SEARCH IN COMPLEX ENVIRONMENTS II

الدكتور/ مصطفى شبل

Evolutionary algorithms()

Developed: USA in the 1970's

Early names: J. Holland, K. DeJong, D. Goldberg

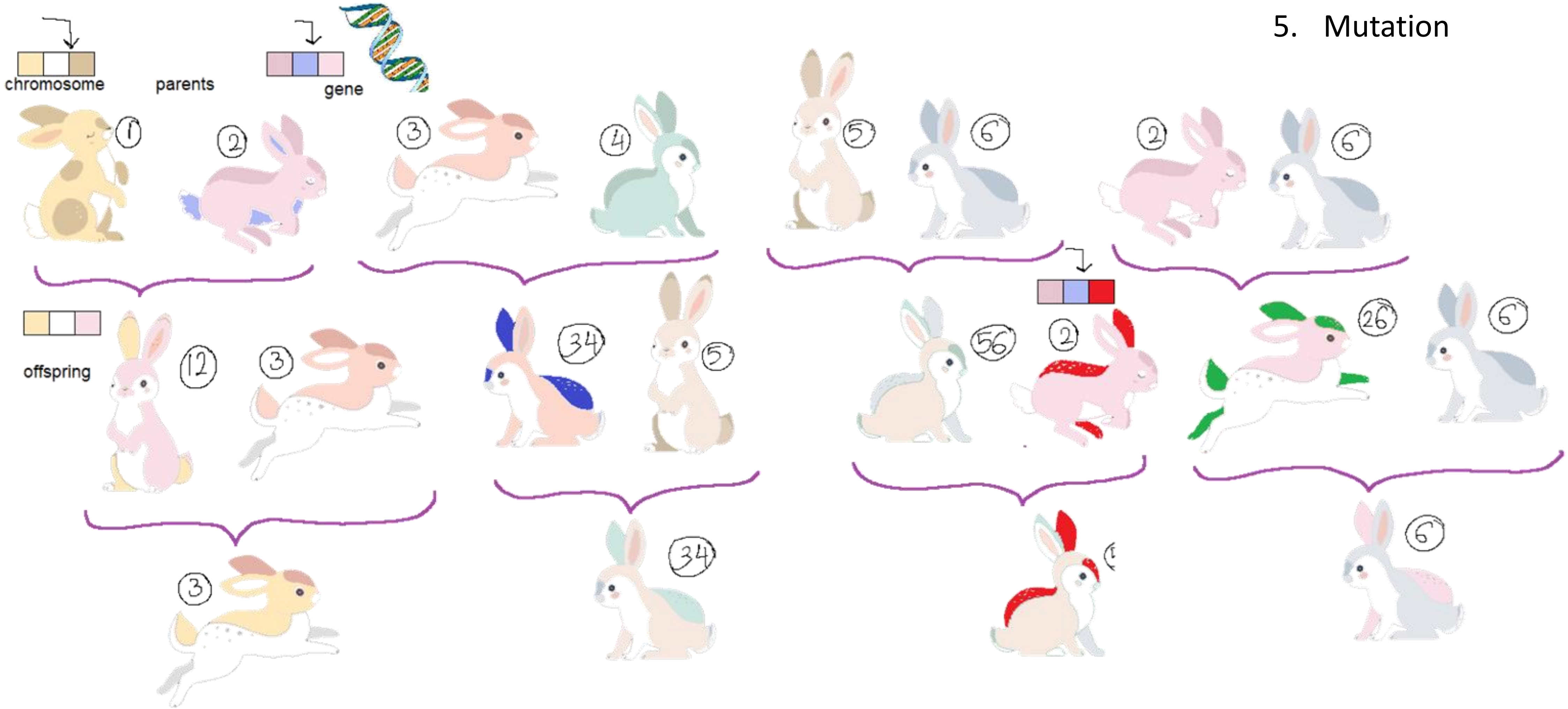
Genetic Algorithms

How is a population with increasing fitness generated?

- Let us consider a population of rabbits. Some rabbits are faster than others, and we may say that these rabbits possess superior fitness, because they have a greater chance of avoiding foxes, surviving and then breeding.
- If two parents have superior fitness, there is a good chance that a combination of their genes will produce an offspring with even higher fitness. Over time the entire population of rabbits becomes faster to meet their environmental challenges in the face of foxes.

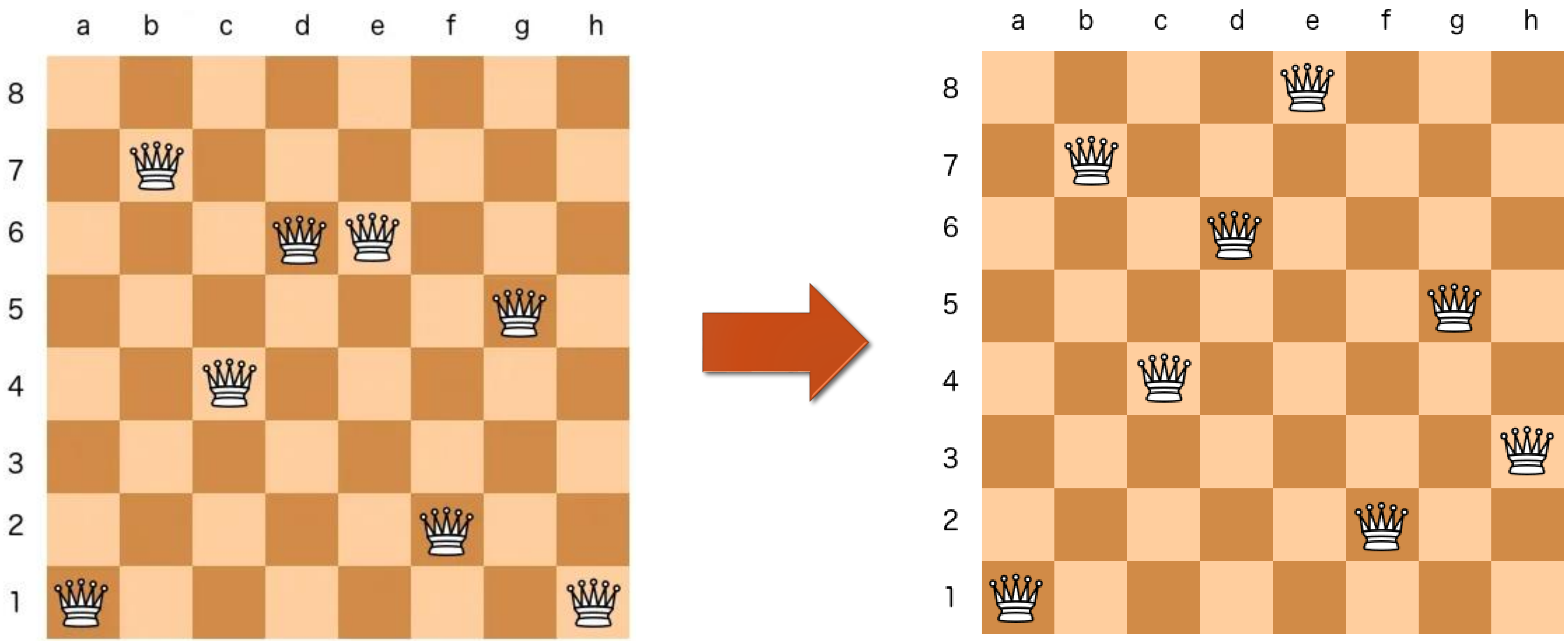
How is a population with increasing fitness generated?

1. Initial Population
2. Fitness Function
3. Selection
4. Crossover
5. Mutation



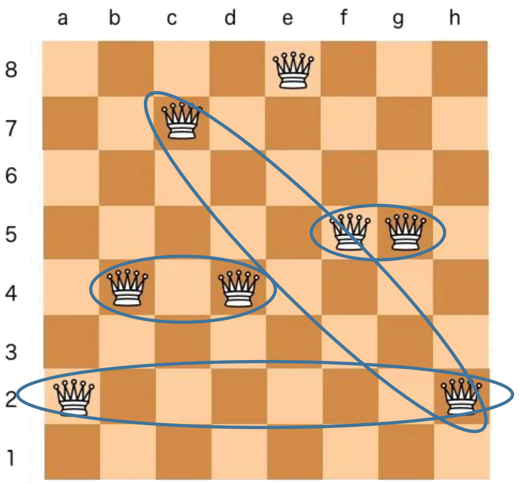
Recessive /dominant trait

Case Study : 8 Queen problem

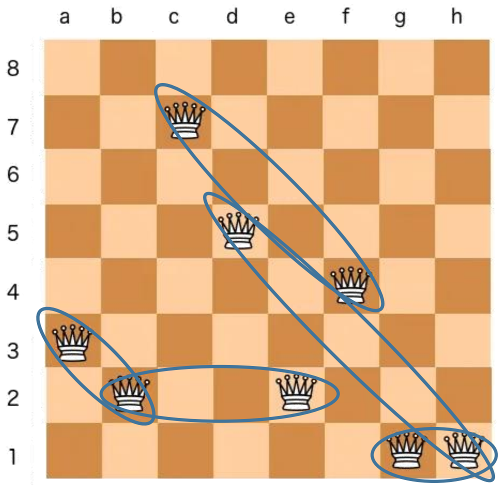


Representation (numeric)

24748552

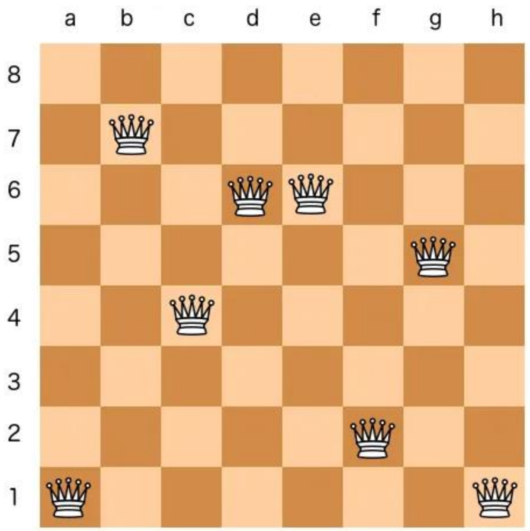


non attacked pair=24



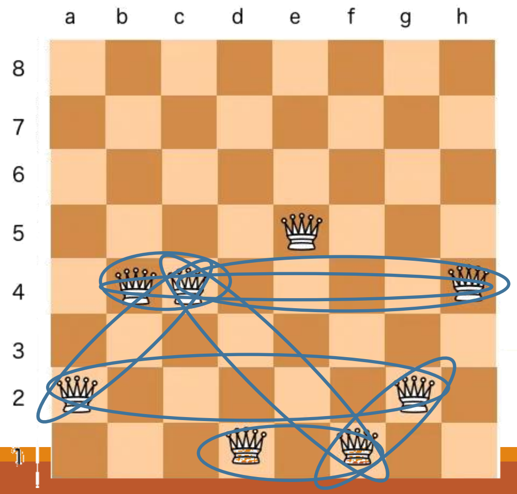
32752411

non attacked pair=23

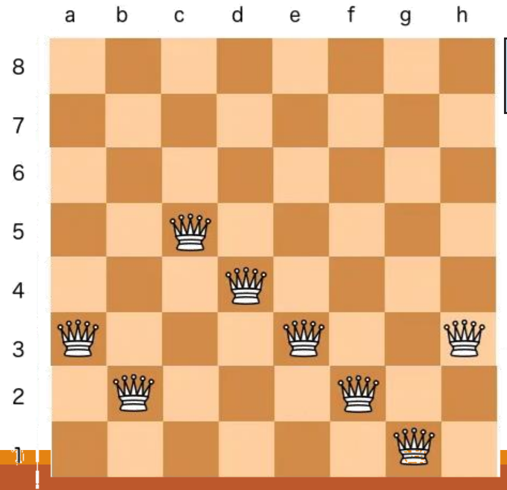


[1,7,4,6,6,2,5,1]

24415124



non attacked pair=20



32543213

non attacked pair=11

$$_nP_r = \frac{n!}{(n-r)!}$$

$_nP_r$ = permutation

n = total number of objects

r = number of objects selected

overall non-attacking queens =8x7=56.

For pair concept = 56/2=28

Selection (fitness function)and Crossover

[2, 4, 7, 4, 8, 5, 5, 2]	24
[3, 2, 7, 5, 2, 4, 1, 1]	23
[1, 5, 7, 2, 6, 1, 3, 5]	22
[6, 3, 6, 5, 2, 4, 6, 7]	21
[2, 1, 3, 3, 7, 0, 3, 3]	21
[1, 0, 1, 0, 7, 4, 2, 6]	21
[2, 4, 4, 1, 5, 1, 2, 4]	20
[0, 7, 3, 6, 0, 6, 2, 6]	20
[0, 3, 5, 1, 4, 3, 2, 2]	20
[0, 7, 2, 3, 0, 1, 2, 5]	19
[5, 1, 1, 3, 3, 4, 5, 3]	19
[6, 5, 0, 5, 3, 3, 0, 5]	18
[0, 2, 2, 3, 5, 1, 0, 1]	18
[4, 2, 7, 3, 4, 4, 7, 3]	18
[2, 3, 6, 3, 0, 5, 2, 5]	17

[2, 4, 7, 4, 8, 5, 5, 2]
[3, 2, 7, 5, 2, 4, 1, 1]
[1, 5, 7, 2, 6, 1, 3, 5]
[6, 3, 6, 5, 2, 4, 6, 7]
[2, 1, 3, 3, 7, 0, 3, 3]
[1, 0, 1, 0, 7, 4, 2, 6]
[2, 4, 4, 1, 5, 1, 2, 4]
[0, 7, 3, 6, 0, 6, 2, 6]
[0, 3, 5, 1, 4, 3, 2, 2]

[2, 4, 7, 4, 2, 4, 1, 1]
[1, 5, 7, 2, 2, 4, 6, 7]
[2, 1, 3, 3, 7, 4, 2, 6]

Mutation , Elitism & Culling

[2, 4, 7, 4, 2, 4, 1, 1]

[1, 5, 7, 2, 2, 4, 6, 7]

[2, 1, 3, 3, 7, 4, 2, 6]

[2, 4, 7, 4, 2, **5**, 1, 1]

[1, 5, 7, 2, 2, 4, 6, 7]

[2, 1, **4**, 3, 7, 4, 2, 6]

[2, 4, 7, 4, 8, 5, 5, 2]	24
[3, 2, 7, 5, 2, 4, 1, 1]	23
[1, 5, 7, 2, 6, 1, 3, 5]	22

Elitism: guarantees that overall fitness will never decrease over time

Culling: individuals below a given threshold are discarded, can lead to a speedup

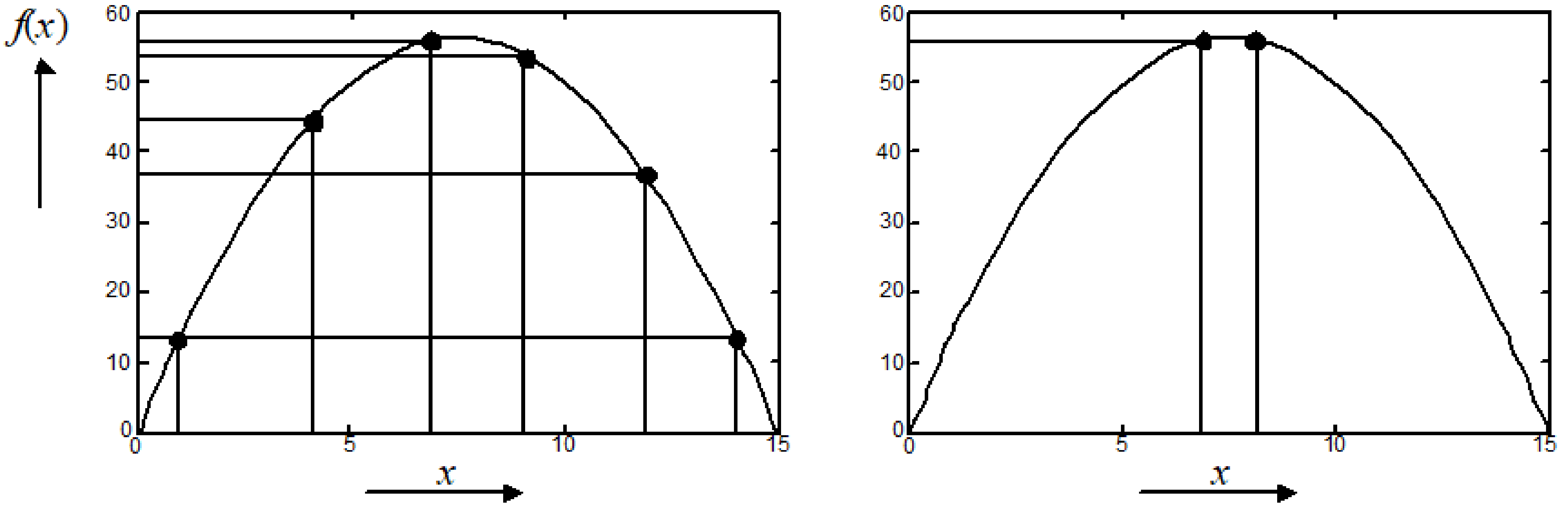
Genetic algorithm

```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights ← WEIGHTED-BY(population, fitness)
    population2 ← empty list
    for i = 1 to SIZE(population) do
      parent1, parent2 ← WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child ← REPRODUCE(parent1, parent2)
      if (small random probability) then child ← MUTATE(child)
      add child to population2
    population ← population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n ← LENGTH(parent1)
  c ← random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```

Genetic algorithms: case study

find the maximum value of the function $(15x - x^2)$ where parameter x varies between 0 and 15



(*a*) Chromosome initial locations.

(*b*) Chromosome final locations.

String Representation

chromosomes

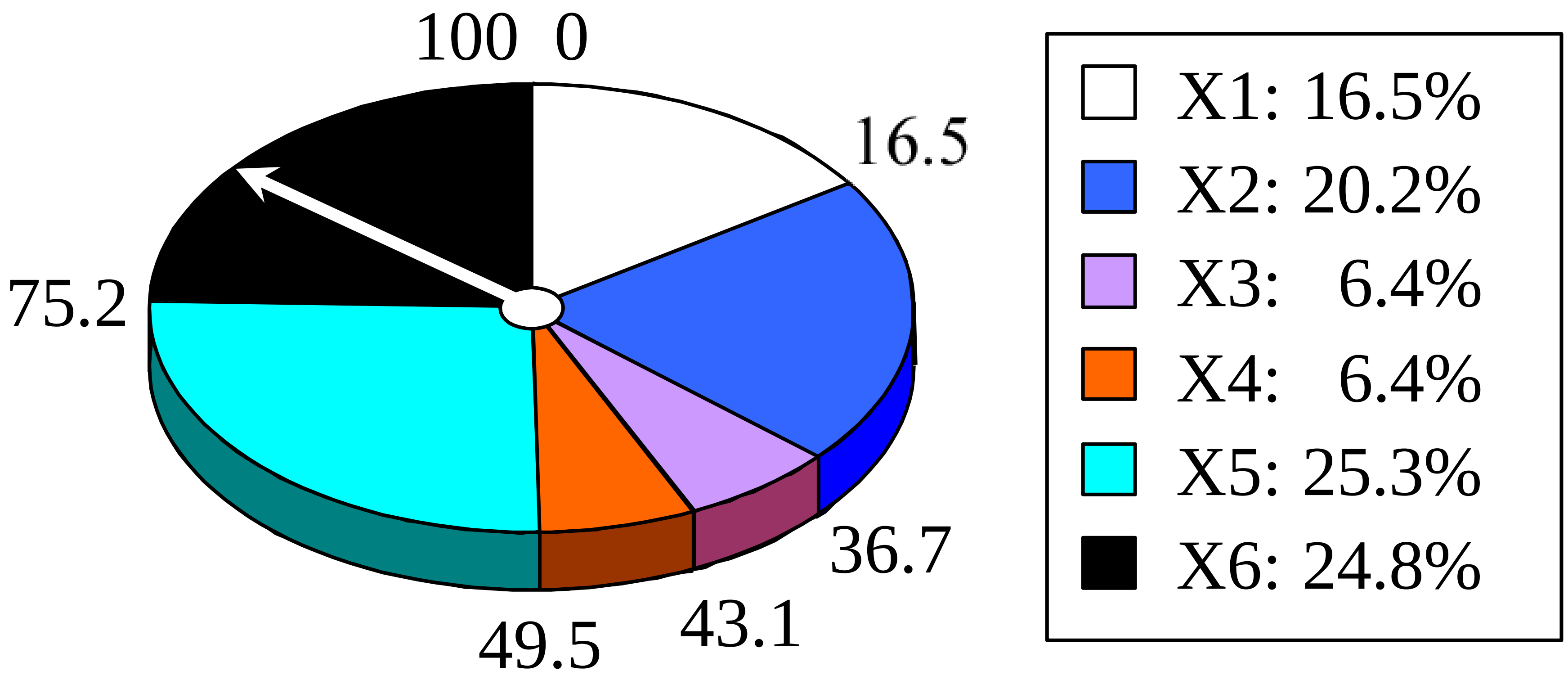
Integer	*Binary code*	*Integer*	*Binary code*	*Integer*	*Binary code*
1	0 0 0 1	6	0 1 1 0	11	1 0 1 1
2	0 0 1 0	7	0 1 1 1	12	1 1 0 0
3	0 0 1 1	8	1 0 0 0	13	1 1 0 1
4	0 1 0 0	9	1 0 0 1	14	1 1 1 0
5	0 1 0 1	10	1 0 1 0	15	1 1 1 1

Random Selection

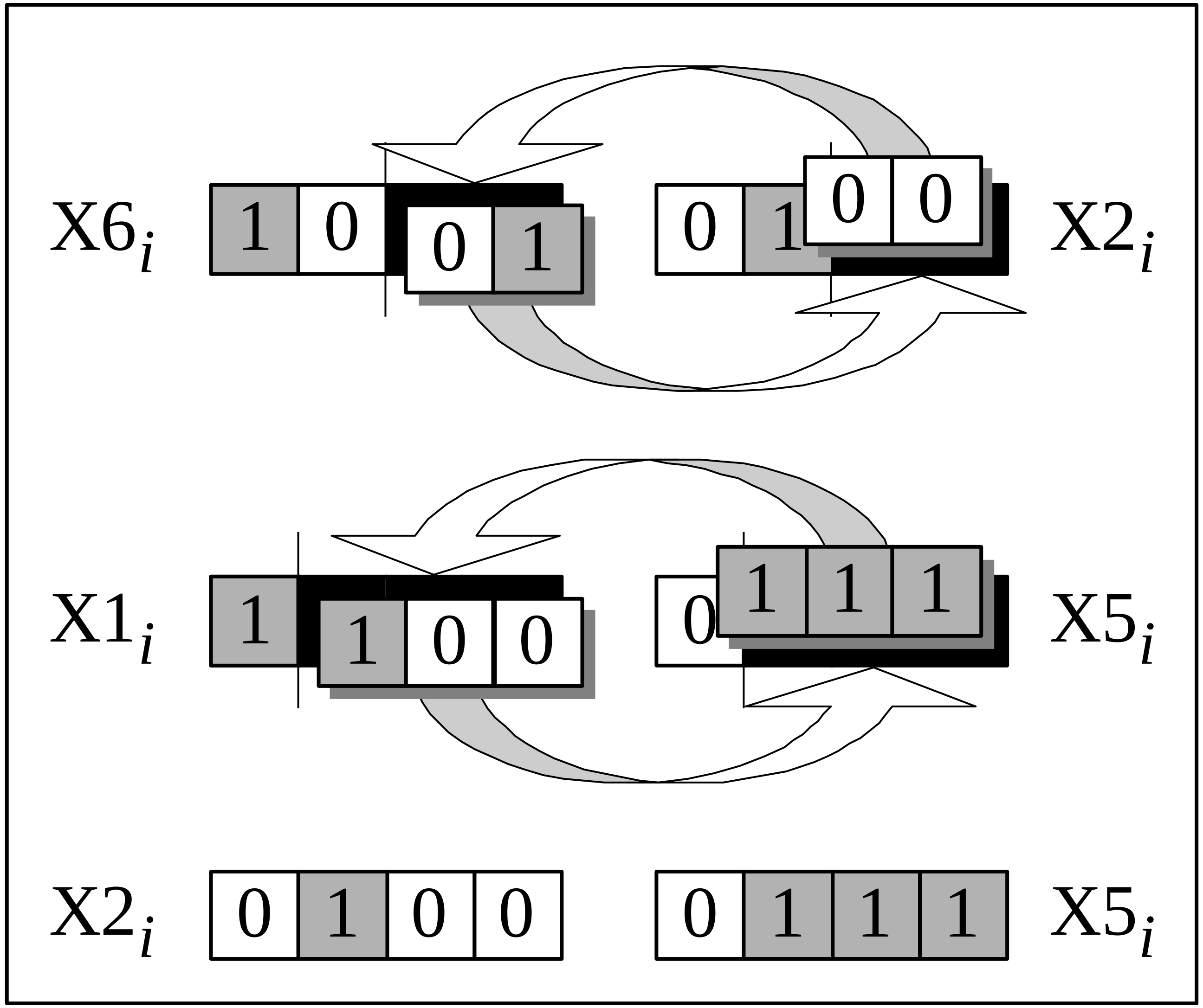
Chromosome label	*Chromosome string*	*Decoded integer*	*Chromosome fitness* f(x)	*Fitness ratio, %* $= \frac{f(x)}{\sum f(x)}$
X1	1 1 0 0	12	36	16.5
X2	0 1 0 0	4	44	20.2
X3	0 0 0 1	1	14	6.4
X4	1 1 1 0	14	14	6.4
X5	0 1 1 1	7	56	25.7
X6	1 0 0 1	9	54	24.8

Roulette wheel selection

The most commonly used chromosome selection techniques is the **roulette wheel selection**.



Crossover



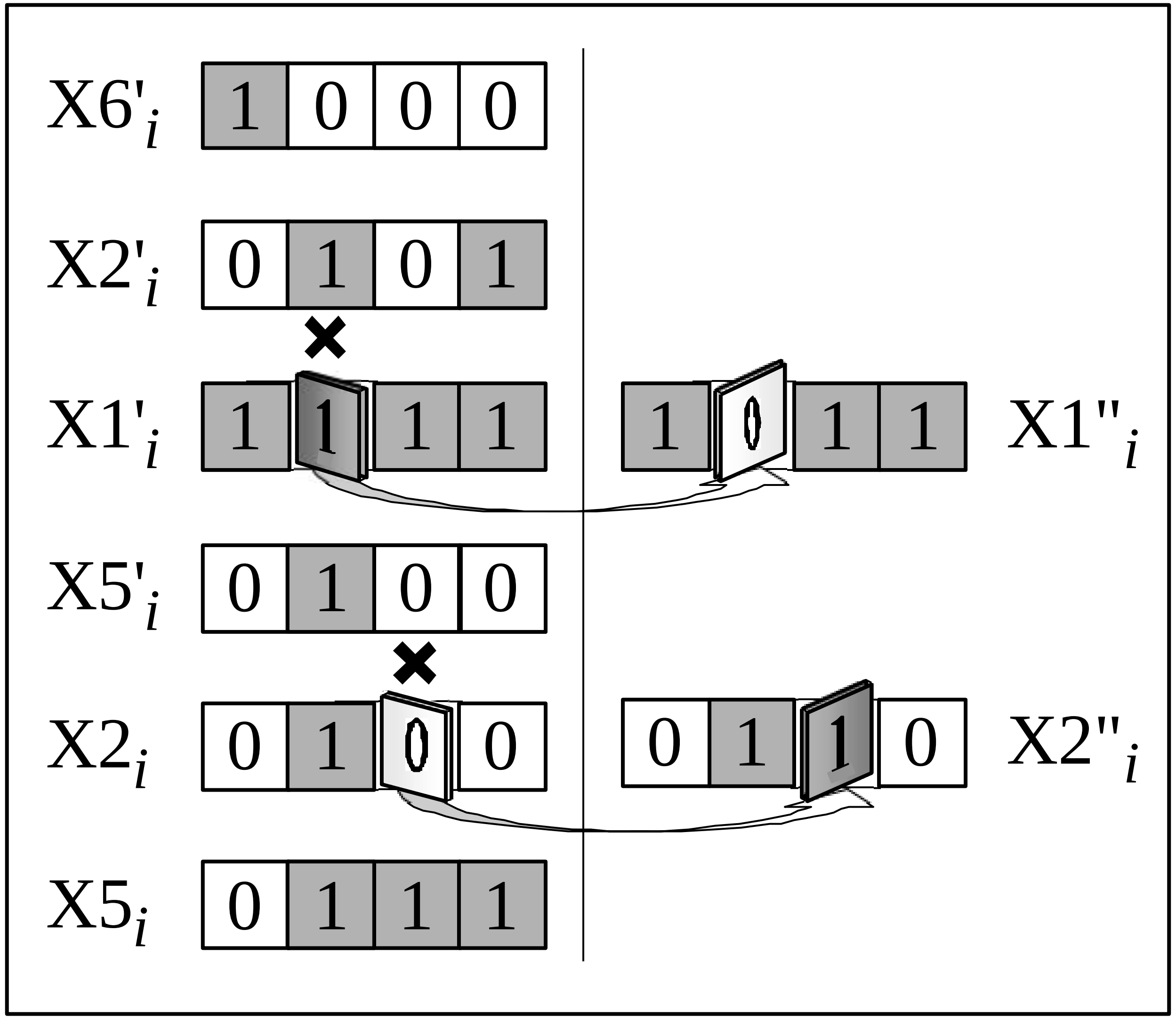
Splitting point randomly

Elitism

Mutation operator

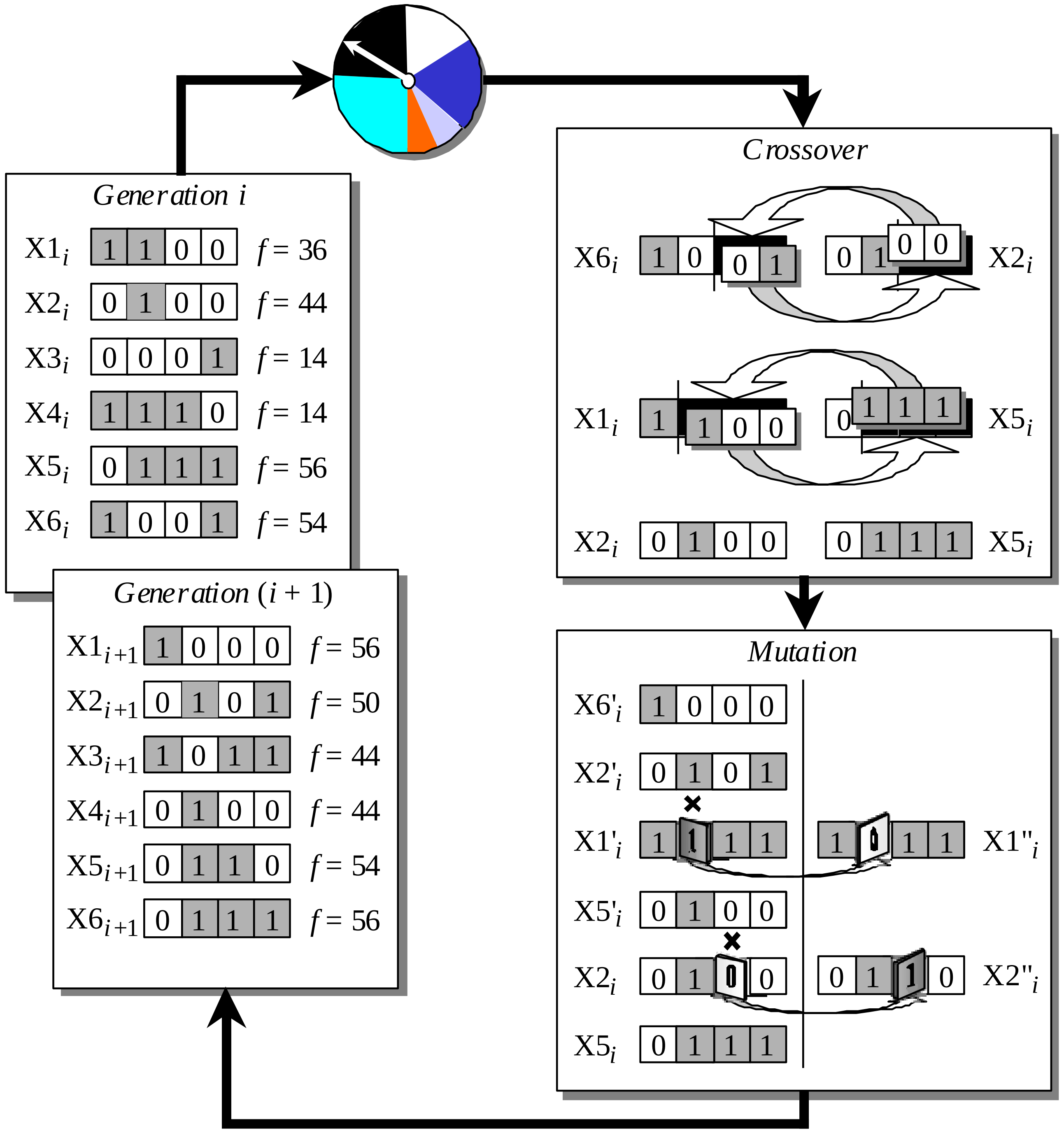
- Mutation represents a change in the gene.
- Mutation is a background operator. Its role is to provide a guarantee that the search algorithm is not trapped on a local optimum.
- The mutation operator flips a randomly selected gene in a chromosome.
- The mutation probability is quite small in nature, and is kept low for GAs, typically in the range between 0.001 and 0.01.

Mutation



X6'i 1 0 0 0
X2'i 0 1 0 1
X1'i 1 1 1 1
1 0 1 1 X1"i
X5'i 0 1 0 0
X2i 0 1 0 0
0 1 1 0 X2"i
X5i 0 1 1 1

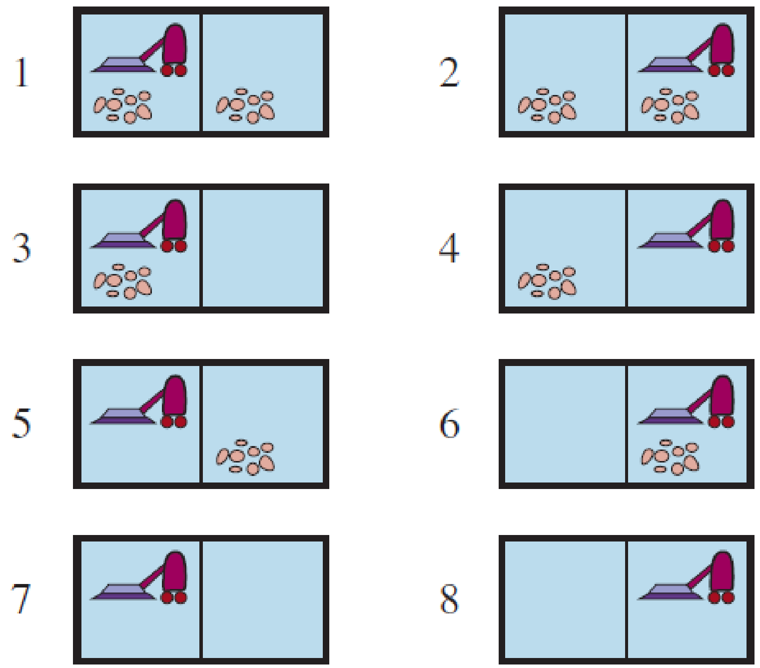
The genetic algorithm cycle

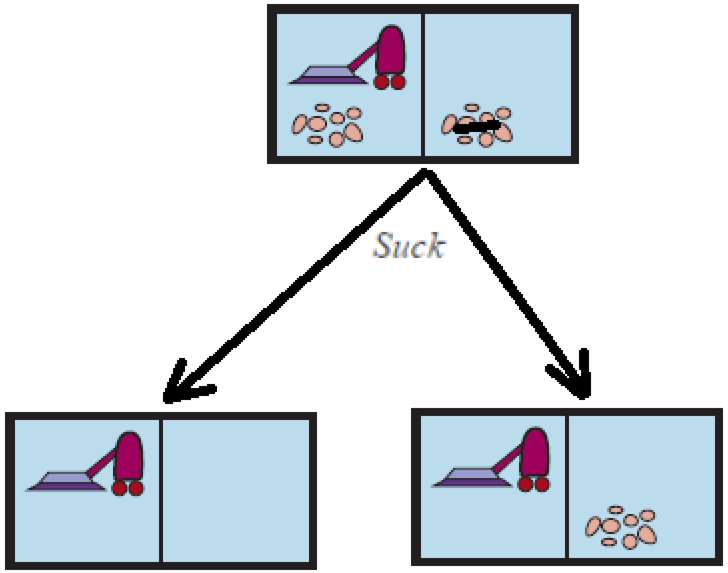


Search with Nondeterministic Actions

Definition

Environment is nondeterministic, the agent doesn't know what state it transitions to after taking an action



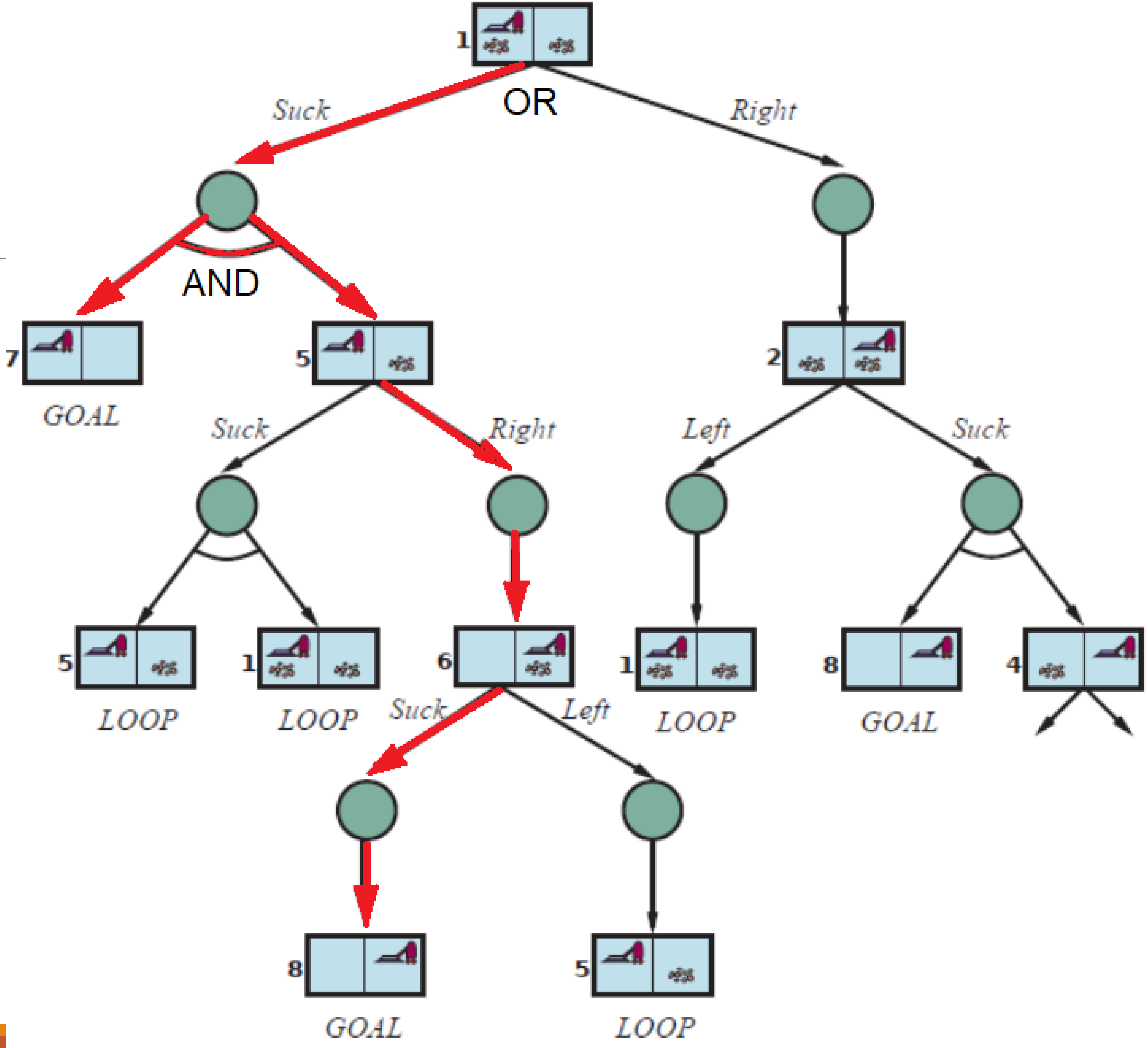


RESULTS(1; Suck) = {5;7}

AND–OR search trees

In a nondeterministic environment, branching is also introduced by the *environment's choice* of outcome for each action (AND node)

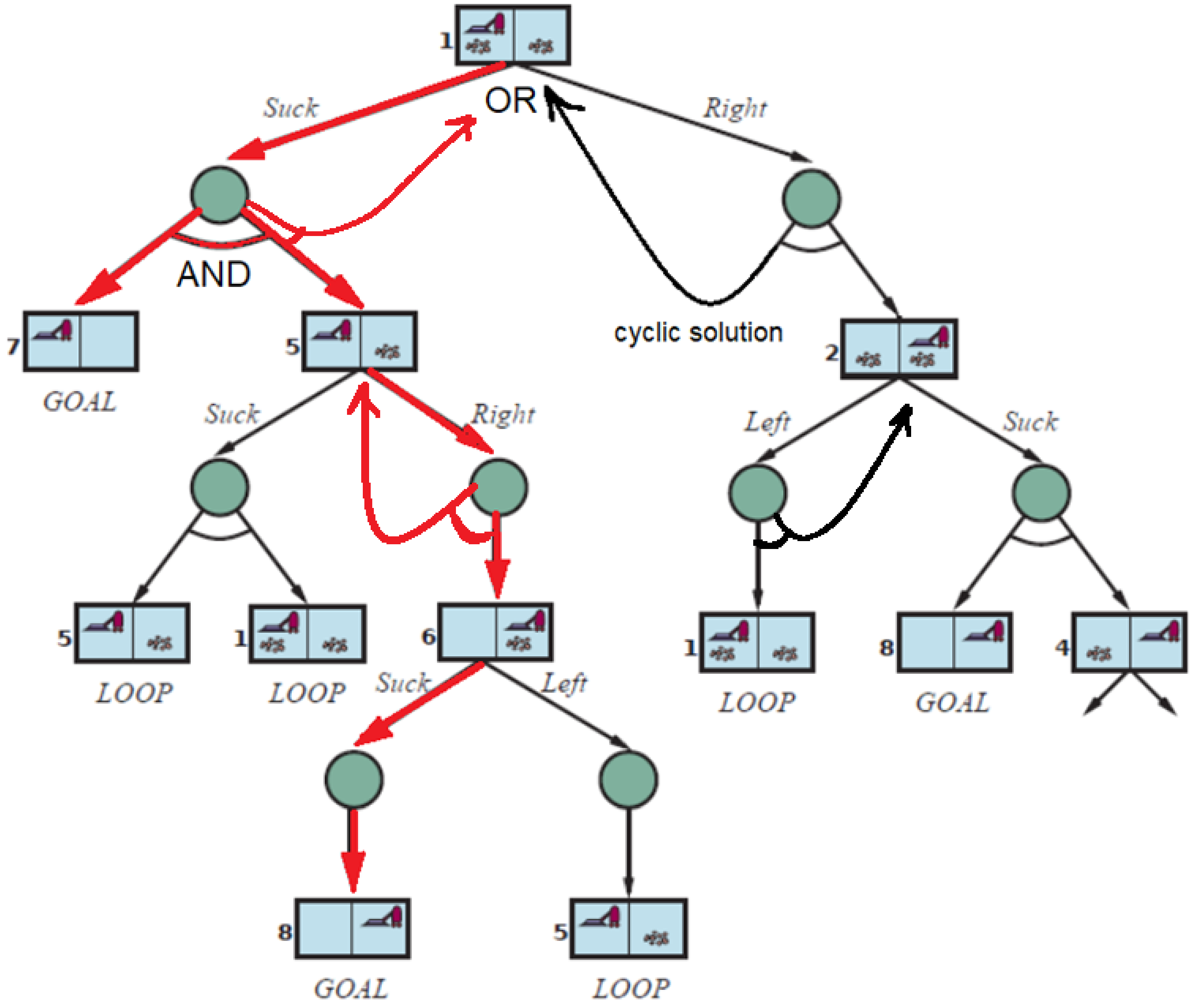
In a deterministic environment, the only branching is introduced by the *agent's own choices* in each state:(OR node)



Looping

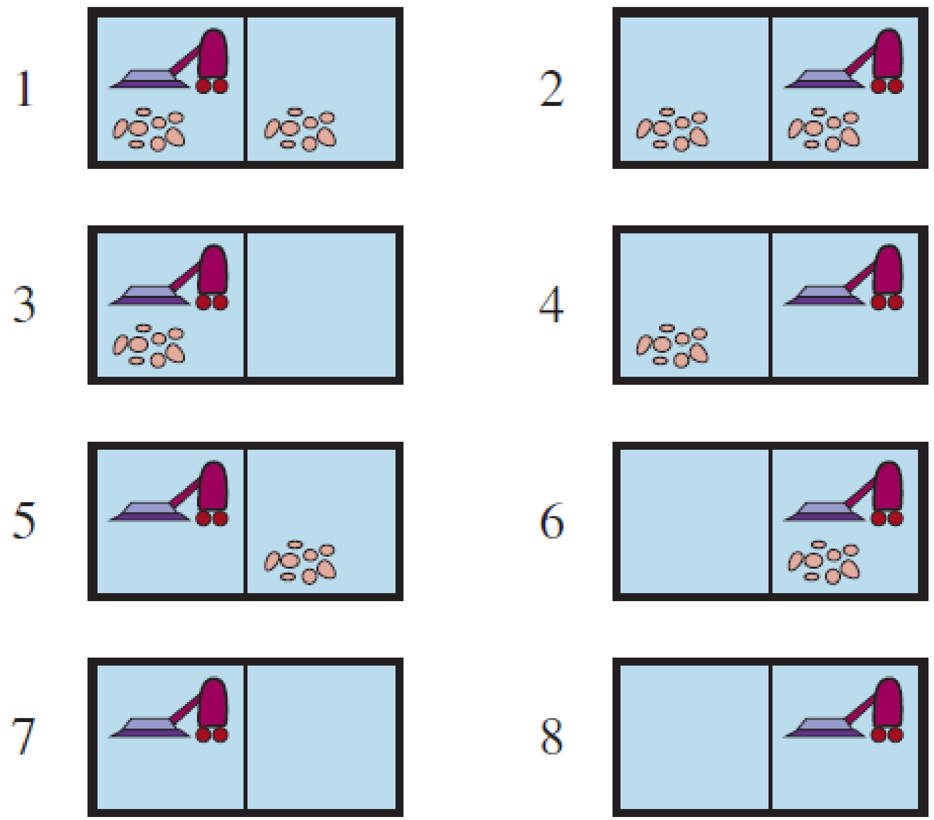
cyclic solution : to keep trying
Action until it works

[**while** State=1 **do** Right;Suck]



Search in Partially Observable Environments

known environment , unknown location



No observation at all.

Imagine the light is down in home !! What u have to do ?

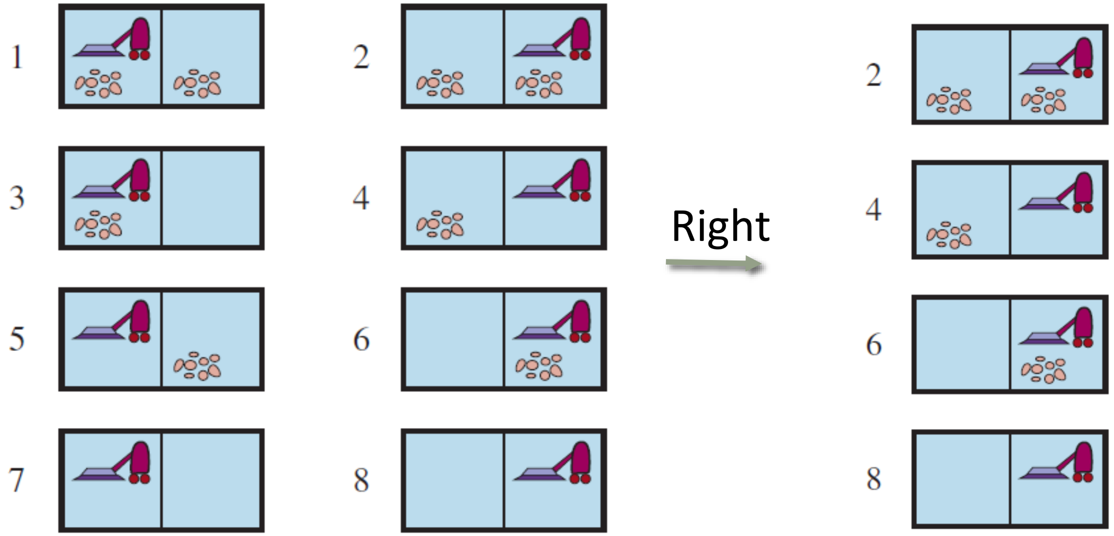
Initial state ?(may be known or not!)

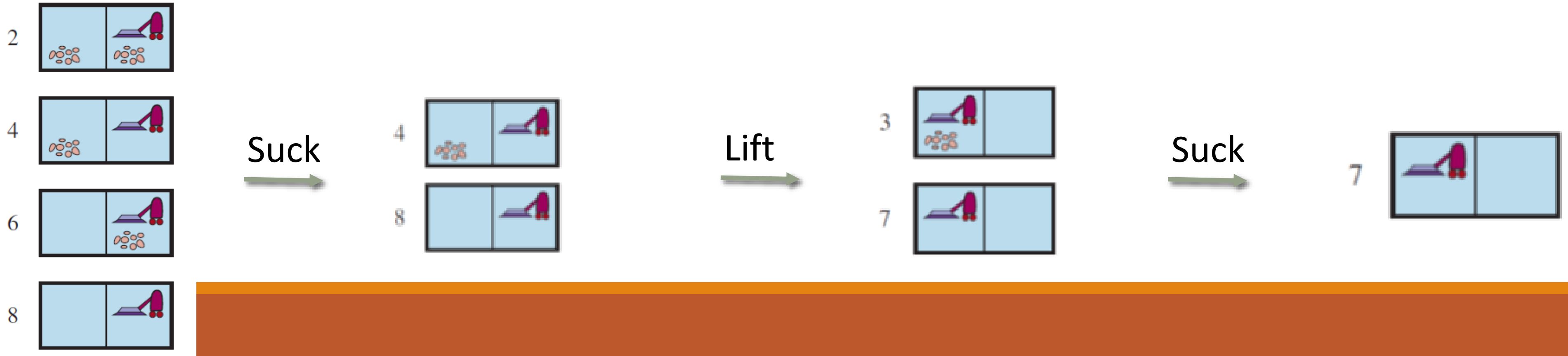
known environment , unknown location

Initial state ?(may be known or not!) but it is limited to state space

Actions : 1- based on current state

$$\text{ACTIONS}(b) = \bigcup_{s \in b} \text{ACTIONS}_P(s)\,.$$





known environment , unknown location

Imagine the light is down in home !! What u have to do ?

Initial state ?(may be known or not!) but it is limited to state space

Actions : 1- based on current state

$$\text{ACTIONS}(b) = \bigcup_{s \in b} \text{ACTIONS}_P(s)\,.$$

Actions : 2- based on current state (some actions are unsafe)→

Limited to safe actions.

Complete solution for vacuum world