the input that would not be possible with a simple linear regression model

- it helps in making a given model learn and execute difficult task

Sigmoidal Function

They are widely used in back propagation nets becouse of the relationship between the value of the functions at a point and the value of the derivative at that point which reduce the computational burden data training

These are of two types;

(1) Binary sigmoidal Function
(2) Bipolar sigmoidal Function

- Binary sigmoidal Function; also termed as logistic sigmoid function or unipolar sigmoid function

It is defined as, $f(x) = \frac{1}{1+e^{-\lambda x}}$

λ is steepness parameter

range is from 0 to 1

Bipolar sigmoidal function, It is defined as

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1 = \frac{1-e^{-\lambda x}}{1+e^{-\lambda x}}$$

range is from -1 and +1

It is closely related to hyperbolic tangent function written as,

$$h(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{1-e^{-2x}}{1+e^{-2x}}$$

4

i) For binary sigmoid activation function

$y = f(y_{in})$

$= \frac{1}{1+e^{y_{in}}} = \frac{1}{1+e^{1.22}} = \frac{1}{4.38} = \underline{\underline{0.228}}$

(ii) For bipolar sigmoid activation function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1$$

$$= \frac{2}{1+e^{-1.22}} - 1 = \frac{2}{1.29} - 1 = \underline{\underline{0.550}}$$

TEAM 1.

Diya Anna Varghese	23
Mehanoor Basheer	41
Tincy Tomy	62
Veena Regikumar	63.

GROUP-5

course outcome 3

1 (a) $X = \left\{ \frac{1.0}{1500} + \frac{0.8}{2175} + \frac{0.6}{7000} + \frac{0.5}{12750} + \frac{0.3}{16500} + \frac{0.1}{20000} \right\}$

$Y = \left\{ \frac{0.2}{20} + \frac{0.4}{25} + \frac{0.5}{32} + \frac{1.0}{50} + \frac{0.6}{90} + \frac{0.3}{105} \right\}$

$$R = X \times Y = \begin{bmatrix} 0.2 & 0.4 & 0.5 & 1.0 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.8 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.6 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.5 & 0.5 & 0.3 \\ 0.2 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

1 $X = \left\{\frac{1.0}{1500} + \frac{0.8}{2175} + \frac{0.6}{7000} + \frac{0.5}{12750} + \frac{0.3}{16500} + \frac{0.1}{20000}\right\}$

$Y = \left\{\frac{0.2}{20} + \frac{0.4}{25} + \frac{0.5}{32} + \frac{1.0}{50} + \frac{0.6}{70} + \frac{0.3}{105}\right\}$

Cartesian pdt

$$\tilde{R} = \underset{\sim}{X} \times \underset{\sim}{Y} = \begin{array}{c|cccccc} & 20 & 25 & 32 & 50 & 70 & 105 \\ \hline 1500 & 0.2 & 0.4 & 0.5 & 1.0 & 0.6 & 0.3 \\ 2175 & 0.2 & 0.4 & 0.5 & 0.8 & 0.6 & 0.3 \\ 7000 & 0.2 & 0.4 & 0.5 & 0.6 & 0.6 & 0.3 \\ 12750 & 0.2 & 0.4 & 0.5 & 0.5 & 0.5 & 0.3 \\ 16500 & 0.2 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 20000 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{array}$$

$Z = \left\{\frac{0.4}{1500} + \frac{0.5}{2175} + \frac{0.6}{7000} + \frac{0.8}{12750} + \frac{0.9}{16500} + \frac{1.0}{20000}\right\}$, 0.32 , 0.27

max min $S = Z \circ R$

$$Z = \begin{bmatrix} 0.4 & 0.5 & 0.6 & 0.8 & 0.9 & 1.0 \end{bmatrix} \quad \begin{bmatrix} 0.2 & 0.4 & 0.5 & 1.0 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.8 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.6 & 0.6 & 0.3 \\ 0.2 & 0.4 & 0.5 & 0.5 & 0.5 & 0.3 \\ 0.2 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

0.08 0.1 0.12 0.16 0.18 0.1

0.2 0.2 0.2 0.2 0.2 0.1

0.4 0.4 0.4 0.4 0.3 0.1

0.4 0.5 0.5 0.5 0.3 0.1

~~0.8 0.5~~ 0.6 0.5 0.3 0.1

0.4 0.5

0.4 0.5 0.6 0.5 0.3 0.1

0.1 0.3 0.1 0.3 0.3 0.1

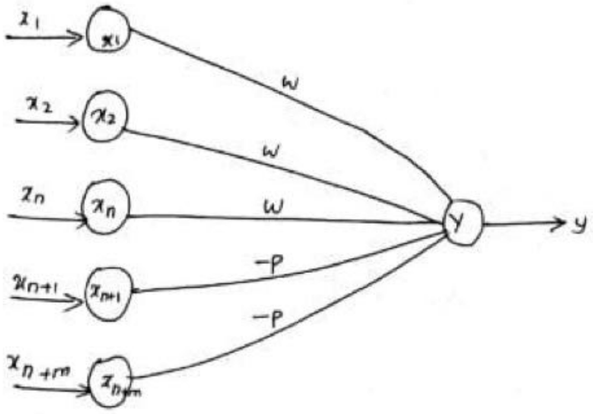
$$\begin{bmatrix} 0.2 & 0.4 & 0.5 & 0.6 & 0.6 & 0.3 \end{bmatrix}$$

$$\begin{bmatrix} 0.18 & 0.32 & 0.4 & 0.4 & 0.4 & 0.27 \end{bmatrix}$$

Course Outcome

1. Architecture of MP neuron

- Mcculloch - Pitts neuron - 1943.
- It is connected by directed weighted path.
- Activation of MP neuron is binary.
- Weights can be excitatory (+ve)
  inhibitory (-ve)
- Thresholds (θ) plays a major rule - fixed threshold for each neuron. If net i/p > threshold ⇒ neuron fires
- Used in logic functions.



- It is excitatory with weight $w > 0$
  inhibitory with weight $-p$ $p < 0$

x_1 to x_n possess excitatory connection

$x_{n+1} - x_{n+m}$ — Inhibitory weighted interconnections

Activation function is

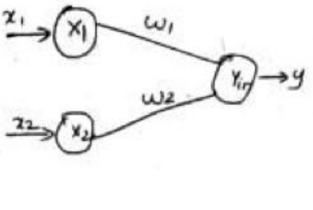
$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < 0 \end{cases}$$

$$\boxed{\theta > nw - p}$$

2. NOT AND / NAND function using MP neuron (binary i/p)

Truth table

x_1	x_2	y_{in}
0	0	1
0	1	1
1	0	1
1	1	0.



$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta. \\ 0 & \text{if } y_{in} < \theta \end{cases} \qquad y_{in} = x_1 w_1 + x_2 w_2$$

Case 1 : $\omega_1 = \omega_2 = 1$

$(0,0)$	$y_{in} = 0 \times 1 + 0 \times 1 = 0$
$(0,1)$	$y_{in} = 1$
$(1,0)$	$y_{in} = 1$
$(1,1)$	$y_{in} = 2$

Not possible

Case 2 : $\omega_1 = -1 \quad \omega_2 = 1$

$(0,0)$	$y_{in} = 0$
$(0,1)$	$y_{in} = 1$
$(1,0)$	$y_{in} = -1$
$(1,1)$	$y_{in} = 0$.

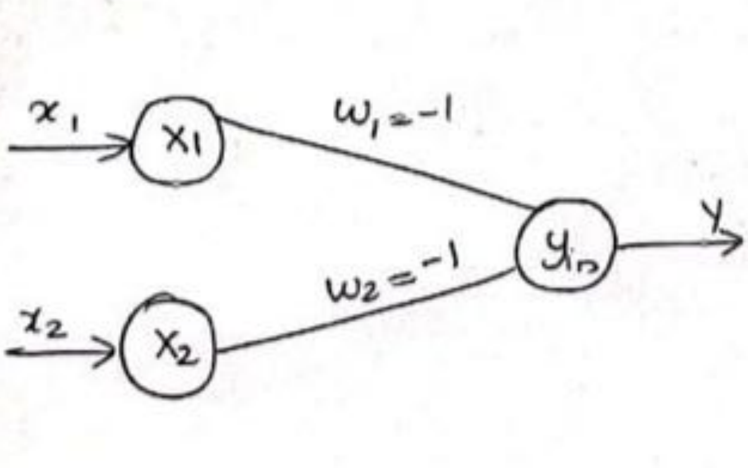
Case 3 : $\omega_1 = \omega_2 = -1$

$(0,0)$	$y_{in} = 0$
$(0,1)$	$y_{in} = -1$
$(1,0)$	$y_{in} = -1$
$(1,1)$	$y_{in} = -2$

$\therefore \theta = -1$

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 1 \\ 0 & \text{if } y_{in} < -1 \end{cases}$$

$$\therefore \omega_1 = \omega_2 = -1 \qquad \theta = -1$$



Q. Using Genetic algorithm with Roulette wheel selection method maximize the function $f(x)=x^2$ over $\{0,1,2,\cdots,31\}$ with initial x values of $(13, 24, 8, 19)$. Show one crossover and mutation

ans. <u>Step 1</u>: Coding of decision variables as some finite length string

X as binary unsigned integer of length 5

$[0, 31] = [00000, 11111]$

<u>Step 2</u>: Constant settings

$P_{mutation} = 0.0333$

$P_{cross} = 0.6$

Population size = 30.

<u>Step 3</u>: Select initial population at random

String number	Initial population	x value	f(x)	pselect $f_i/\Sigma f$	expected count $\frac{f_i}{\bar{f}}$	actual Count (roulette)
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1

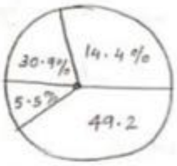
Sum 1170

Ave : 293

Max : 576

<u>Step 4</u>: Reproduction :

select mating pool by spinning roulette wheel 4 times



	Picked
01101	0.14
11000	0.49
01000	0.06
10011	0.31

Weighted Roulette wheel

→ The best get more copies
The average stay even
The worst die off.

<u>Step 5</u>: Crossover

strings are mated randomly using coin tosses to pair the couples.

- mated string couples crossover using coin tosses to select the crossing site.

String number	Mating pool after reproduction	Mate (randomly selected)	Crossover site (random)	New population	x-value	$f(x)=x^2$
1	0110|1	2	4	01100	12	144
2	1100|0	1	4	11001	25	625
3	11|000	4	2	11011	27	729
4	10|011	3	2	10000	16	256

Sum : 1754

Ave : 439

Max : 729

Neuro fuzzy hybrid system

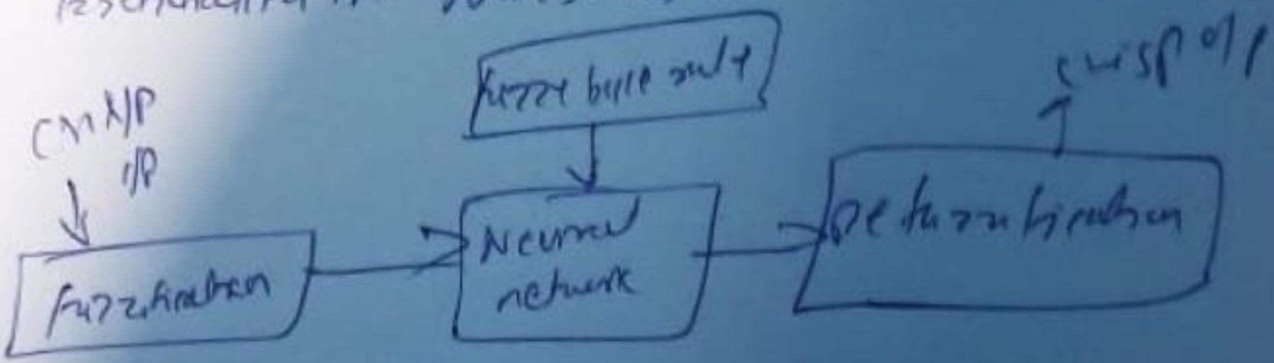
It is based on the learning of working of neural network theory. The learning process operates only on local information & causes local changes in the underlying fuzzy system. A neuro fuzzy system can be seen as a layer feed forward neural network.

The first layer - i/p variables

middle layer - fuzzy rules

third layer - o/p variables

Fuzzy sets are encoded as connection weights within the layers of the network which provides functionality in processing and training the model



working flow

each neuron transmits external crisp signals directly to the next layer.

Each fuzzification neuron receives a crisp i/p and determines the degrees to which the i/p belongs to the i/p fuzzy set

The fuzzy rule layer receives neuron that represent fuzzy sets

Advantages

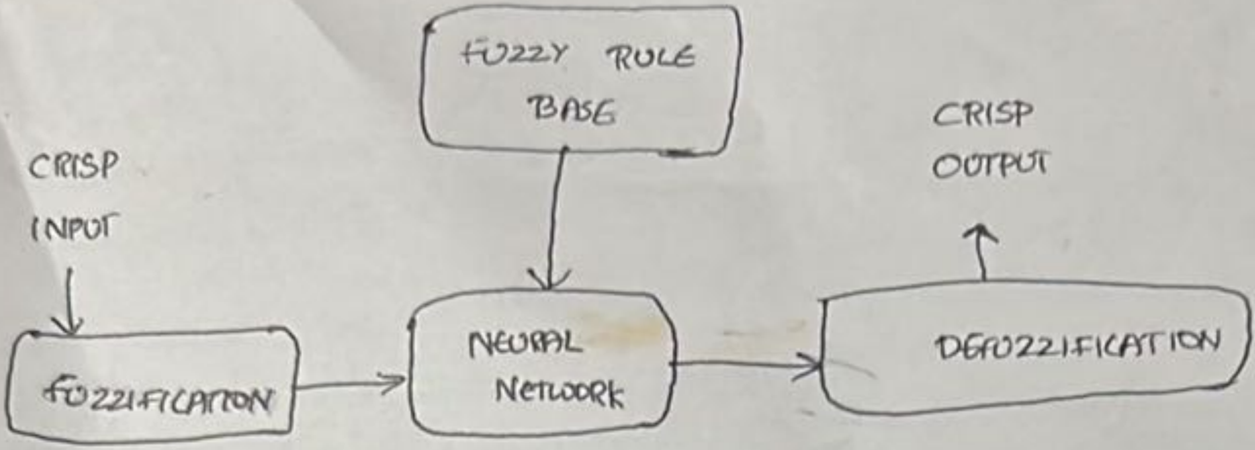
- It can handle numeric, linguistic logic kind of information
- It can manage imprecise partial vague information
- It has self learning, self organizing
- It can resolve conflicts

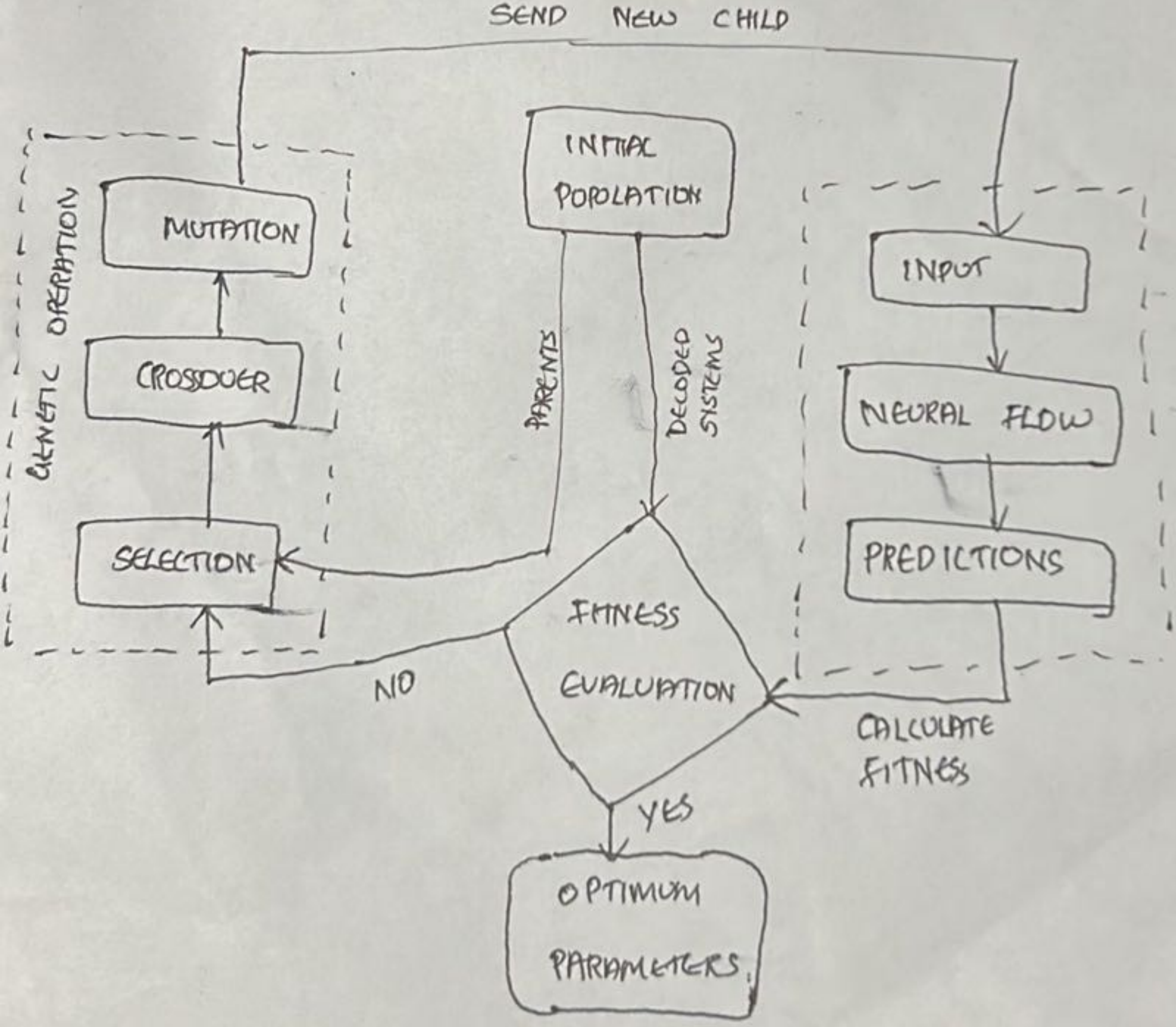
Disadvantages

- Problem of finding members values
- Hard to develop a model from a fuzzy system
- Neural networks cannot be used if training data is not available

Application:

- Student modelling
- medical systems
- Traffic control systems
- Forecating and predictions


FUZZY RULE BASE
CRISP INPUT
FUZZIFICATION
NEURAL NETWORK
DEFUZZIFICATION
CRISP OUTPUT


SEND NEW CHILD
GENETIC OPERATION
MUTATION
CROSSOVER
SELECTION
INITIAL POPULATION
PARENTS
DECODED SYSTEMS
INPUT
NEURAL FLOW
PREDICTIONS
FITNESS EVALUATION
NO
YES
CALCULATE FITNESS
OPTIMUM PARAMETERS