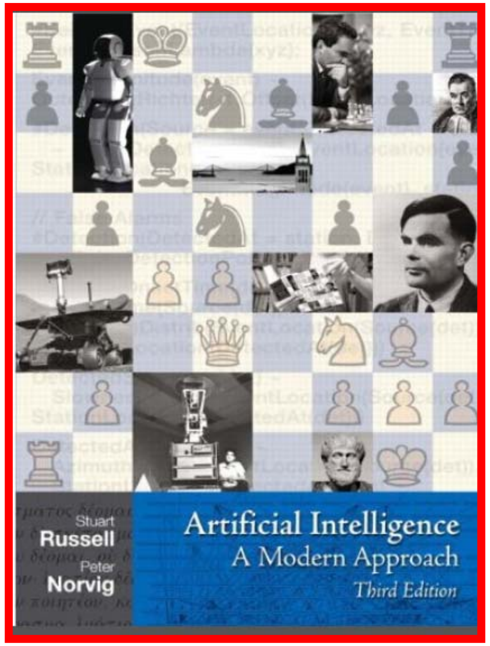


(Chapter-4-2)

BEYOND CLASSICAL SEARCH

Genetic Algorithm Search

Yanmei Zheng

Genetic Algorithm Search

Dr.Yanmei Zheng

Biological Basis

- **Biological systems adapt themselves to a new environment by evolution.**
  - **Generations of descendants are produced that perform better than do their ancestors.**
- **Biological evolution**
  - **Production of descendants *changed* from their parents**
  - ***Selective survival* of some of these descendants to produce more descendants**

Evolutionary Computation

- **What is the Evolutionary Computation?**
  - **Stochastic search (or problem solving) techniques that mimic the metaphor of natural biological evolution.**
- **Metaphor**

EVOLUTION		PROBLEM SOLVING
Individual	⟷	Candidate Solution
Fitness	⟷	Quality
Environment	⟷	Problem

Genetic Algorithms

- **Formally introduced in the US in the 70s by John Holland.**
- **GAs emulate ideas from genetics and natural selection and can search potentially large spaces.**
- **Before we can apply Genetic Algorithm to a problem, we need to answer:**

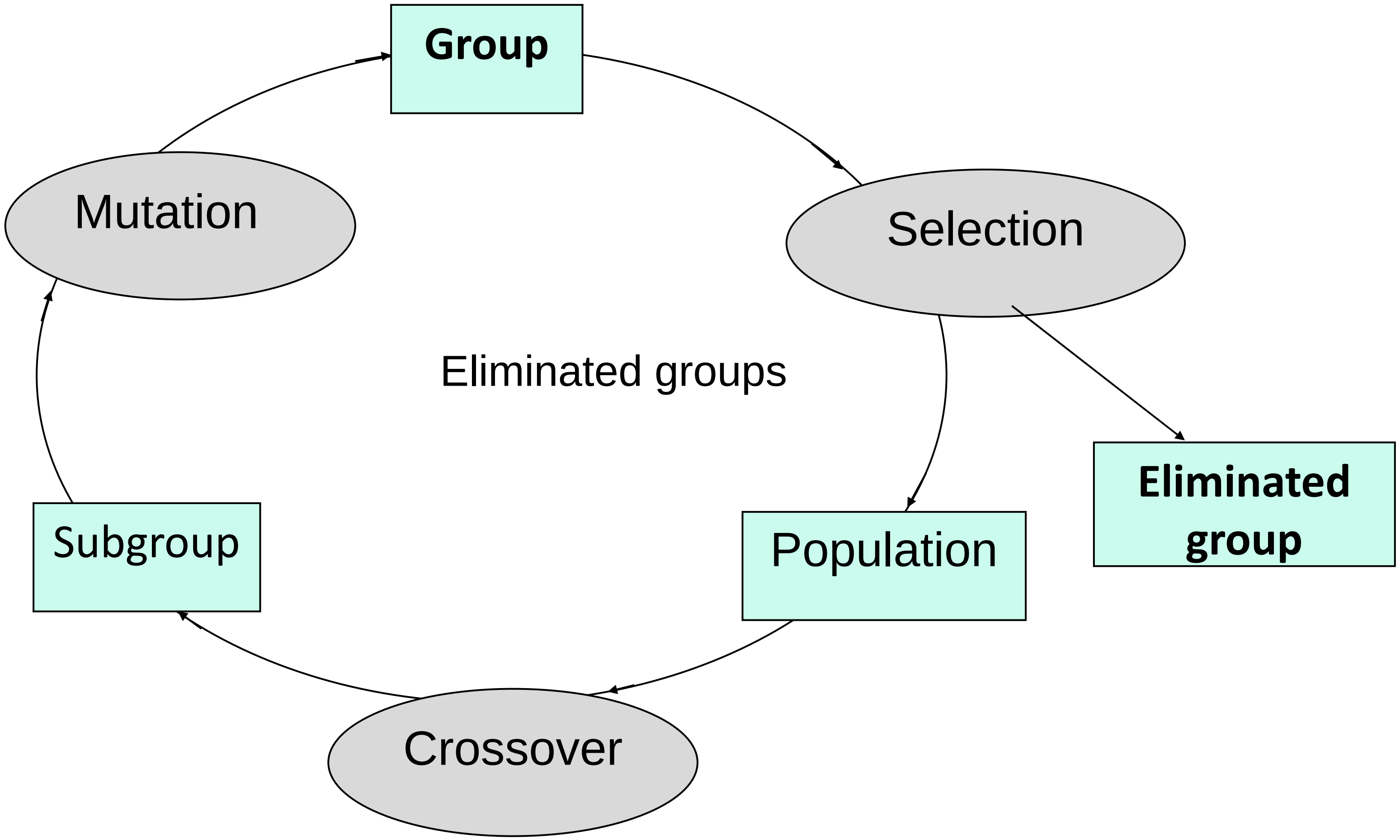
  - How is an individual represented?

  - What is the fitness function?

  - How are individuals selected?

  - How do individuals reproduce?

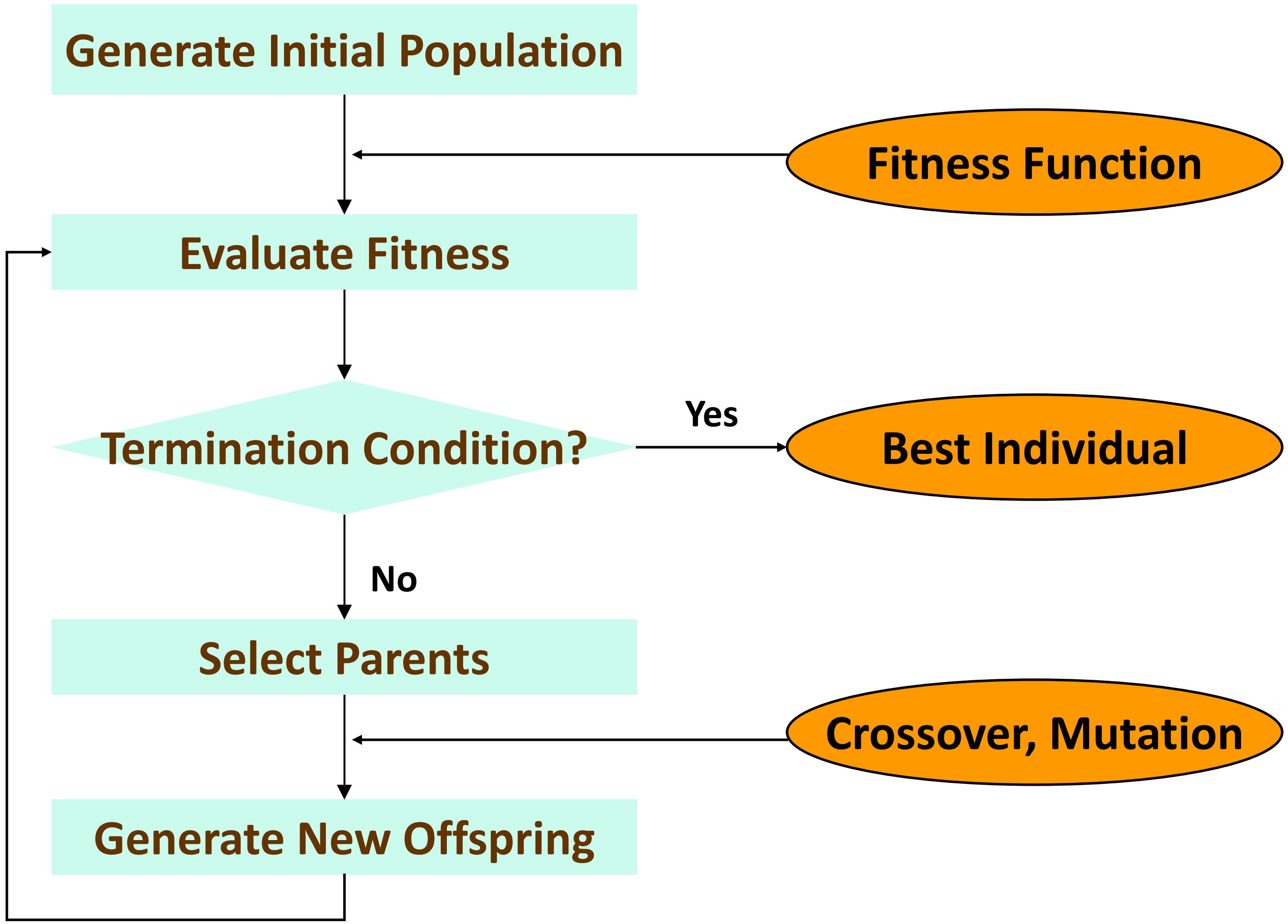
Biological evolution and genetic algorithms



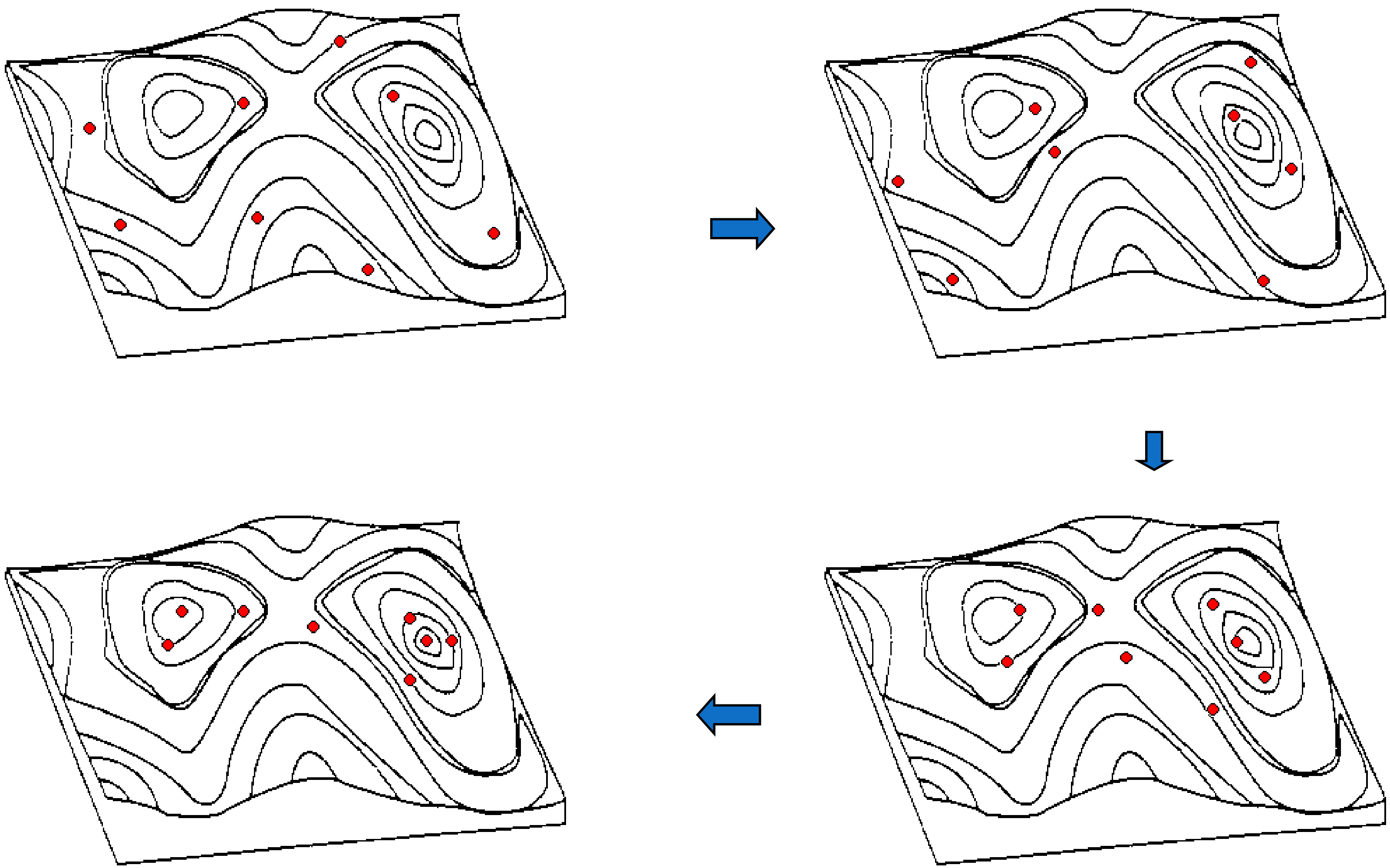
The correspondence between biological evolution and genetic algorithm

Concepts in biological evolution	The role of GA
The environment	The fitness function
Adaptive	Value of the fitness function
The survival of the fittest	The solution with the greatest value of the fitness function has the greatest probability of being retained
Individual	A solution of the problem
Chromosome	Encoding of the solution
Gene	Element of the Encoding
Group	A selected set of solutions
Population	A set of solutions selected according to the fitness function
Crossover	The process of producing offspring in a certain way from both parents
Mutation	The process of certain element of an encoding changes

General Framework of EC



Geometric Analogy
- Mathematical Landscape



Dr.Yanmei Zheng

Paradigms in EC

- **Evolutionary Programming (EP)**
  - **[L. Fogel et al., 1966]**
  - **FSMs, mutation only, tournament selection**
- **Evolution Strategy (ES)**
  - **[I. Rechenberg, 1973]**
  - **Real values, mainly mutation, ranking selection**
- **Genetic Algorithm (GA)**
  - **[J. Holland, 1975]**
  - **Bitstrings, mainly crossover, proportionate selection**
- **Genetic Programming (GP)**
  - **[J. Koza, 1992]**
  - **Trees, mainly crossover, proportionate selection**

(Simple) Genetic Algorithm (1)

- **Genetic Representation**
  - **Chromosome**
    - **A solution of the problem to be solved is normally represented as a *chromosome* which is also called an individual.**
    - **This is represented as a bit string.**

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

    - **This string may encode integers, real numbers, sets, or whatever.**
  - **Population**
    - **GA uses a number of chromosomes at a time called a *population*.**
    - **The population evolves over a number of *generations* towards a better solution.**

Genetic Algorithms: Representation of states (solutions)

- **Each state or individual is represented as a string over a finite alphabet. It is also called chromosome which Contains genes.**

genes

Solution: 607 → Encoding → 110000111

Chromosome:

Binary String

Genetic Algorithms: Fitness Function

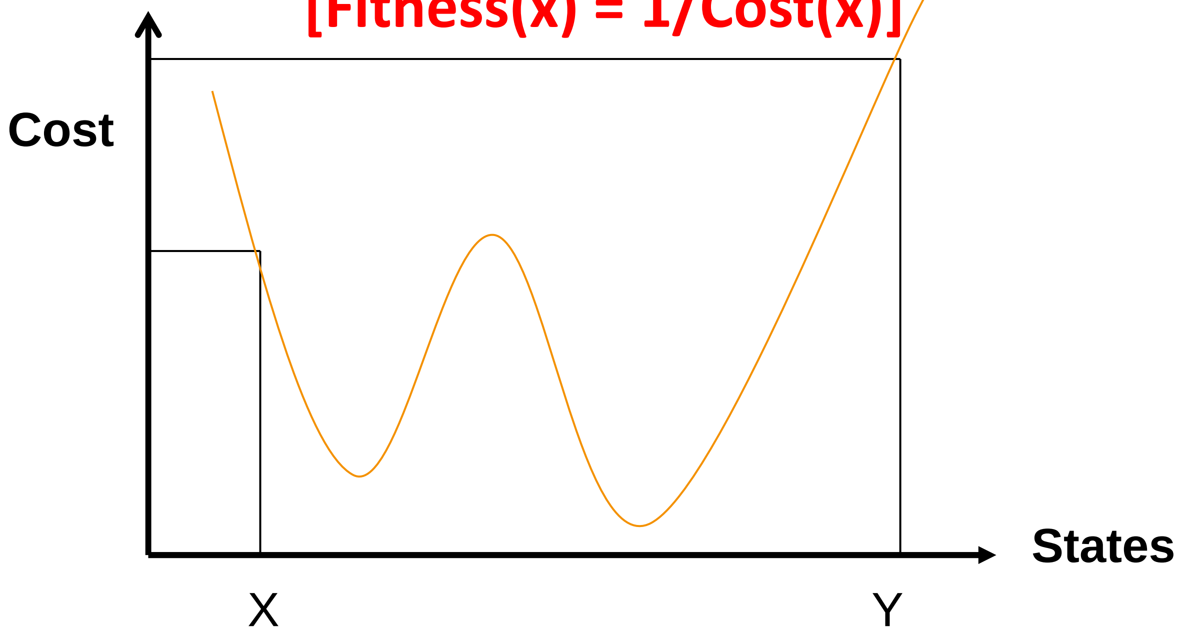
- **Fitness Function**
  - **The GA search is guided by a *fitness function* which returns a single numeric value indicating the *fitness* of a chromosome.**
  - **The fitness is maximized or minimized depending on the problems.**
  - **Eg)**
    1. **The number of 1's in the chromosome**
    2. **Numerical functions**

Genetic Algorithms: Fitness Function

Each state is rated by the evaluation function called fitness function. Fitness function should return higher values for better states:

Fitness(X) should be greater than Fitness(Y) !!

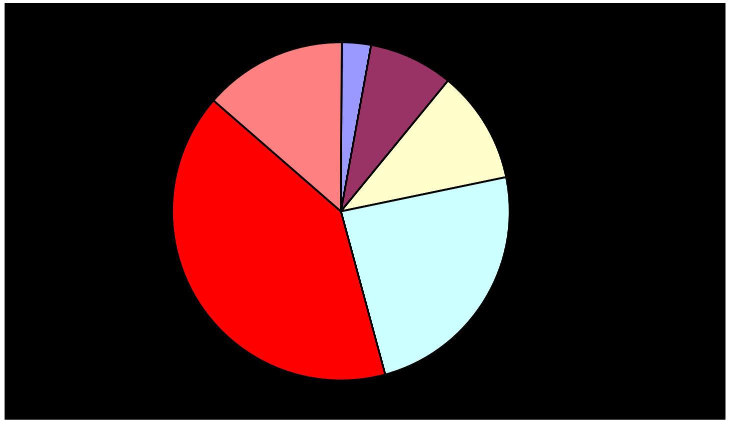
[Fitness(x) = 1/Cost(x)]



Genetic Algorithm (3)

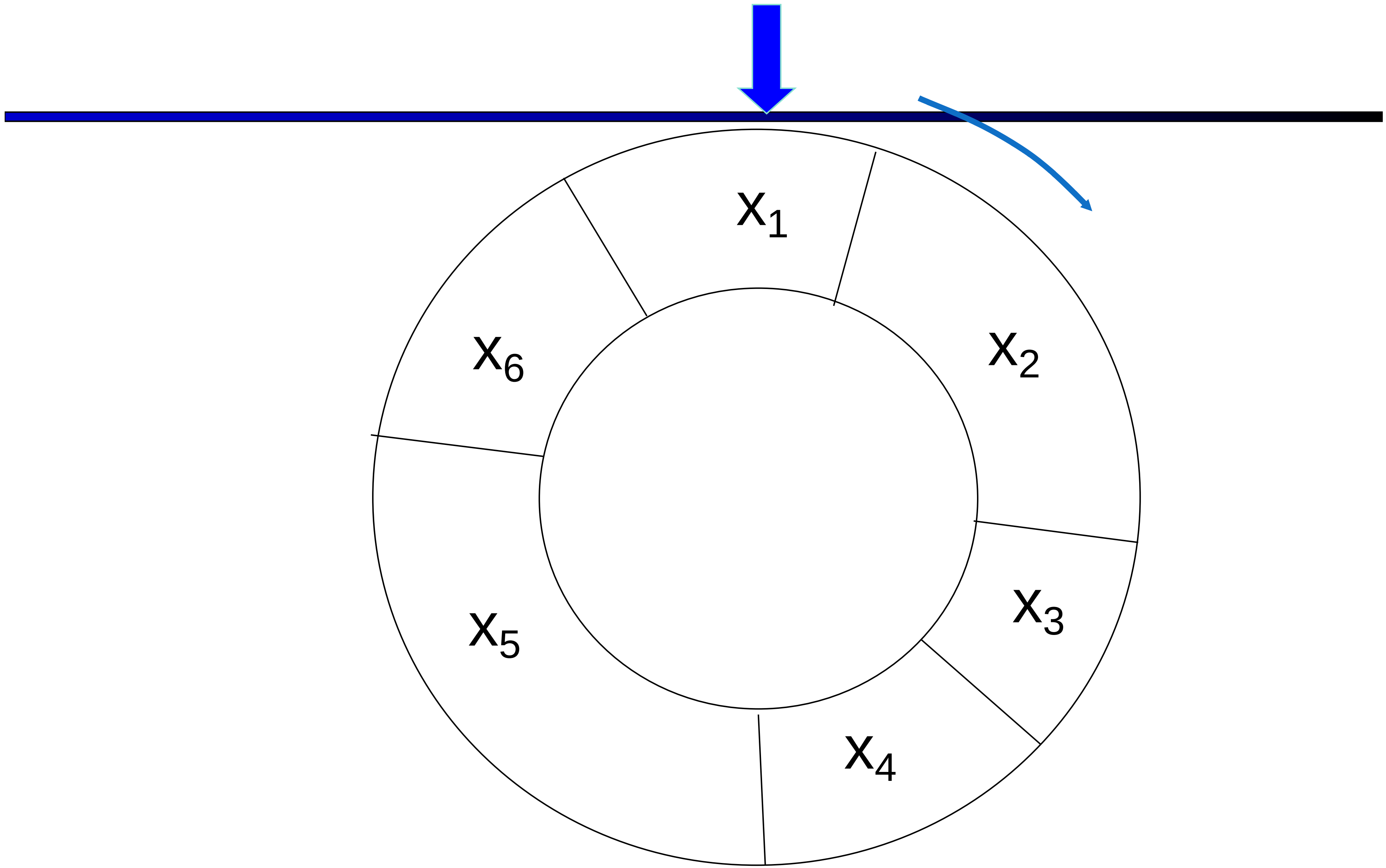
- Selection
  - Selecting individuals to be parents
  - Chromosomes with a higher fitness value will have a higher probability of contributing one or more offspring in the next generation
  - Variation of Selection
    - Proportional (Roulette wheel) selection
    - Tournament selection
    - Ranking-based selection

GA Parent Selection - Roulette Wheel



- **Sum the fitnesses of all the population members, *TF***
- **Generate a random number, *m*, between 0 and *TF***
- **Return the first population member whose fitness added to the preceding population members is greater than or equal to *m***

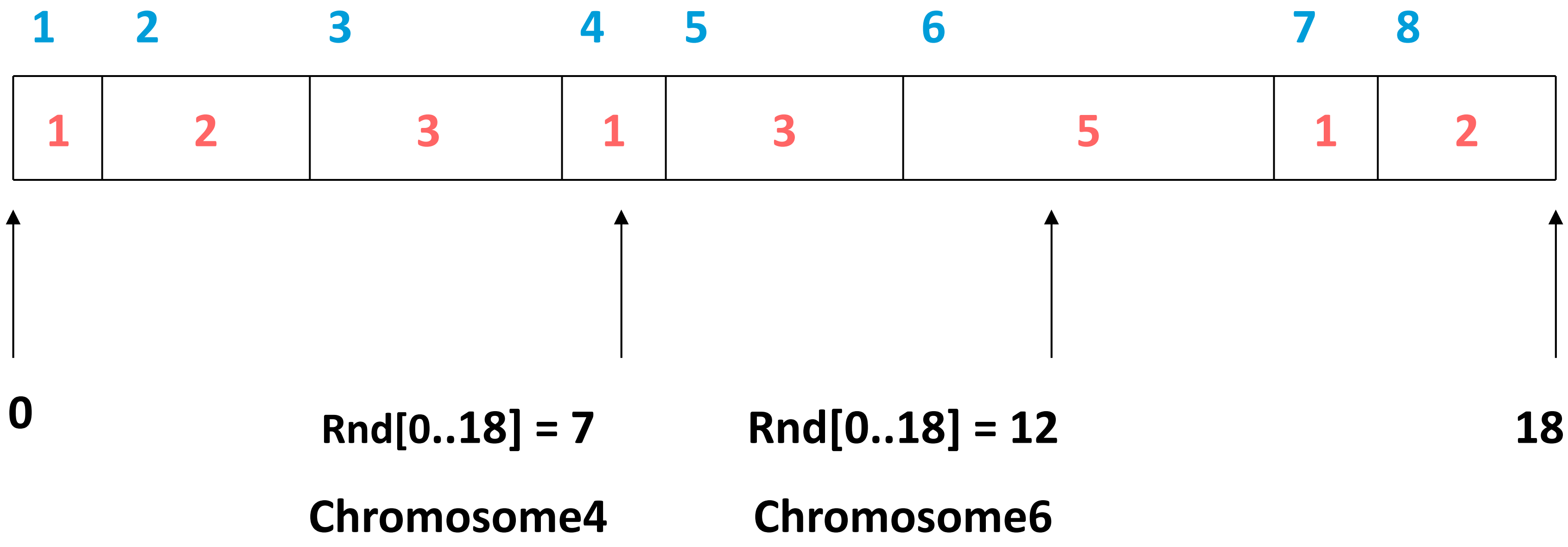
Roulette Wheel Selection


x_1
x_2
x_3
x_4
x_5
x_6

Genetic Algorithms: Selection

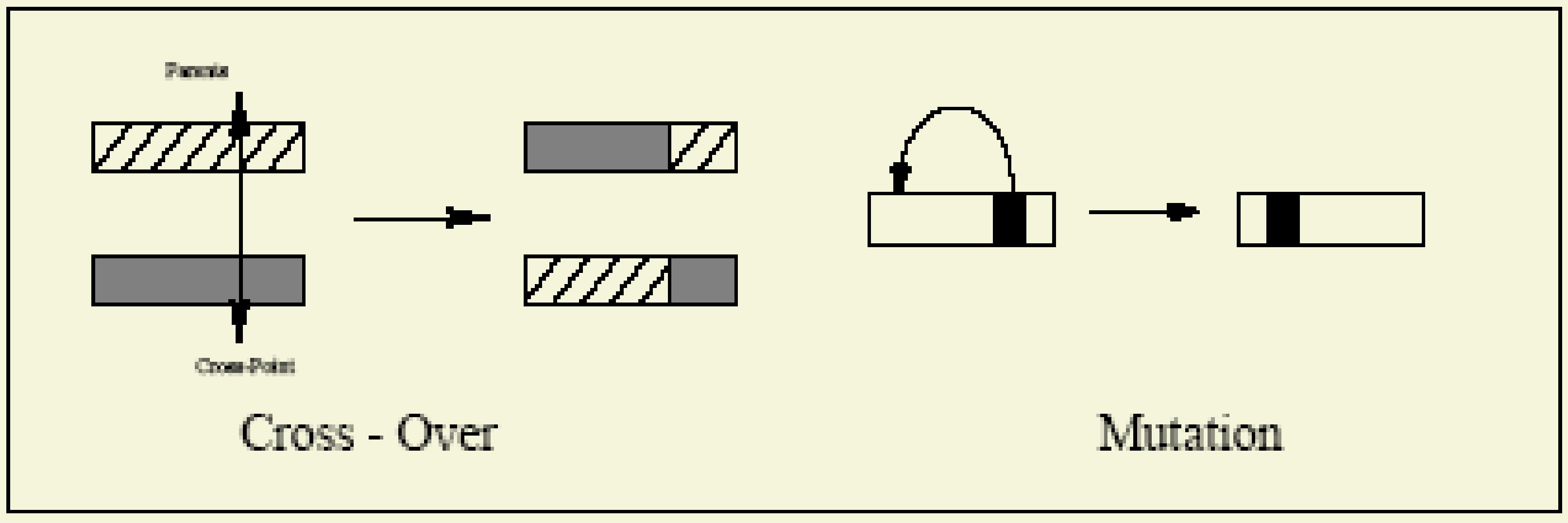
How are individuals selected ?

Roulette Wheel Selection



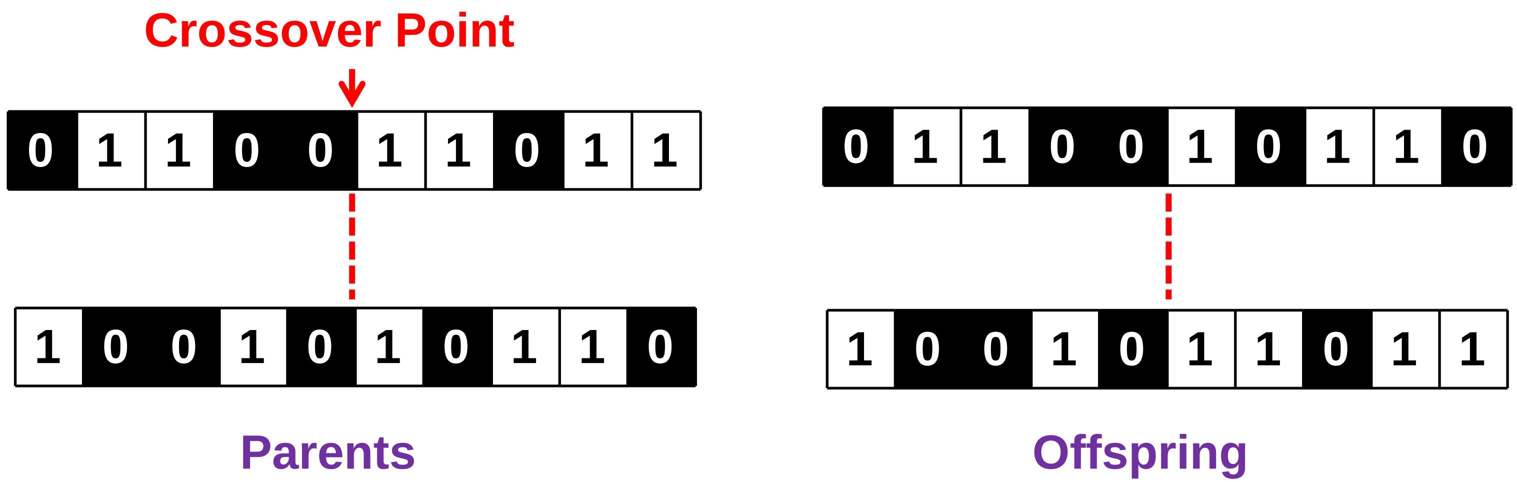
Genetic Algorithms: Cross-Over and Mutation

How do individuals reproduce ?



Genetic Algorithms Crossover - Recombination

- **Genetic Operators**
  - **Crossover (1-point)**
    - **A crossover point is selected at random and parts of the two *parent* chromosomes are swapped to create two *offspring* with a probability which is called *crossover rate*.**



    - **This mixing of genetic material provides a very efficient and robust search method.**
    - **Several different forms of crossover such as *k*-points, uniform**

Genetic Algorithms Crossover - Recombination

1010000000	Parent1	Offspring1	1011011111
1001011111	Parent2	Offspring2	1000000000

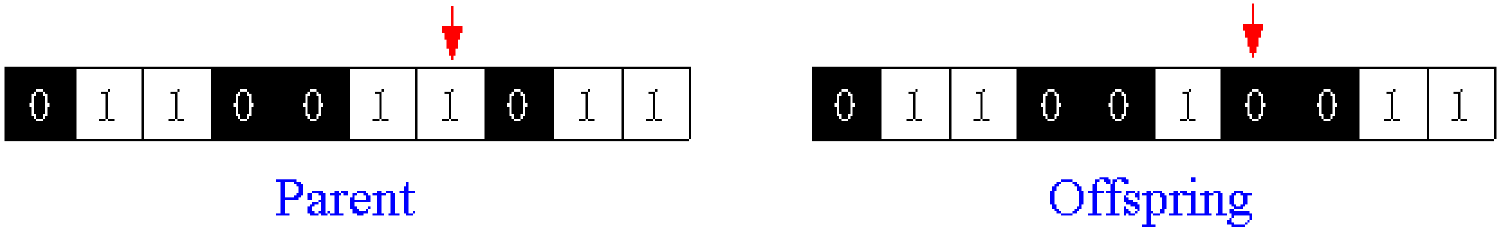
Crossover single point - random

With some high probability (crossover rate) apply crossover to the parents. (typical values are 0.8 to 0.95)

Dr.Yanmei Zheng

Genetic Algorithms Mutation

- **Mutation**
  - **Mutation changes a bit from 0 to 1 or 1 to 0 with a probability which is called *mutation rate*.**
  - **The mutation rate is usually very small (e.g., 0.001).**
  - **It may result in a random search, rather than the guided search produced by crossover.**



- **Reproduction**
  - **Parent(s) is (are) copied into next generation without crossover and mutation.**

Genetic Algorithms Mutation

mutate

Offspring1 1011011111

Offspring2 1000000000

Original offspring

Offspring1 1011001111

Offspring2 1010000000

Mutated offspring

With some small probability (the *mutation rate*) flip each bit in the offspring (*typical values between 0.1 and 0.001*)

Genetic Algorithm

- Given: population P and fitness-function f
- repeat
  - newP ← empty set
  - for *i* = 1 to size(P)

    x ← RandomSelection(P,f)

    y ← RandomSelection(P,f)

    child ← Reproduce(*x*,*y*)

    if (small random probability) then child ← Mutate(child)

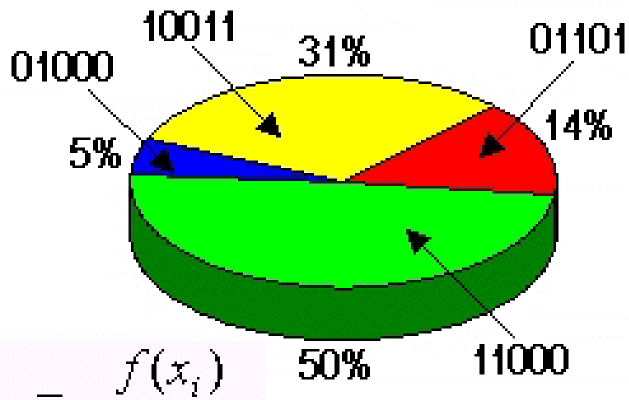
    add *child* to newP
  - P ← newP
- until some individual is fit enough or enough time has elapsed
- return the best individual in P according to f

Example of Genetic Algorithm

The Problem is to Maximize $f(x) = x^2$

Number	String	Fitness	% of the Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

1. *Roulette Wheel Selection*



$$p_{s_i} = \frac{f(x_i)}{\sum_{k=1}^{n} f(x_k)}$$

2. *One-Point Crossover*

$P_c \in [0.6 \,..\, 1.0]$

parents	offspring
01 | 101 (169)	01000 (64)
11 | 000 (576)	11101 (841)

3. *Mutation*

$P_m \in [0.001 \,..\, 0.1]$

Mutate the first allele

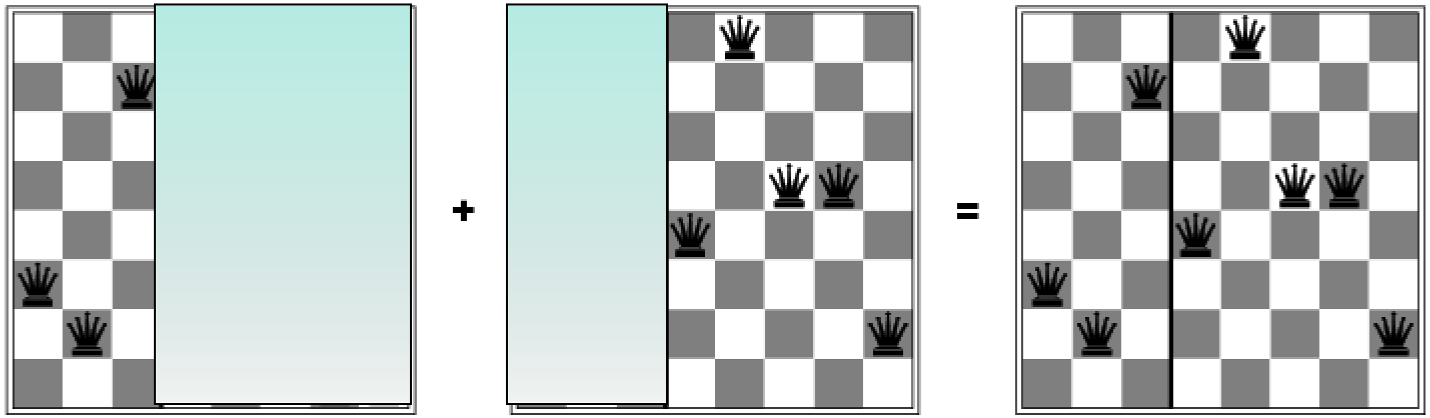
0	1	0	0	0
4	3	2	1	0

fitness=64 ⇨ fitness=576

1	1	0	0	0
4	3	2	1	0

Dr.Yanmei Zheng

Genetic algorithms



Has the effect of "jumping" to a completely different new part of the search space (quite non-local)

First Project: 8-queens Problem

- **To formulate 8-queens problem into the search problem in a state space**
- **To define a evaluation function**
- **To implement local Algorithm to estimate the rate of falling into local minimums.**
- **To write a report on the simulation result.**

Thank you

End of

Chapter 4-2

Dr.Yanmei Zheng