

Binary \rightarrow A string of binary digits

Genotype space = $\{0, 1\}$

X \rightarrow

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

 \rightarrow phenotype that can be evaluated
0 1 2 3 4 5 6 7 \leftarrow locus

Mutation \rightarrow flip each gene with a mutation rate

\hookrightarrow mutation rate \rightarrow probability pm.

parent

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

child

0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

Recombination

1-point crossover

p1

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

p2

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

c1

1	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---

c2

0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---

n-point crossover

p1

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

p2

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

c1

0	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

c2

1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

Uniform crossover \rightarrow flip a coin for each gene of the first child

\hookrightarrow then: make an inverse copy of the gene for the second child.

P1

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

P2

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

	P2	P1	P1	P2	P1	P2	P2	P1
C1	0	1	1	0	1	0	0	1

C2

1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

 \rightarrow inverse copy of C1

Integer \rightarrow a list of integers or categorical values (ex. colors) that can be converted to integers

Mutation \rightarrow creep and random resetting

Creep \rightarrow add a small positive or negative value to a gene with probability p .

5	1	7	8	3	10
---	---	---	---	---	----

 $\xrightarrow{+1}$

6	1	7	9	4	10
---	---	---	---	---	----

Random resetting \rightarrow choose a new random value with probability p .

$\{ \underset{1}{\text{blue}}, \underset{2}{\text{purple}}, \underset{3}{\text{red}}, \underset{4}{\text{yellow}} \} \rightarrow$ set of categories



Real-values and floats \rightarrow vector with one real value or float for each feature.

Mutation \rightarrow uniform and nonuniform

Uniform mutation \rightarrow choose a random value from the set of values for each value

$\{ \underset{1}{0.1}, \underset{2}{0.7}, \underset{3}{1.3}, \underset{4}{0.9} \} \rightarrow$ set of values



Non-uniform mutation \rightarrow add a random deviate to each variable separately, taken from $N(0, \sigma)$ Gaussian distribution and then curtail to range

\hookrightarrow **mutation step size** (standard deviation)

\hookrightarrow controls amount of change.

2/3 of drawings will lie in range $(-\sigma \text{ to } +\sigma)$ \rightarrow close to the original value.

0.7	1.3
-----	-----

 \rightarrow

0.8	1.1
-----	-----

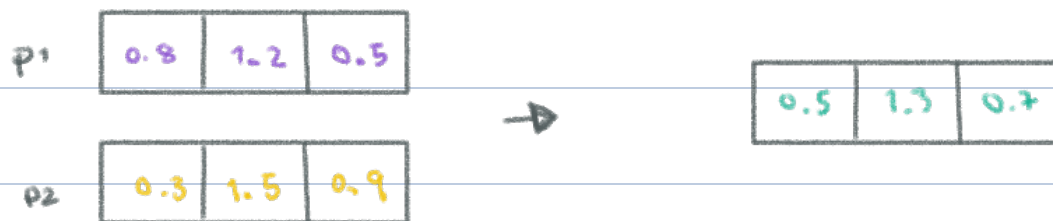
Recombination \rightarrow Discrete, arithmetic or both $\&$

Discrete \rightarrow each allele in offspring 2 comes from one of its parents with equal probability OR we could use uniform or n-point

p_1	0.8	1.2	0.5
p_2	0.3	1.5	0.9
\rightarrow			
c_1	0.8	1.5	0.9
c_2	0.3	1.2	0.5

Arithmetic \rightarrow performs an arithmetic operation on the parents to create one offspring.

\hookrightarrow can use a parameter α \rightarrow constant, variable (e.g. depend on the age of the population) or picked at random every time.



Permutations \rightarrow a list where each element occurs exactly once in a specific order.

Mutation \rightarrow must change at least two values.

\hookrightarrow swap, insert, scramble and inversion

Swap \rightarrow pick two alleles at random and swap their position



Insert → pick two allele values at random, and move the second to follow the first, shifting the rest to accommodate.



Scramble → pick a subset of genes at random, and randomly rearrange the alleles in those positions



Inversion → pick two positions at random and invert the substring between them.



Crossover \rightarrow preserving adjacency

\hookrightarrow partially mapped crossover and edge recombination

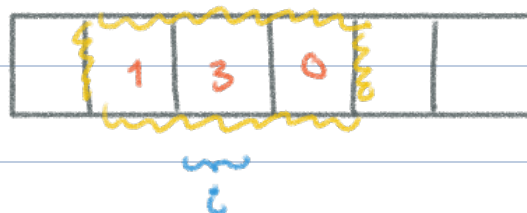
Partially mapped crossover \rightarrow create two children starting from a segment from p_1 , then starting from p_2 .



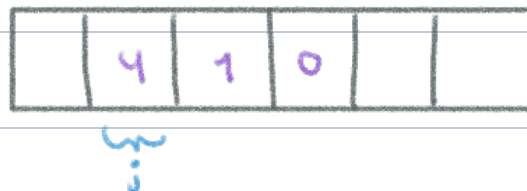
① choose a random segment from p_1



② look for i elements that didn't get copied from p_2



③ see what element j has been copied in its place



④ place i into position occupied by j in p_2

3	4	1	0		
---	---	---	---	--	--

* if the position is already occupied by some k , put i in the position occupied by k in $P2$

* Happens if element i is in the segment area in $P2$

⑤ fill the rest with the elements from $p2$

3	4	1	0	7	2
---	---	---	---	---	---

Edge recombination

1	3	9	5	0
---	---	---	---	---

3	9	1	0	5
---	---	---	---	---

construct an edge table with common edges:

● → common

0	1 5
1	0 3 9
3	9 1 5
5	0 9 3
9	3 5 1

① pick a random element x and put it in

1				
---	--	--	--	--

REPEAT: ♡

② remove x from edge list

● \rightarrow common

0	1 5
1	0 3 9
3	9 1 5
5	0 9 3
9	3 5 1

● \rightarrow common

0	5
1	0 3 9
3	9 5
5	0 9 3
9	3 5

③ if x has common edge, pick that:

1	0			
---	---	--	--	--

else:

pick the element from x 's edges with the shortest list, ties are split at random

④ if you reach an empty list, pick a new element at random.

Crossover \rightarrow Preserving order

\hookrightarrow order crossover and cycle crossover

Order crossover \rightarrow copy randomly selected set from p_1 , the rest (if not present) from p_2 .

1	3	9	5	0
---	---	---	---	---

3	9	1	0	5
---	---	---	---	---

① copy from p_1 and put in same pos

	3	9	5	
--	---	---	---	--

② iterate over p_2 from right cut point

3	9	1	0	5
---	---	---	---	---

2 3 4 5 1
 $\swarrow \quad \swarrow$

③ copy from p_2 if not present and in order

0	3	9	5	1
---	---	---	---	---

Cycle crossover \rightarrow make a cycle of alleles and put in offspring, each allele comes thus from one parent together with its position

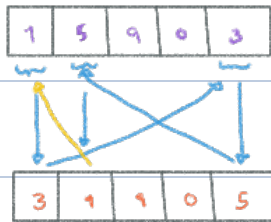
① start with the first allele of p1 or p2

1	3	9	5	0
---	---	---	---	---

~~~~~

|   |   |   |   |   |
|---|---|---|---|---|
| 3 | 9 | 1 | 0 | 5 |
|---|---|---|---|---|

② place alternate cycles from p1 and p2



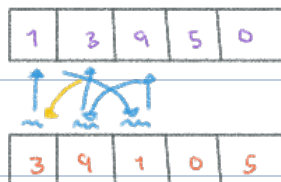
|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 5 | 9 | 0 | 3 |
|---|---|---|---|---|

~~~~~   ~~~~~   ~~~~~



| | | | | |
|---|---|---|---|---|
| 1 | 5 | 9 | 0 | 3 |
|---|---|---|---|---|

~~~~~   ~~~~~



|   |   |   |   |   |
|---|---|---|---|---|
| 3 | 9 | 1 | 5 | 0 |
|---|---|---|---|---|

~~~~~   ~~~~~   ~~~~~   ~~~~~   ~~~~~



Tree \rightarrow arithmetic or logical formulas,
programs and graphs

Mutation \rightarrow replace randomly chosen subtree
by randomly generated tree from set of
possible trees/nodes

Recombination \rightarrow make two children from
two parent trees by choosing a random
point in each parent assigning left
subtree from p_1 + right subtree from p_2
to c_1 and the other way.