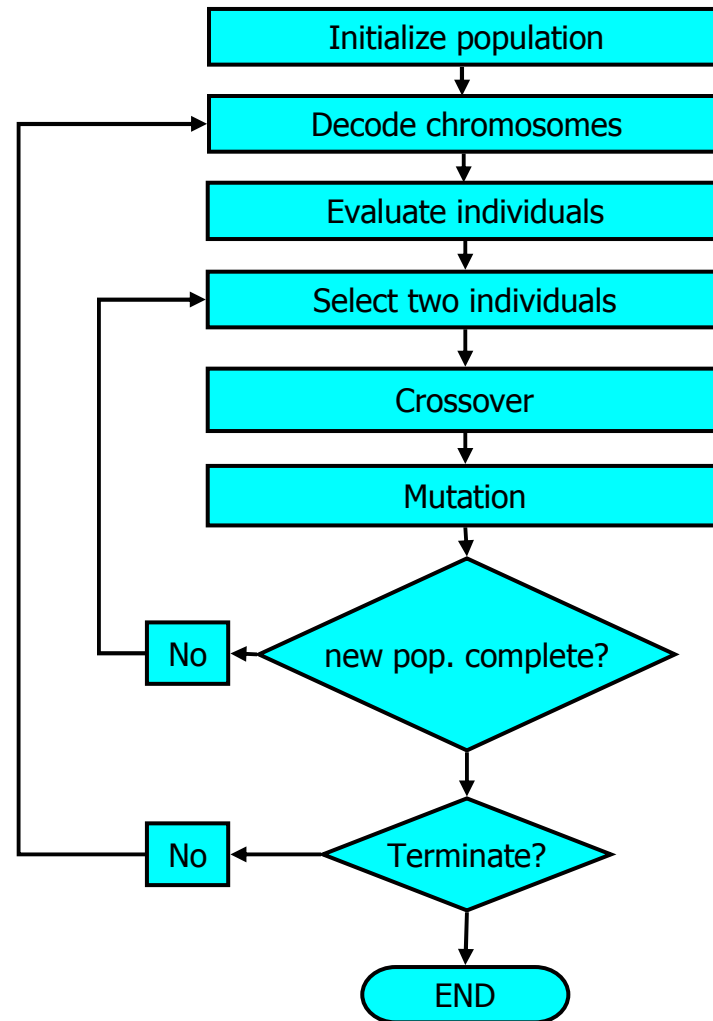
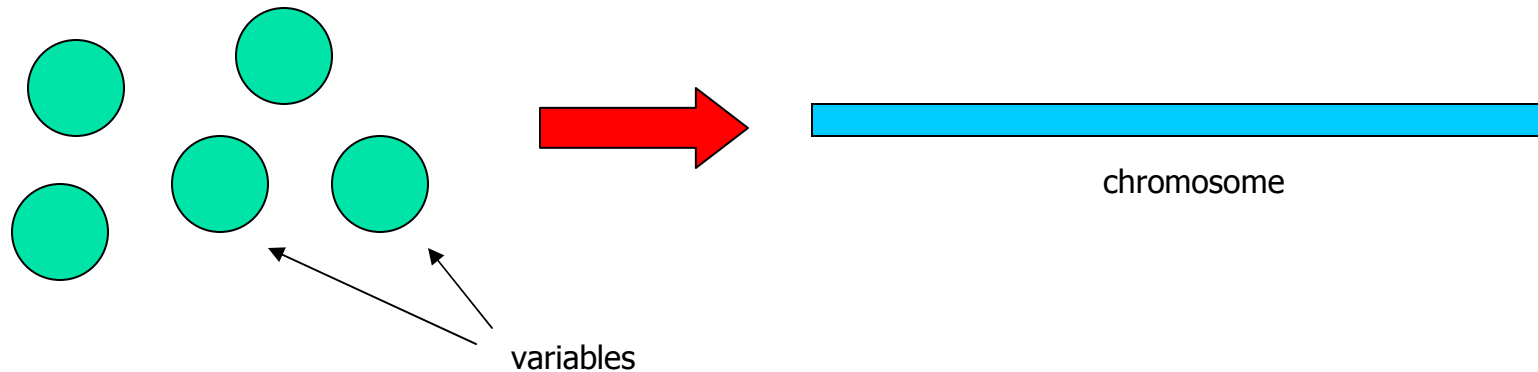




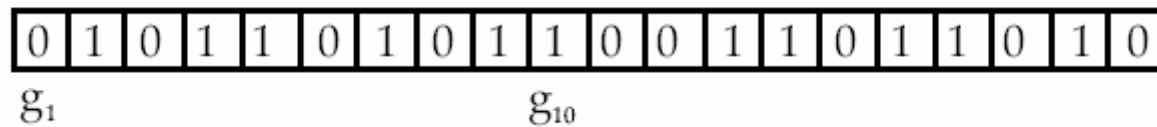
Encoding: the variables of the problem are encoded in strings of digits known as *chromosomes*.



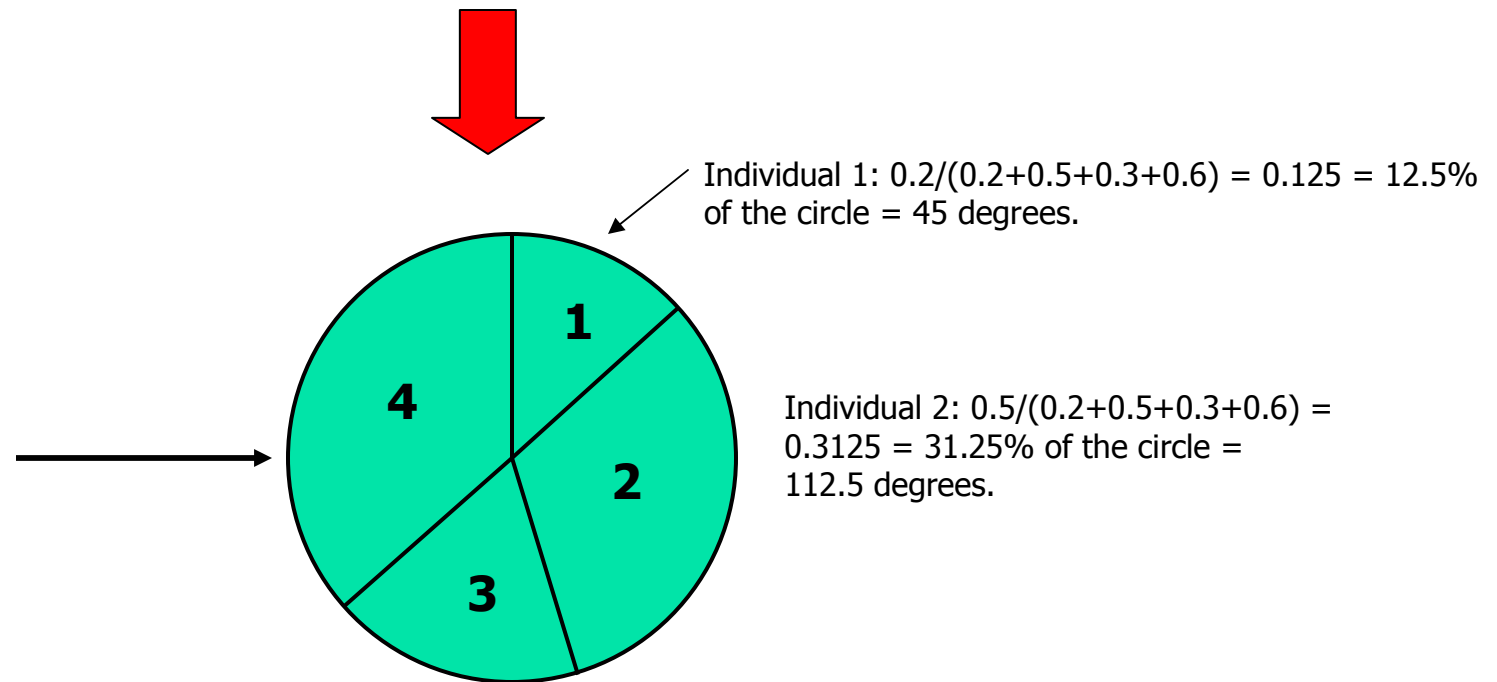
The exact encoding procedure varies from problem to problem – in the simple case of optimization of a function $f(x_1, x_2, x_3, \dots, x_n)$ the variables x_i , $i = 1, \dots, n$ should be encoded. In more complex problems (e.g. optimization of artificial brains for robots or optimization of neural networks), the specification of the encoding procedure can be much more complicated.

Random initialization of a population containing N chromosomes

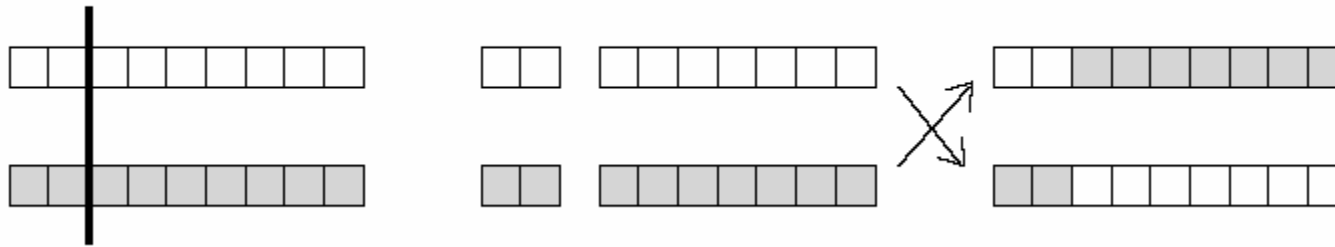
1	0 1 0 1 1 0 1 1 1 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0 1
2	1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0
3	0 1 0 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 1
	...
N	0 1 0 1 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1



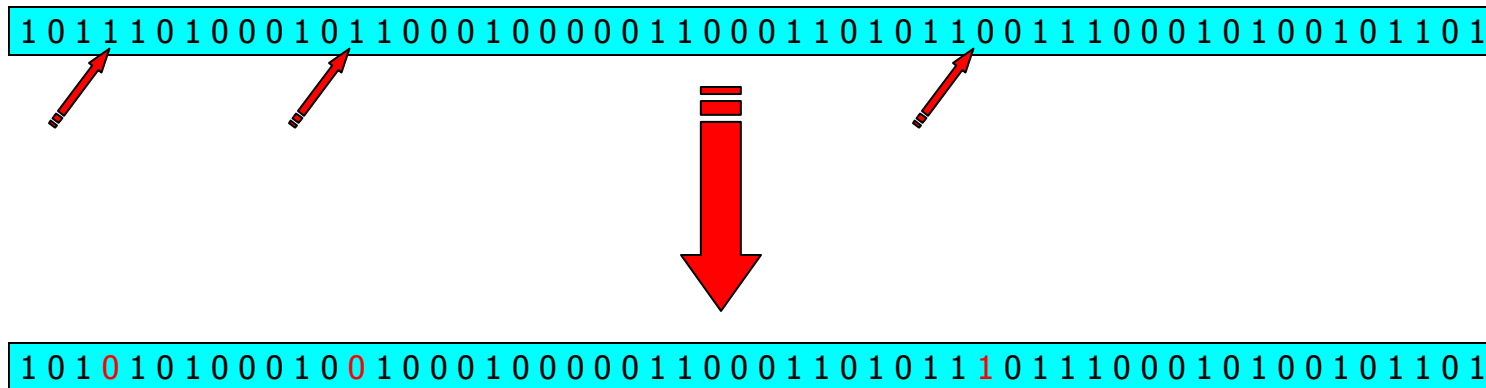
1 0 0 0 0 1 1 0 1 1 0 1 1 0	→	Individual 1	$f(1) = 0.2$	$\Sigma f(i) = 1.6$
0 1 1 0 0 0 0 1 1 1 1 0 0 1 0	→	Individual 2	$f(2) = 0.5$	
1 1 0 1 1 1 0 0 1 1 0 1 0 1 1 1	→	Individual 3	$f(3) = 0.3$	
0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0	→	Individual 4	$f(4) = 0.6$	



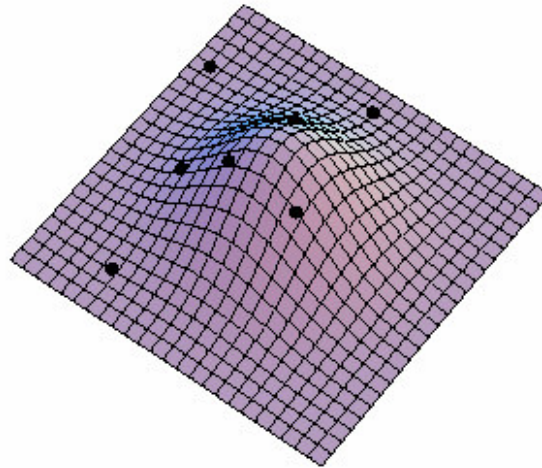
Crossover



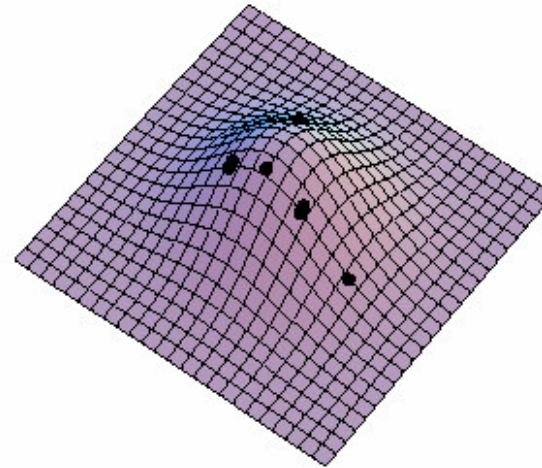
Mutation



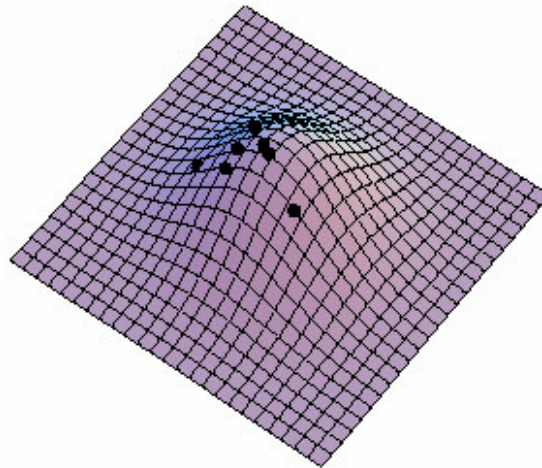
Generation 2



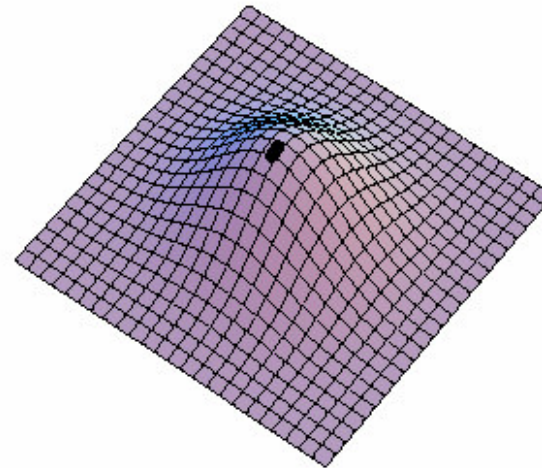
Generation 4



Generation 8



Generation 16



0.465148391133

0.953185830213

0.159105940423

5

6

1

5

0

4

7

3

5

9

9

8

1

0

0

1

0

1

1

1

1

0

0

1

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

x

y

0.34215678 0.34767614 0.75636461 0.77663421 0.91893711 0.01375721 0.67485732

x_1

x_2

x_3

x_4

x_5

x_6

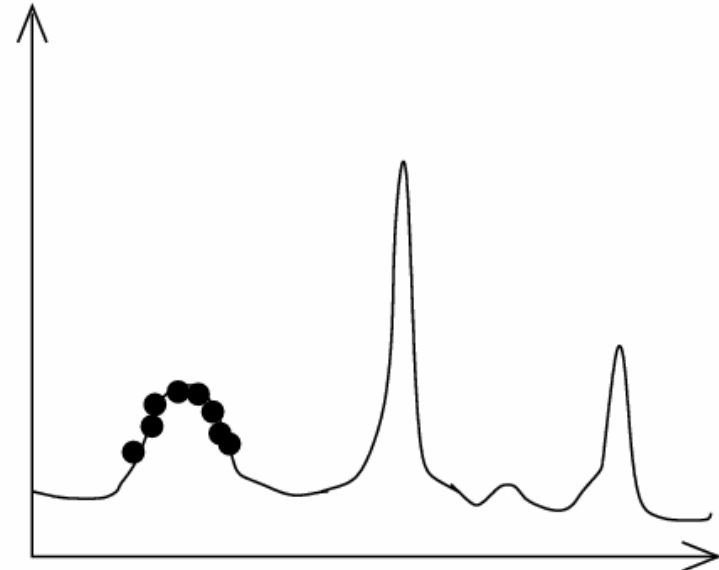
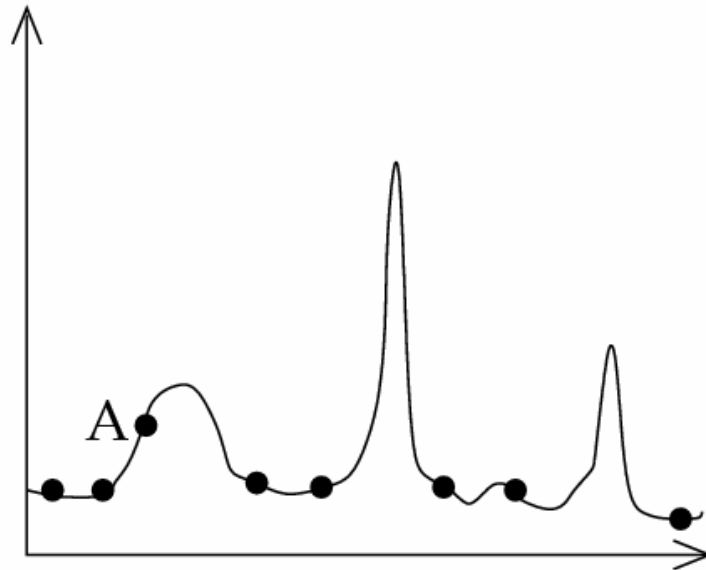
x_7

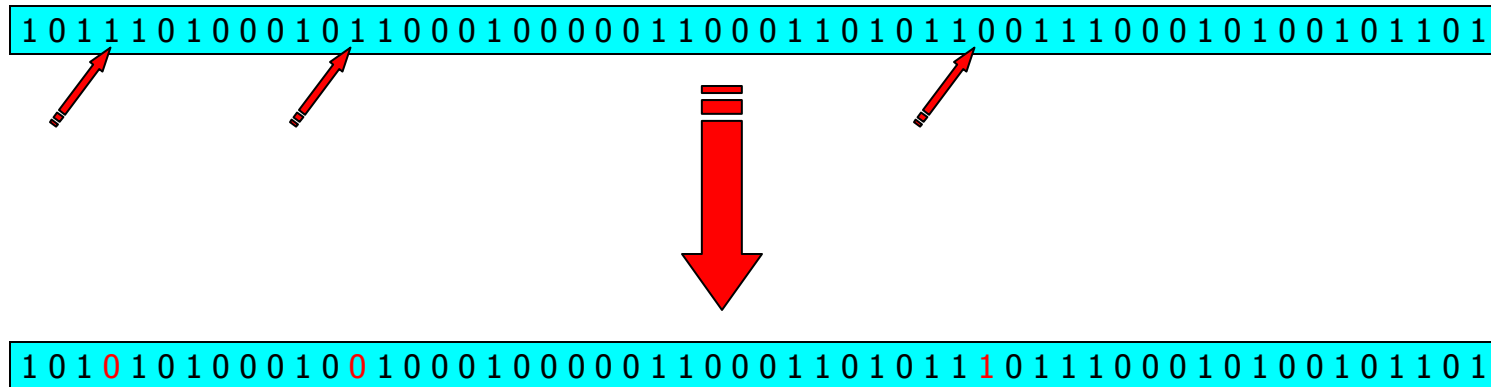
Fitness ranking:

$$f(i) = (N + 1 - R(i)).$$

$$f(i) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(i) - 1}{N - 1} \right).$$

Premature convergence:





Creep mutations: (example of implementation, in connection with real-number encoding)

0.23131541	0.59182321	0.98148176	0.01987612	0.59176192	0.19857612
------------	------------	------------	------------	------------	------------

For each gene, check whether or not the gene should be mutated (using p_{mut})

If the gene should be mutated, check whether or not creep mutations should be used (using p_{creep})

If yes, carry out a creep mutation:

$$g \rightarrow g' = g - c + 2 c r$$

If no, carry out an ordinary mutation:

$$g \rightarrow g' = \text{rand}$$