Lecture #3

Evolutionary Computation

Dr. Charnon Pattiyanon
Assistant Director of IT, Instructor
Department of Artificial Intelligence and Computer Engineering
CMKL University

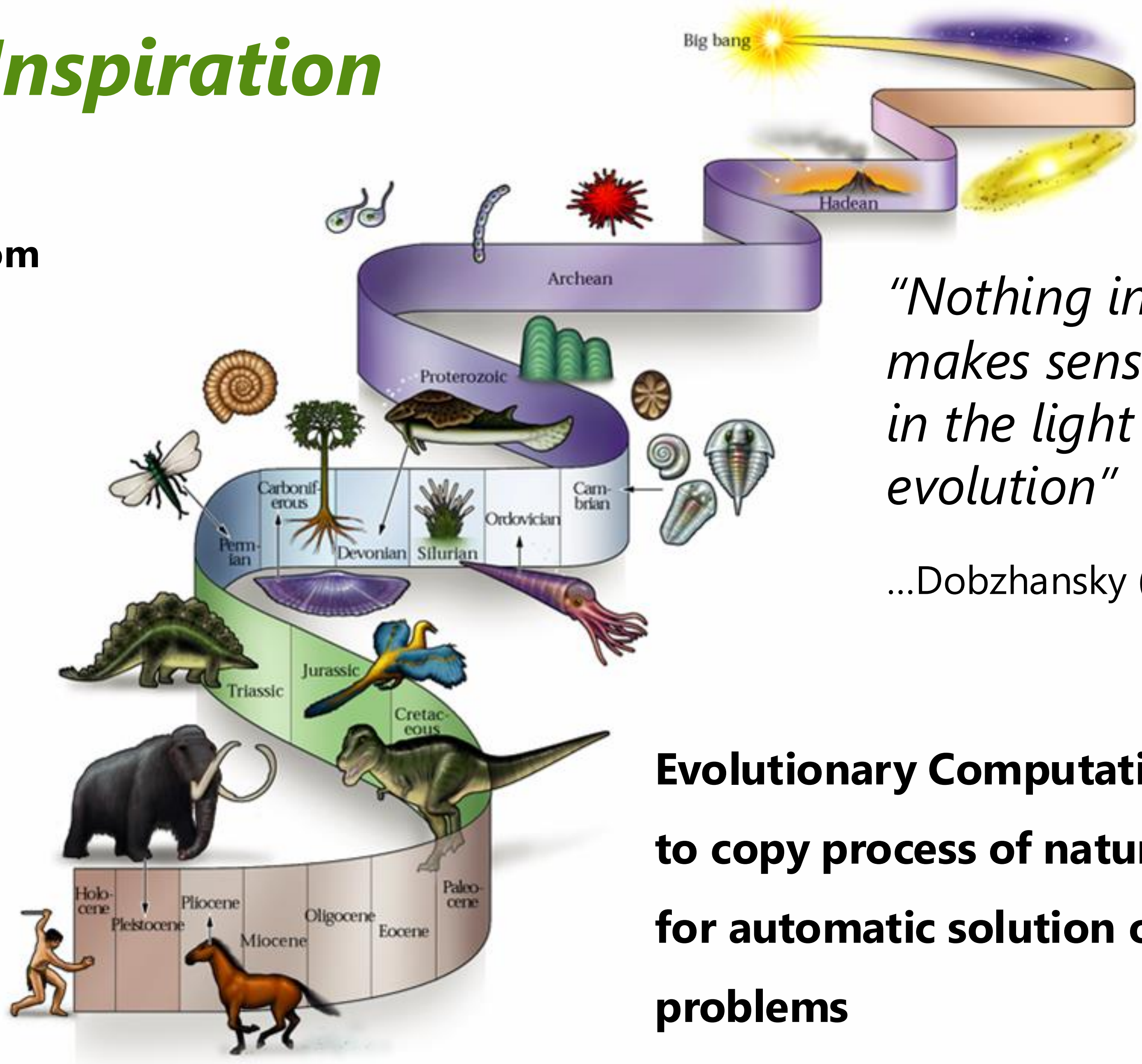
Modified from Companion Slides for “D. Floreano and C. Mattiussi, Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies, 2008”

Evolutionary Inspiration

Biological systems result from an evolutionary process

Biological systems are

- Robust
- Complex
- Adaptive



"Nothing in biology makes sense except in the light of evolution"

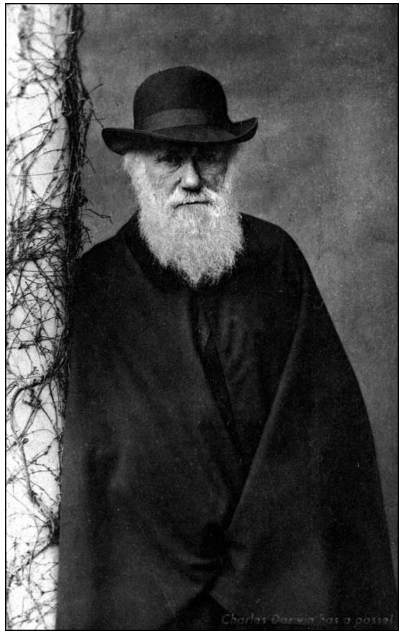
...Dobzhansky (1973)

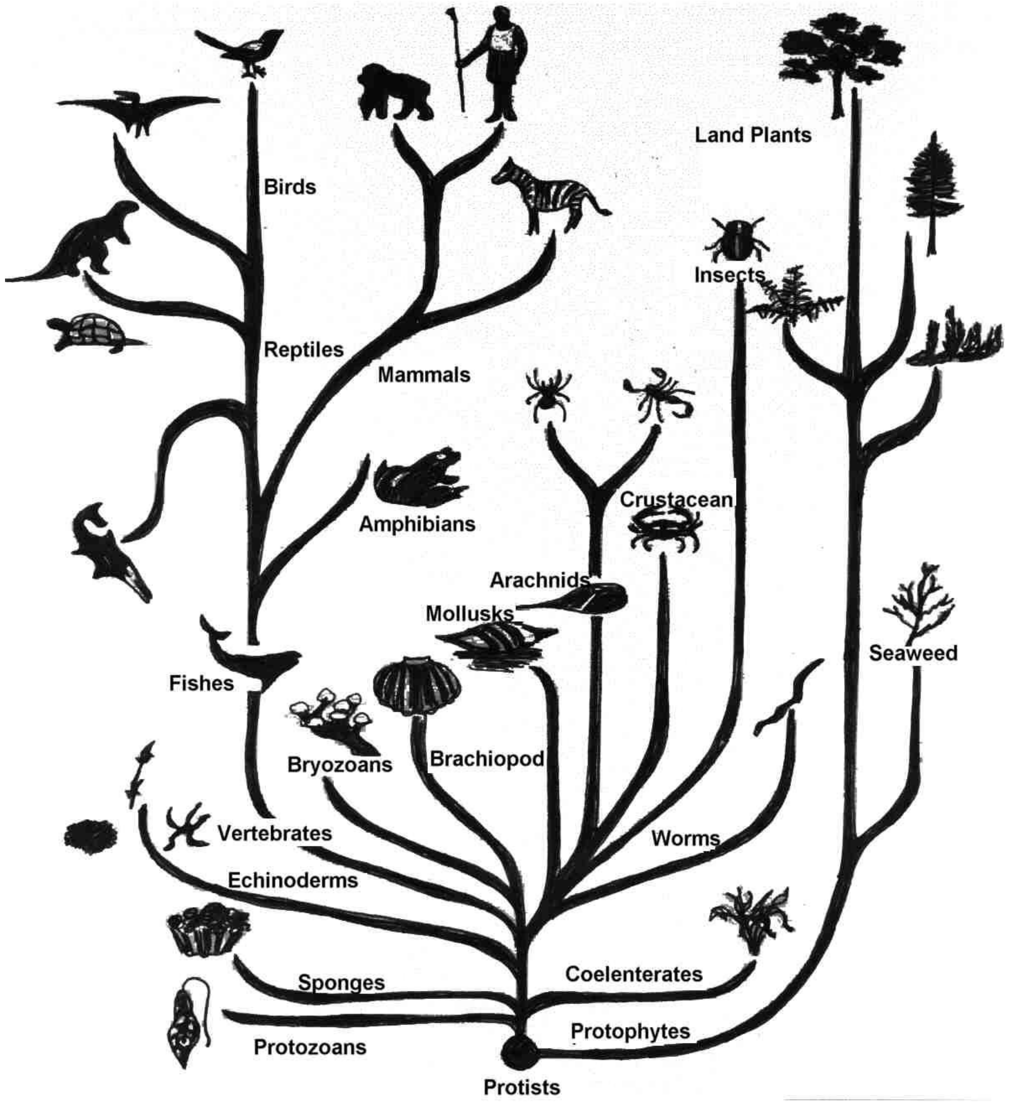
Evolutionary Computation attempts to copy process of natural evolution for automatic solution of complex problems

The Four Pillars of Evolution

"All species derive from common ancestor"

Charles Darwin, 1859
On the Origins of Species





- **Population**

  Group of several individuals

- **Diversity**

  Individuals have different characteristics.

- **Heredity**

  Characteristics are transmitted over generations.

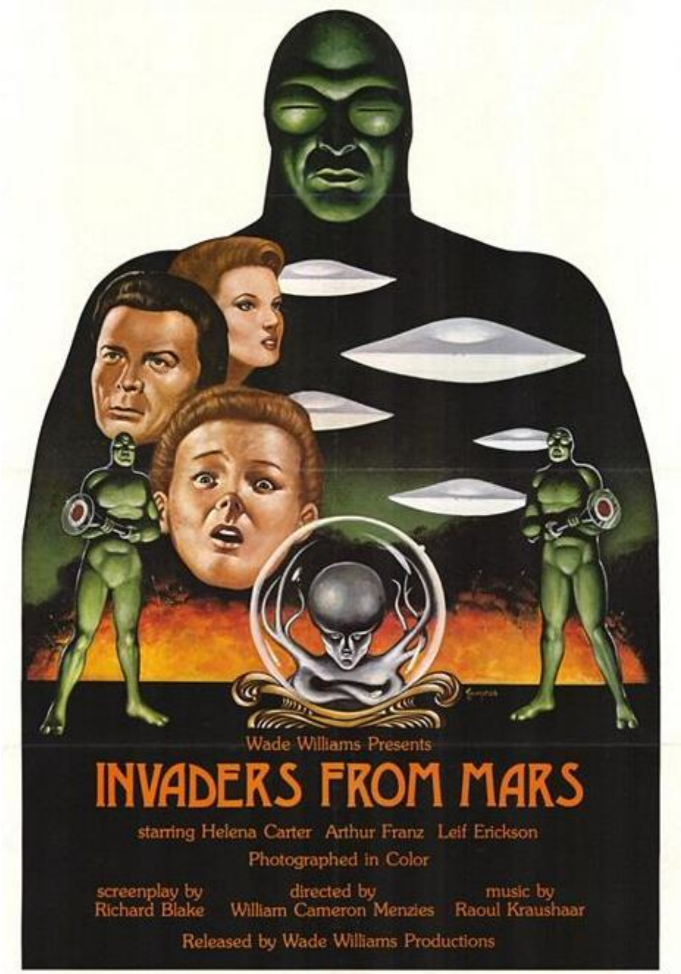
- **Selection**

  - Individuals make more offspring than the environment can support.

  - Better at food gathering = better at surviving = make more offspring.

Evolution Without Progress

… or "why we should not fear an invasion from Mars" (Gould, 1997)



Humans are **not** the **top of the evolutionary ladder** (misleading image of evolution with humans at top or end).

Evolution without Progress:

Selective reproduction of the fittest does not necessarily imply progress.

- If no competition, no selection of the fittest
- Individuals selected against current environment

Natural selection has no comparative memory

It is not guaranteed that recent generations are comparatively better than older generations selected in different environment conditions.

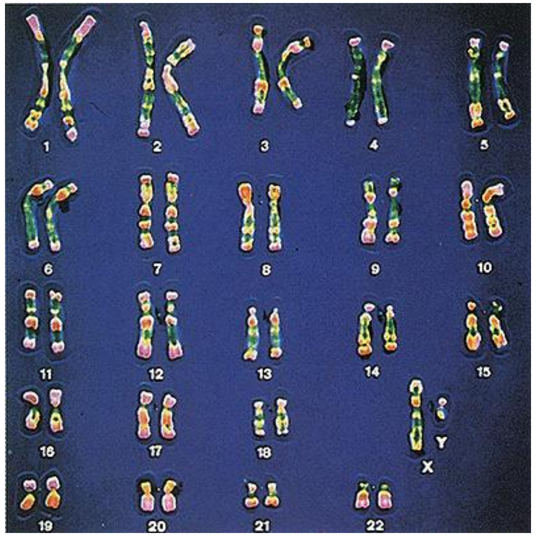
- Accumulation of change with no cost or benefit (also known as ***Neutral Evolution***)

Evolutionary Algorithms

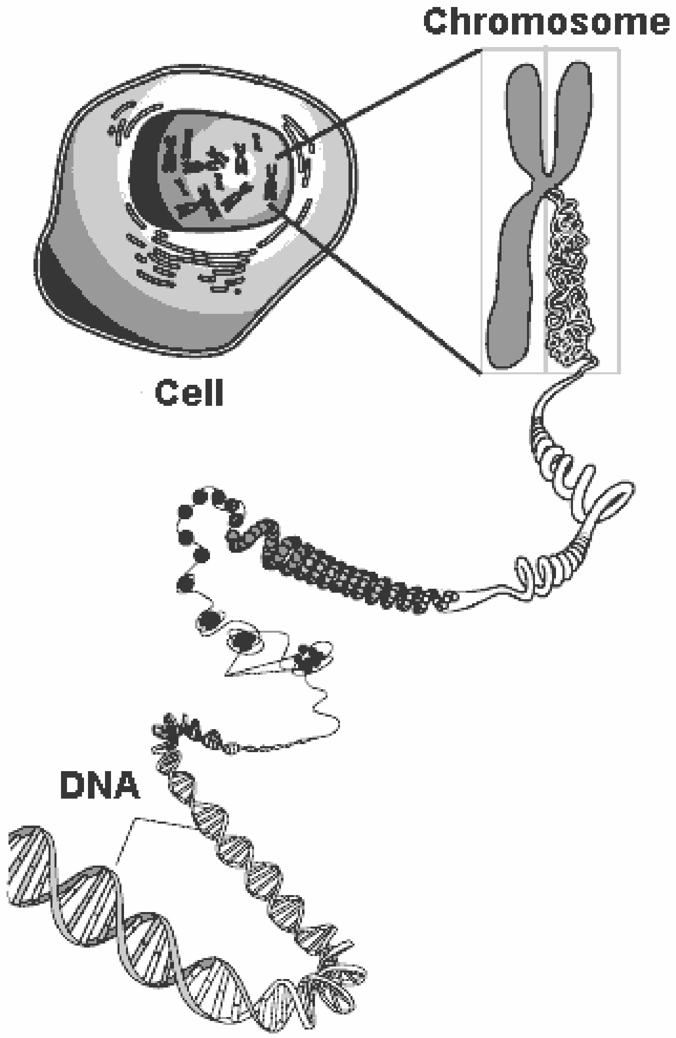
- Inspired by biological evolution
- **Based on ideas from:**
  - Darwin's principles of natural and sexual selection
  - Mendel's particulate model of inheritance (dominant gene versus recessive gene), and
  - Hardy and Weinberg's work on population genetics
- **Survival of the fittest** (*if no competition, no selection of the fittest*)
- **The best genes are transferred to the next generation** (*individuals selected against currrent environment*)
- **Reproduction process**

DNA (DeoxyriboNecleic Acid)

Long Molecule, Twited in Spiral, and Compressed

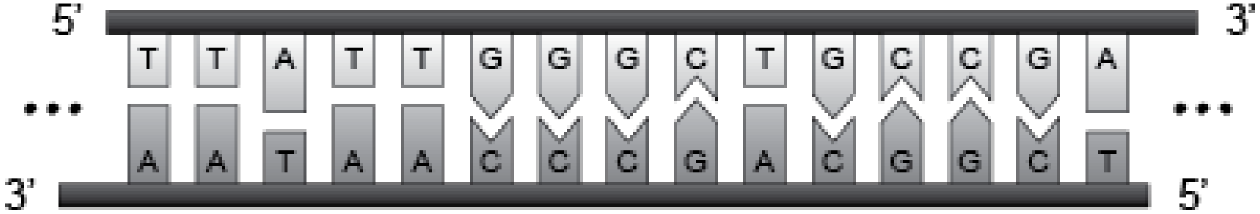


Humans have 23 pairs of DNA molecules (Chromosomes)



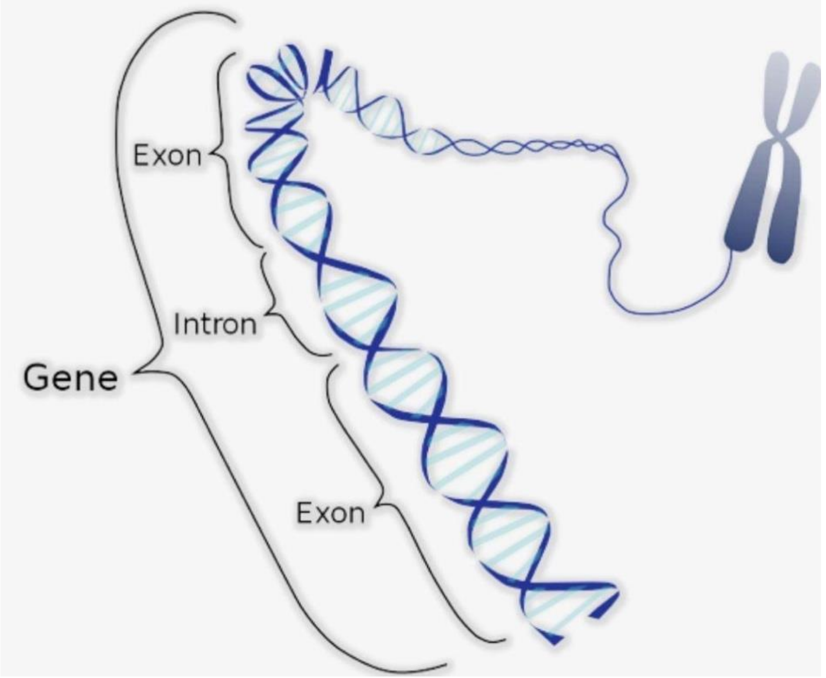
DNA is composed of 2 complementary sequences (strands) of 4 nucleotides (A, T, C, G), which bind together in pairs (A-T and C-G)

Adenine (A)
Cytosine (C)
Guanine (G)
Thymine (T)

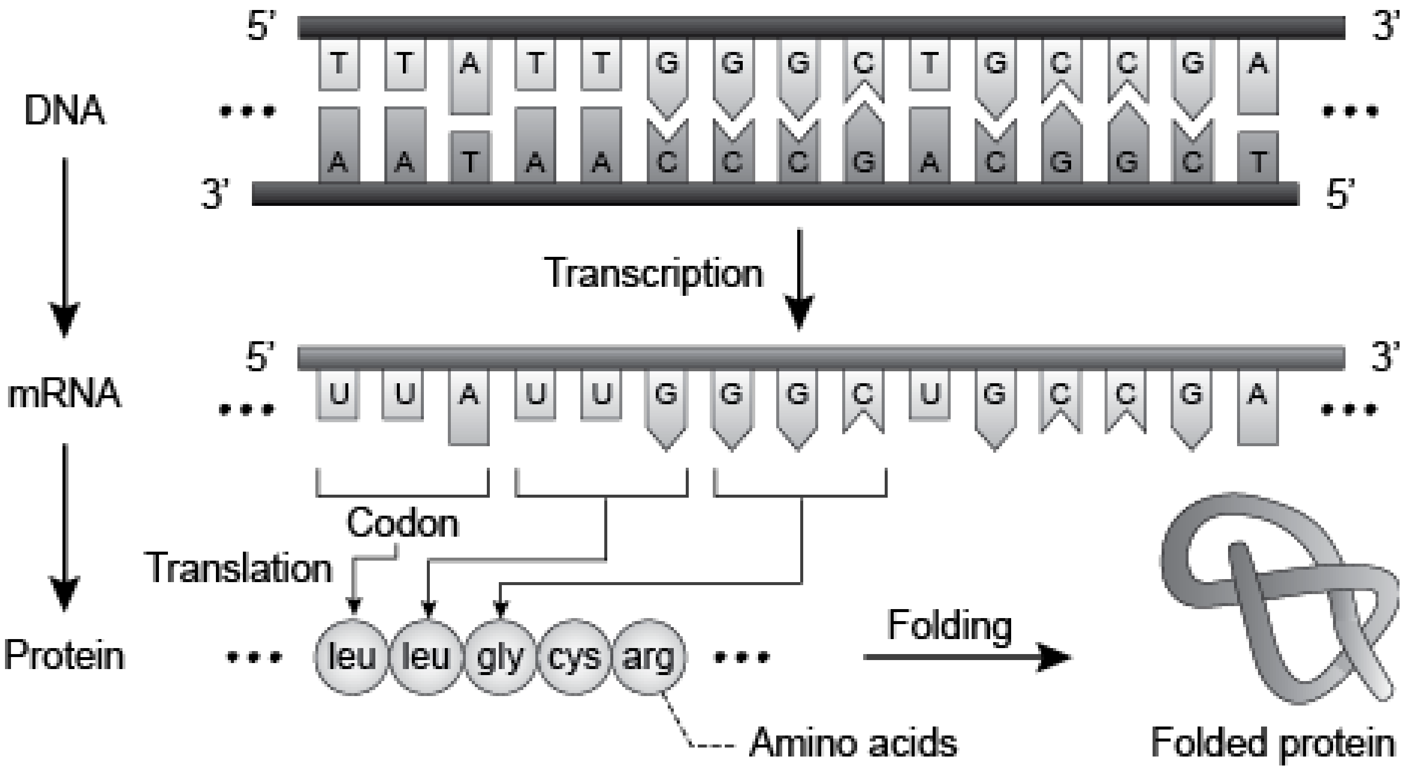


The DNA is the genetic material that is transmitted over generations

From Genes to Protein *(Gene Expression)*



- A **gene** is a sequence of several nucleotides that produce a protein.
- **Proteins** are molecules that define the type and function of cells (hair and muscle cells are made of different proteins)



The expression of the gene into a protein is mediated by another molecule, known as **messenger RNA**.

Genotype and Phenotype

- **Genotype** = the genetic material of the organism.

  It is transmitted during the reproduction and affected by mutations;

  Selection does not operate directly on it.

- **Phenotype** = the manifestation of the organism (appearance, behavior, etc.)

  Selection operates on the phenotype;

  It is affected by environment, development, and learning.

Genetics = Structure and operation of genes

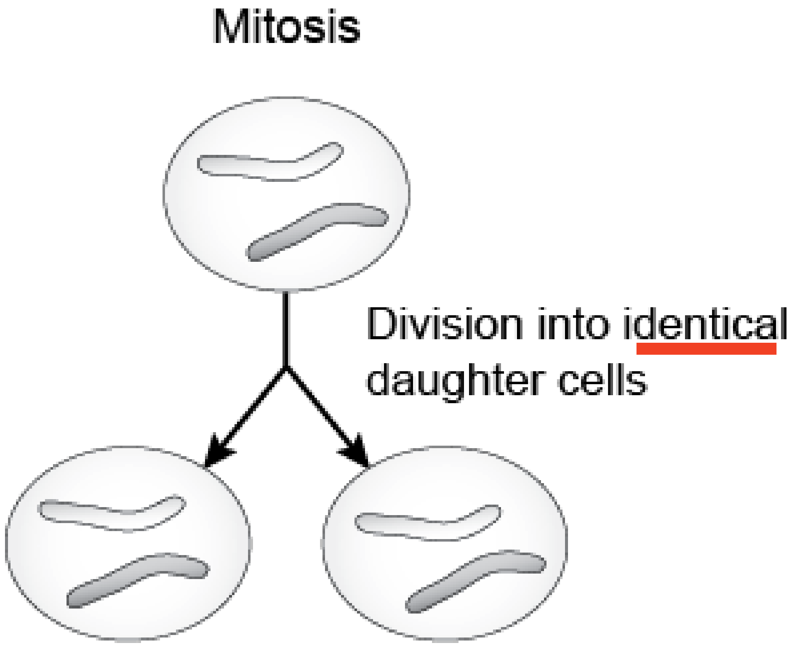
Functional genomics = role of genes in the organism

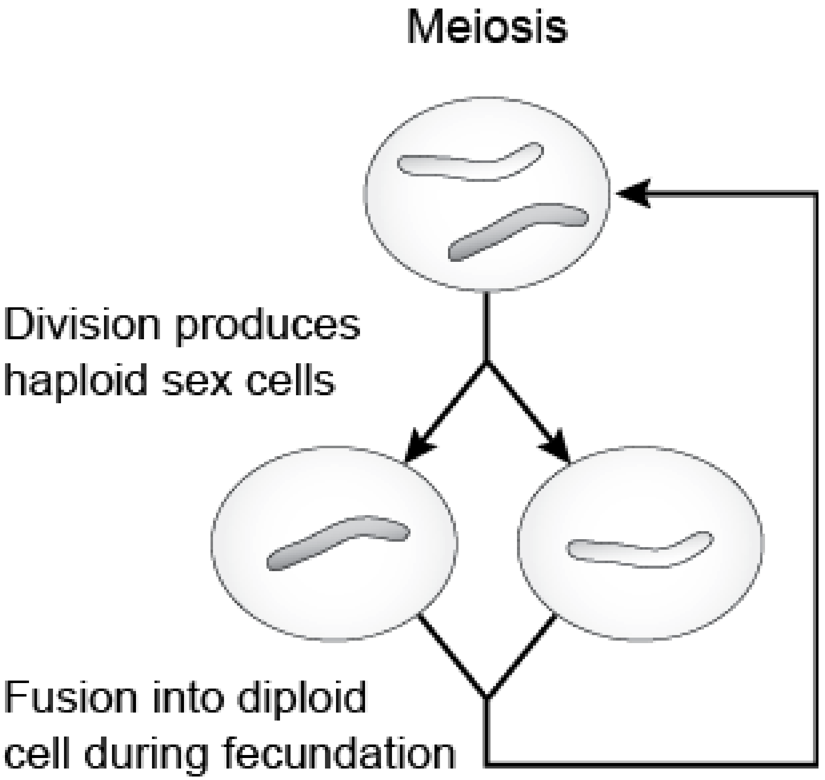
What do monozygous twins share with each other?

Jean-Felix & Auguste Piccard

Cell Replication

- Cells replicate in two ways:
  - **Mitosis:** during growth/maintenance of the organism
  - **Meiosis:** during production of sex cells (sperm and egg)

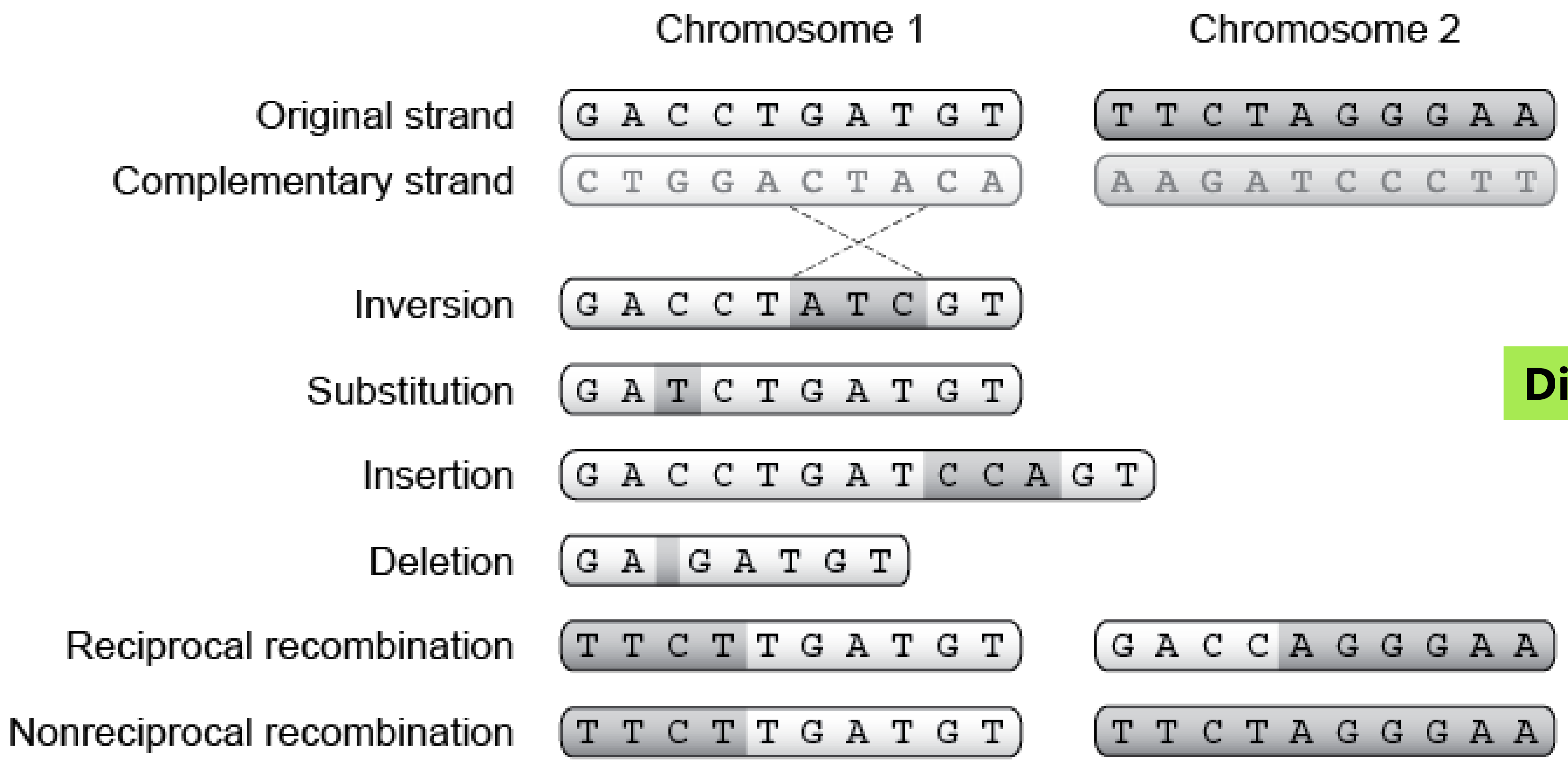




The two strands are separated.
Each strand goes to one cell and rebuilds the missing strand.

Genetic Mutations

- Genetic mutations occur during cell replication (4^{-10} per nucleotide per year)
- Those that occur in sex cells can affect evolution.
- Recombination is a mutation that affects two homologous chromosomes.



Diversity

Evolutionary Computation (EC) refers to computer-based **problem solving systems** that use computational **models of evolutionary processes**, such as

- Natural Selection,
- Survival of the fitness and
- Reproduction,

as the fundamental components of such computational systems.

Utilization as Intelligent Systems

Problem-Solving Techniques

Choosing the best element from some set of available alternatives.

Automatically find solutions to hard **optimization problems**, i.e.:

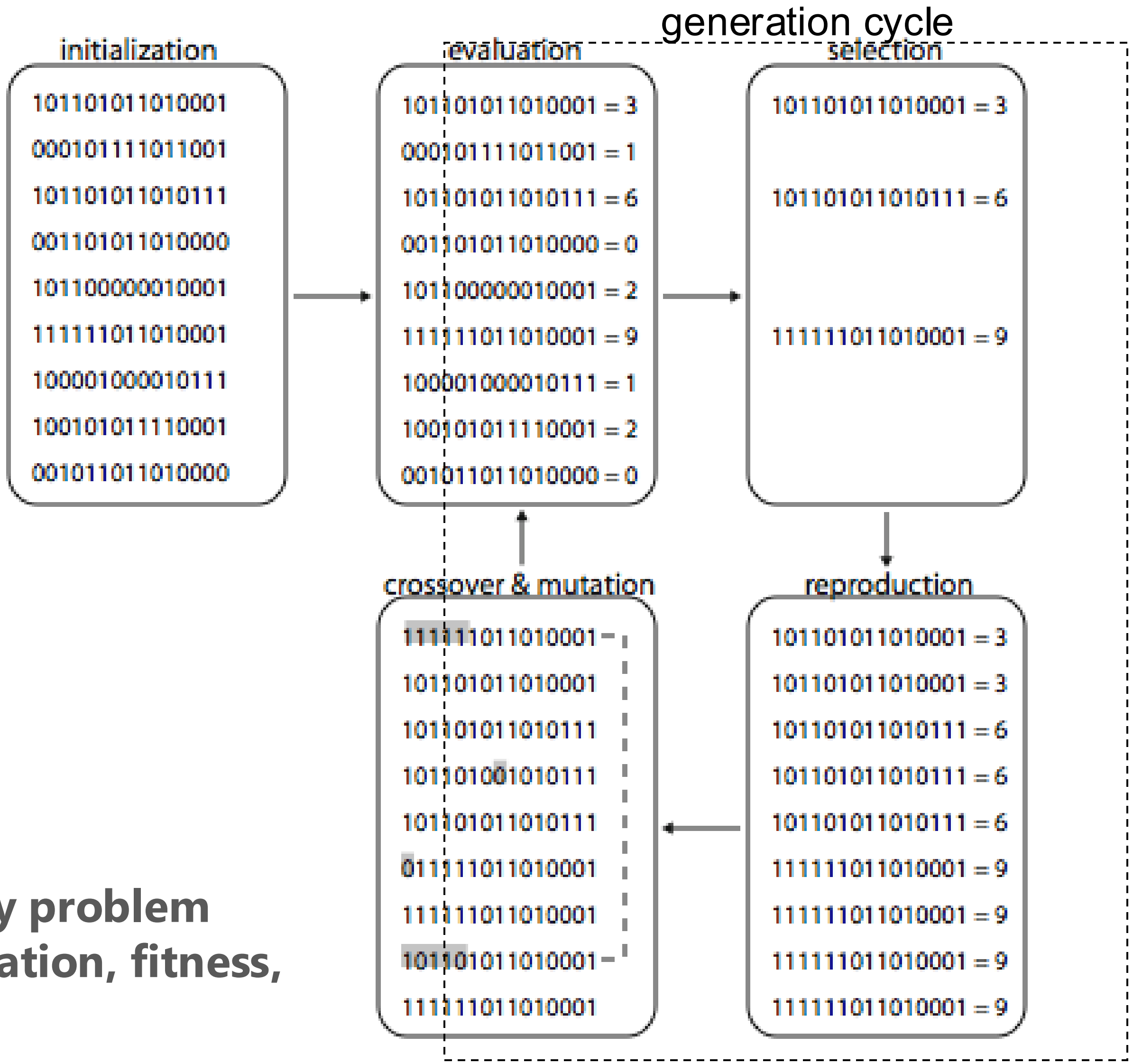
- Improve object shapes,
- Discover novel computer programs,
- Design electronic circuits, and
- Explore several other areas that are usually addressed by human design.

Natural Evolution Versus Artificial Evolution

Natural Evolution	Artificial Evolution
• **Phenotype**	Solutions to a problem
• **Genotype**	Genetic representation of those solutions
• **Population**	Maintenance of a population • Initialization • Maintain the population
• **Diversity**	Creation of diversity • Perform genetic mutation
• **Selection**	Selection mechanism • Evaluation of the phenotype • Reproduction the better phenotype.
• **Inheritance**	Process of genetic inheritance.

Algorithm of Evolutionary Computation

- **Devise genetic representation**
- **Build a population**
- **Design a fitness function**
- **Choose selection method**
- **Choose crossover & mutation**
- **Choose data analysis method**



Repeat generation cycle until:

- Maximum fitness value is found
- Solution found is good enough
- No fitness improvement for several generatio

Evolutionary algorithms are applicable to any problem domain as long as suitable genetic representation, fitness, and genetic operators are chosen.

Evolutionary algorithms are often used on **hard problems** where other optimization methods fail or are trapped in **suboptimal** solution.

Those problems typically include cases that:

- Have several free parameters with complex and non-linear interactions,
- Are characterized by non-continuous function,
- Have missing or corrupted data, or
- Display several local optima.

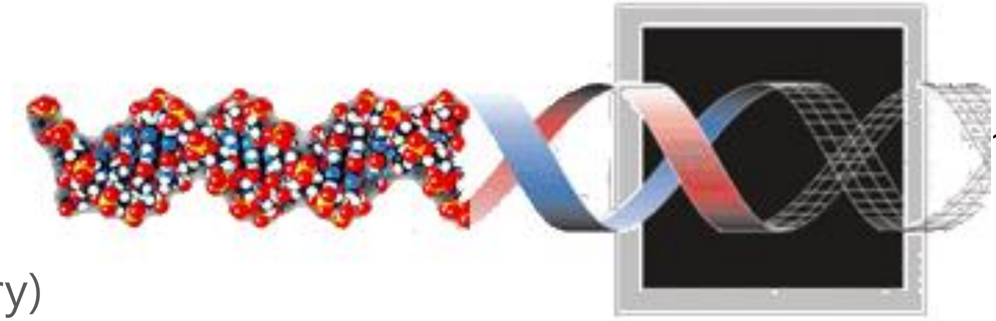
Genetic Representation

A **sutiable** genetic representation should be devised so that:

- The recombination and mutation operators have a **high likelihood** of generating increasingly **better individuals**.
- Set of possible genotypes **should include optimal solution** to the problem.

Choice of representation benefits from domain knowledge:

- Encoding of relevant parameters
- Appropriate resolution of parameters



Great simplification of genetics:

- Single stranded sequence of characters (e.g., binary)
- Fixed length
- Oftern one chromosome
- Often one-to-one direct correspondence between gene and parameter
- Gene expression and genetic regulation used only in specific situations

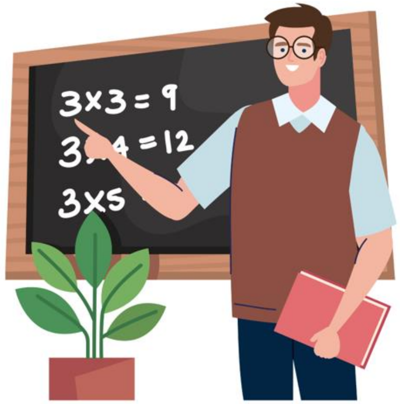
Discrete Representation

A sequence of l discrete values drawn from alphabet with cardinality k

- E.g., binary string of 8 positions ($l = 8, k = 2$): 01010100
- Can be mapped into several phenotypes:

To configuration string of FPGA electronic circuit

To integer i using binary code



01010100

84

0.328125

To real value r in range $[min, max]$

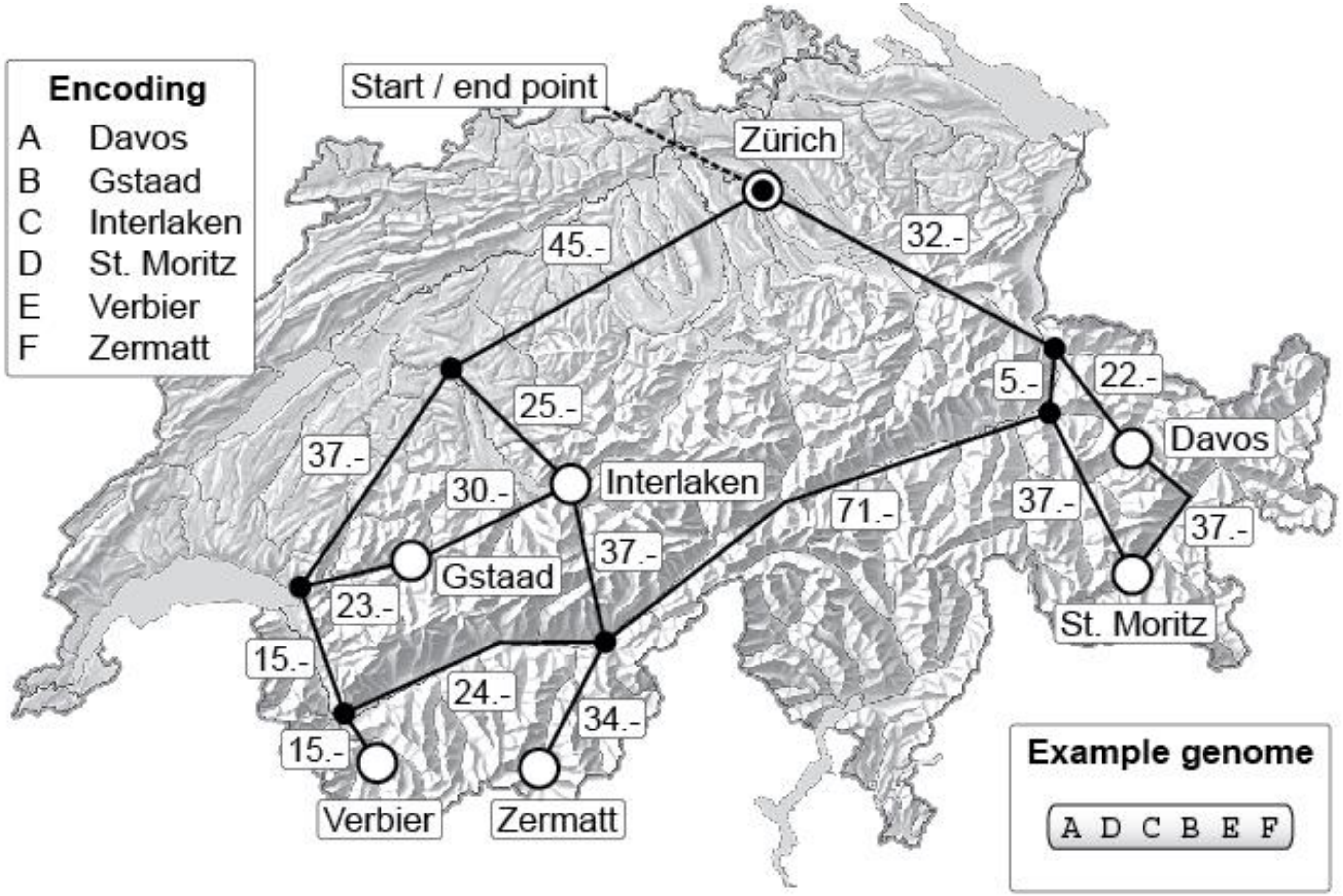
$$r = \min + (\frac{i}{255})(\max - min)$$

Job	A.M.	P.M.
1	x	
2		x
3	x	
4		x
5	x	
6		x
7	x	
8	x	

To job schedule:
- Job = gene position
- Time = gene value

Sequence Representation

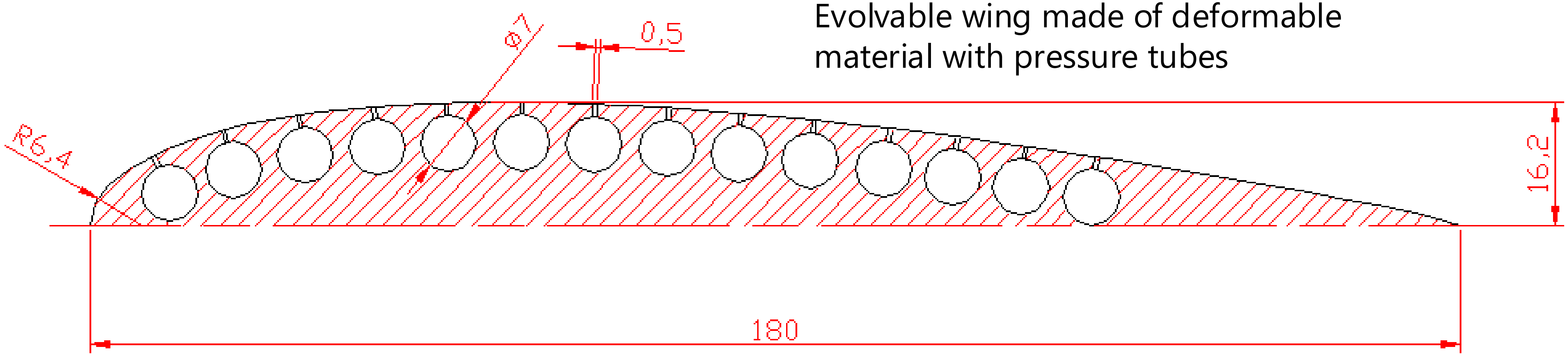
It is a particular case of discrete representation used for class of **Travelling Salesman Problems**, or **TSPs** (plan a path to visit n cities under some constraints), e.g., planning ski holidays with lowest transportation costs.



Real-Valued Representation

Genotype is sequence of real values that represent parameters

- Used when high-precision parameter optimization is required.
- For example, genetic encoding of wing curvature profile for shape optimization.



Evolvable wing made of deformable material with pressure tubes

<u>Genotype</u> = pressure values of 14 tubes

Althernatively, encoded values of variables of equations describe the profile.

Tree-based Representation

Genotype describes a tree with branching points and terminals

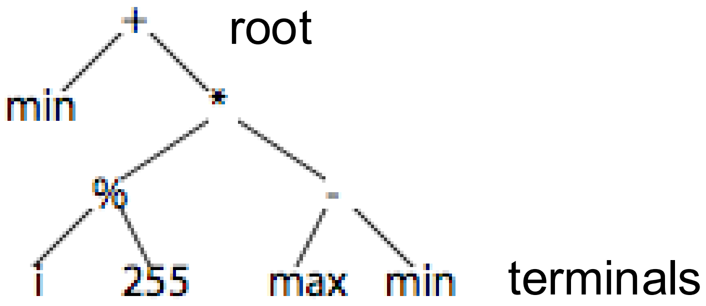
Suitable for encoding hierachical structures, for example, encoding computer programs.

Computer program is made of:

- Operators (Function set: multiplication, If-then, Log, etc.) and
- Operands (Terminal set: constants, variables, sensoor reading , etc.)

Expression r=min+(i/255)(max-min)

Nested list (+, min, (*, (/, i, 255), (-, max, min)))



The function set and the terminal set should satisfy

- **Closure:** all functions must accept all terminals in Terminal set and outputs of all functions in Function set (e.g., protected division %).
- **Sufficiency:** elements in Function and Terminal sets must be sufficient to generate program that solve the problem.

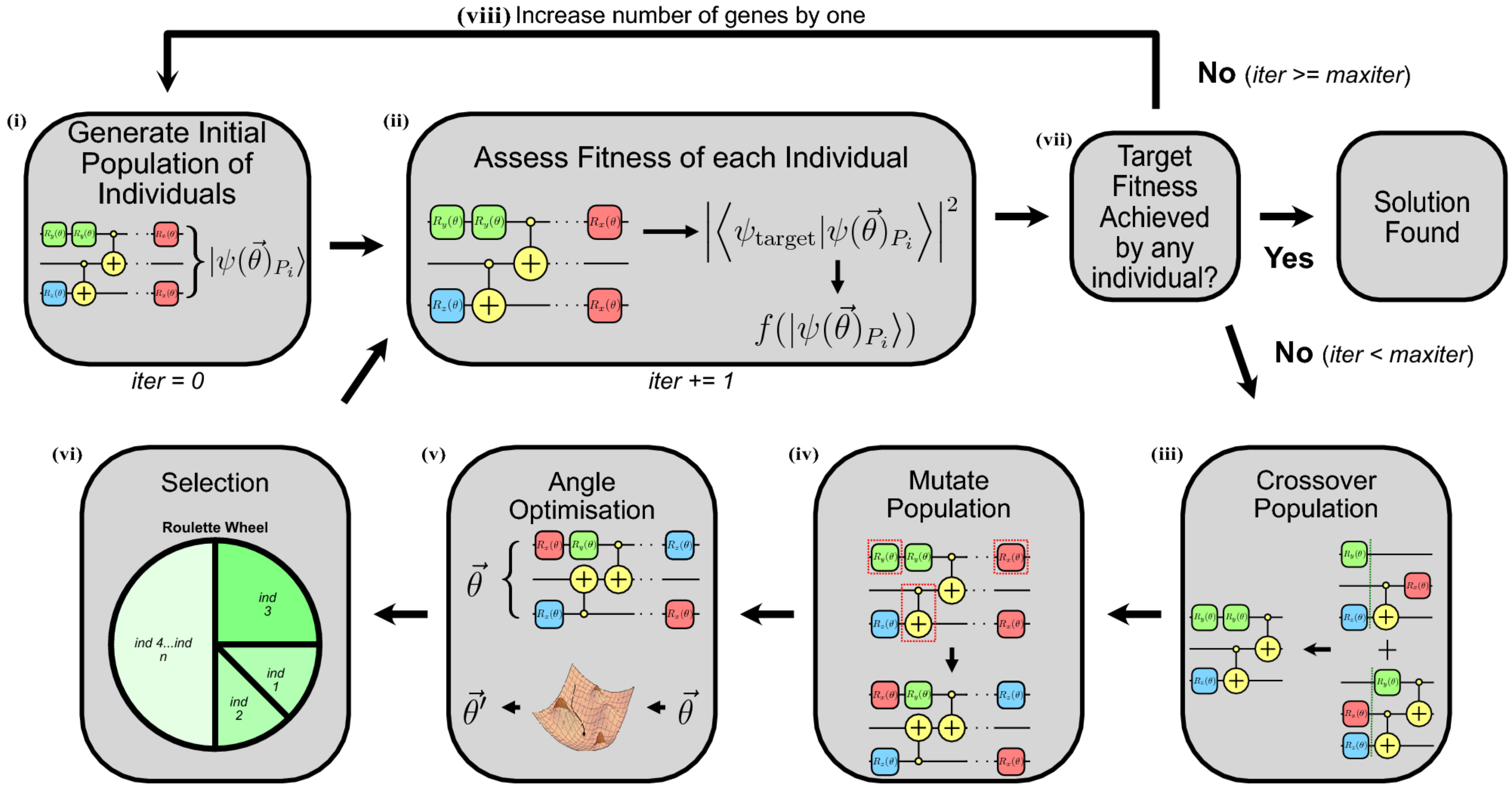
Generic Evolutionary Algorithm

Algorithm 8.1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
$\boxed{\text{Create and initialize an } n_x\text{-dimensional population, } \mathcal{C}(0)\text{, to consist of } n_s \text{ individuals;}}$
while *stopping condition(s) not true* **do**
  Evaluate the fitness, $f(\mathbf{x}_i(t))$, of each individual, $\mathbf{x}_i(t)$;
  Perform reproduction to create offspring;
  Select the new population, $\mathcal{C}(t+1)$;
  Advance to the new generation, i.e. $t = t + 1$;
end

Source: A.P. Engelbrecht, Computational Intelligence, An Introduction, Second Edition, 2007.

Generic Evolutionary Algorithm



Initial Population

Sufficiently large to cover problem space, but **sufficiently small** for evaluation costs (typical size: between 10s and 1000s individuals)

Uniform sample of search space:

- Binary strings: 0 or 1 with probability of 0.5
- Real-valued representations: uniform on a given interval if bounded phenotype (e.g., +2.0, -2.0); otherwise best guess or binary string with dynamic mapping resolution. (Schraudolph and Be
- Trees are built recursively starting from root: Root is a randomly chosen from function set; for every branch, randomly choose among all elements;

Uniform sample of search space:

- Possible loss of genetic diversity
- Possible unrecovable bias

Generic Evolutionary Algorithm

Algorithm 8.1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create and initialize an n_x-dimensional population, $\mathcal{C}(0)$, to consist of n_s individuals;
while *stopping condition(s) not true* **do**
  Evaluate the fitness, $f(\mathbf{x}_i(t))$, of each individual, $\mathbf{x}_i(t)$;
  Perform reproduction to create offspring;
  Select the new population, $\mathcal{C}(t+1)$;
  Advance to the new generation, i.e. $t = t + 1$;
end

Source: A.P. Engelbrecht, Computational Intelligence, An Introduction, Second Edition, 2007.

Fitness Function

Evaluates **performance** of phenotype with a numerical score,

- Choice of components: e.g., lift and drag of wing
- Combination of components e.g., (lift + 1/drag) or (liff – drag)
- Extensive test of each phenotype

Warning! You Get What You Evaluate!

Selective fitness: select phenotype by visual inspection

- Human observers rate the performance of evolving infidividual
- Used when aesthetic properties, e,g,,
  - Fugurative Art,
  - Architectural struck, and
  - Musics
- Can be combined with objective fitness function,

Generic Evolutionary Algorithm

Algorithm 8.1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create and initialize an n_x-dimensional population, $\mathcal{C}(0)$, to consist of n_s individuals;
while *stopping condition(s) not true* **do**
    Evaluate the fitness, $f(\mathbf{x}_i(t))$, of each individual, $\mathbf{x}_i(t)$;
    Perform reproduction to create offspring;
    Select the new population, $\mathcal{C}(t+1)$;
    Advance to the new generation, i.e. $t = t + 1$;
end

Source: A.P. Engelbrecht, Computational Intelligence, An Introduction, Second Edition, 2007.

Selection

- A method to make sure that better individuals make comparatively more offspring.
- Used in artificial evolution and breeding.

- Selection pressure is inversely proportional to the number of selected individuals.

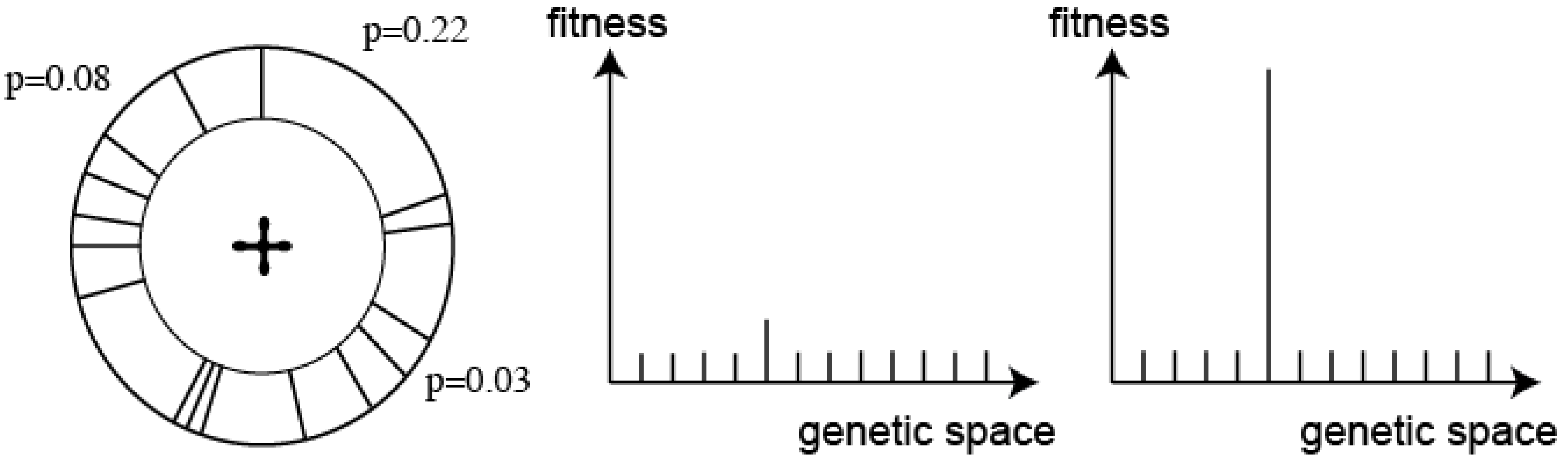
- **High selection pressure** = **rapid fitness increment**
  = **rapid loss of diversity and premature covergence**

- Make sure that less performing individuals can also reproduce to some extent.

Proportionate Selection

The probability that an individual makes an offspring is proportional to how good its fitness is with respect to the population fitness: $\boldsymbol{p(i) = f(i)/\Sigma f(i)}$

Also known as **Roulette Wheel Selection**

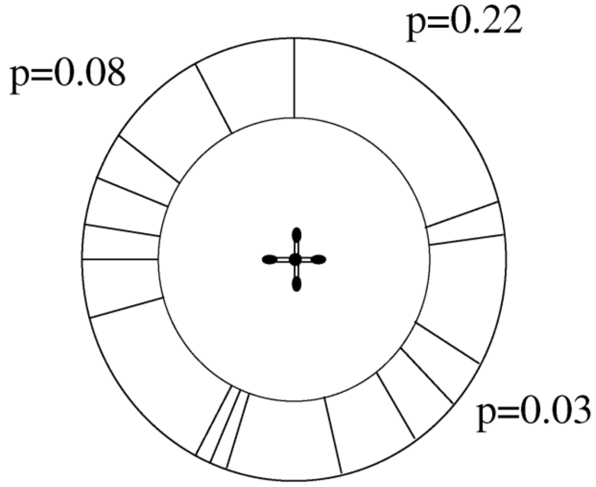


Problems:
Uniform fitness value = random search
Few high-fitness individuals = high selection pressure

Rank-based Selection

- Individuals are sorted on their fitness value from best to worse. The place in this sorted list is called **the rank r**.
- Instead of using the fitness value of an individual, the rank is used to select individuals: $\boldsymbol{p(i) = 1 - r(i)/\Sigma \mathrm{r}(\mathrm{i})}$
- Use roulette wheel

individual	fitness	rank
A	5	5
B	7	3
C	8	2
D	2	8
E	3	7
F	9	1
G	7	4
H	4	6



No matter how small the difference between any two individuals is, the better one of the two will always have a higher probability of making offspring

Truncated Rank-based Selection

- Only the best **x** individuals are allowed to make offspring and each of them makes the same number of offspring: N/x, where **N** is the population size.
- E.g., in population of 100 individuals, make 5 copies of 20 best individuals

Provided that x is not too small, this method ensures that also individuals that have obtained relatively low fitness scores, but still higher than the worst ones, are given the same number of offspring as the best individuals.

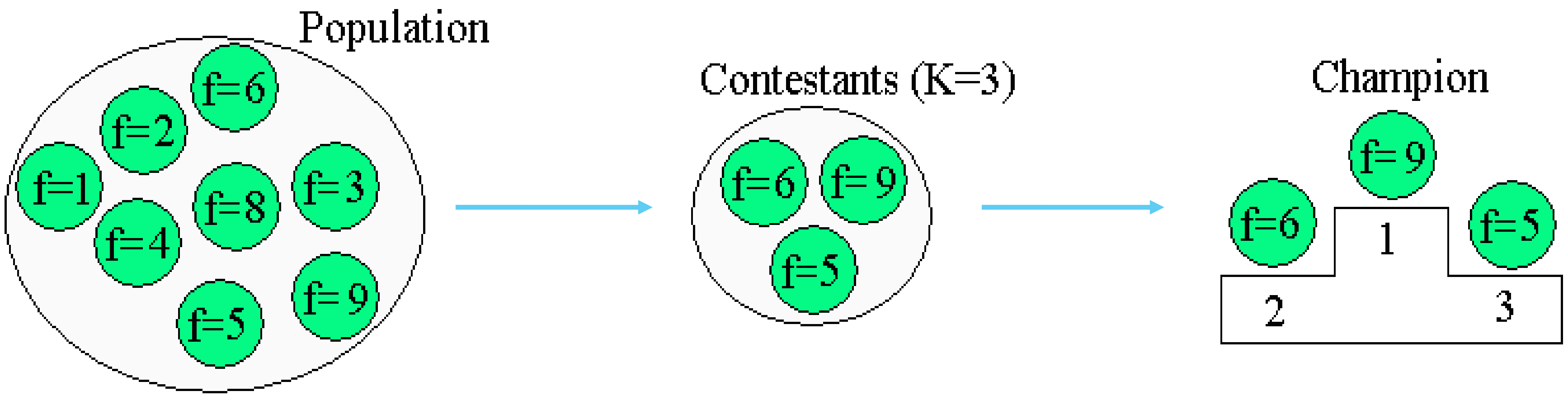
Individual	Fitness	Rank	List
A	5	5	F
B	7	3	C
C	8	2	B
D	2	8	G
E	3	7	A
F	9	1	H
G	7	4	E
H	4	6	D

Quite useful when their fitness scores may not reflect their true fitness

Tournament Selection

For every offspring to be generated:

- Pick randomly ***k*** individuals from the population
- Choose the individual with the highest fitness and make a copy
- Put all individuals back in the population



K is the tournament size (larger size = larger selection pressure)

Tournament selection achieves a good compromise in maintaining both selection pressure and genetic diversity in the population.

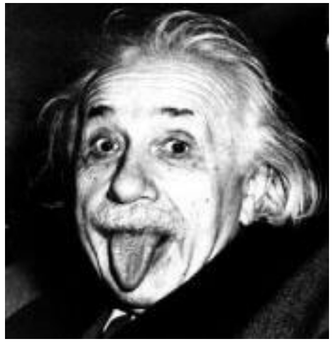
Hall of Fame Selection

- For each generation, the best individual is selected to be inserted into the hall of fame.
- The hall of fame will therefore contain an archive of the best individuals found from the first generation.
- The hall of fame can be used as a parent pool for the crossover operator.

Replacement

- **Generational replacement:** old population is entirely replaced by offsprings (most frequent method).

- **Elitism:** maintain ***n*** best individuals from previous generation to prevent loss of best individuals by effects of mutations or sub-optimal fitness evaluation.

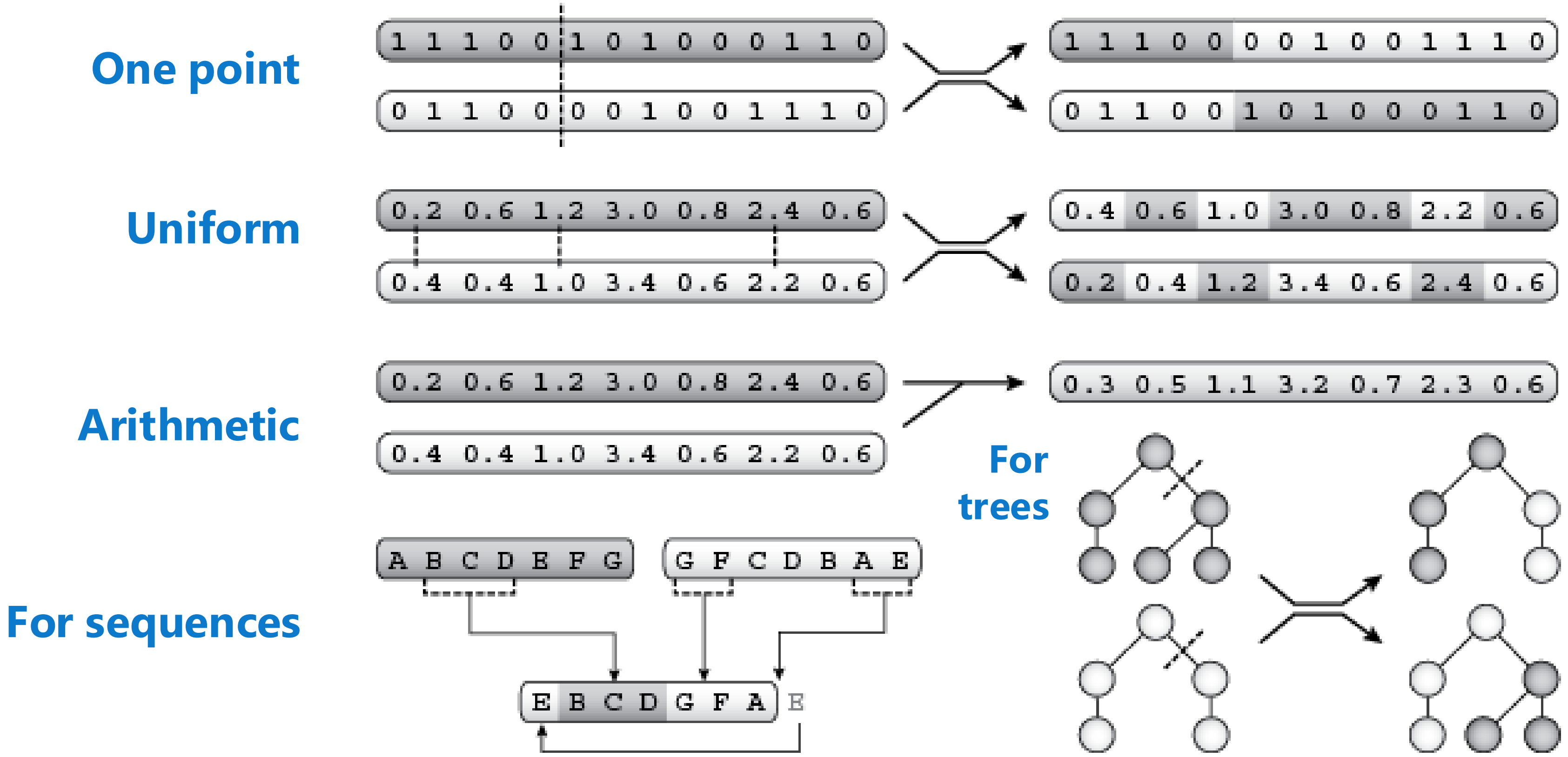


- **Generational rollover:** inserts offspring at the place of worst individuals.

Crossover

Emulates recombination of genetic material from two parents.

Exploitation of synergy of sub-solutions (building blocks) from parents. Applied to randomly paired offspring with probability $p_c(pair)$.



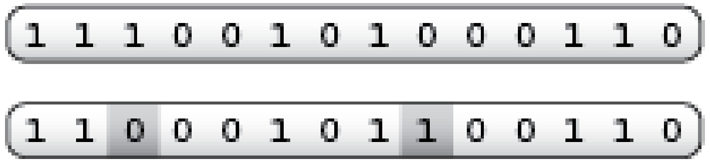
Mutation

Emulates genetic mutations

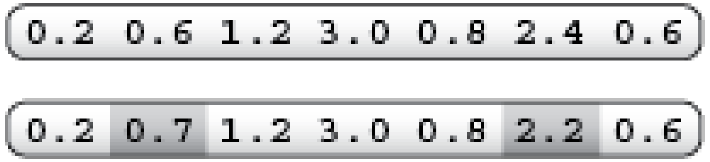
Exploration of variation of existing solutions.

Applied to each character in the genotype with probability $p_m(char)$

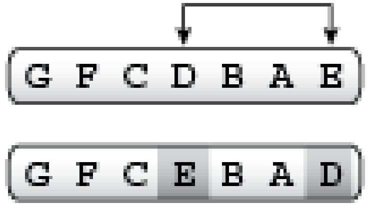
Binary Genotypes



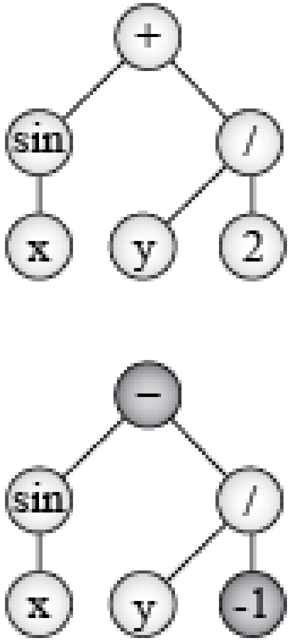
Real-valued Genotypes



Sequence Genotypes



For Tress



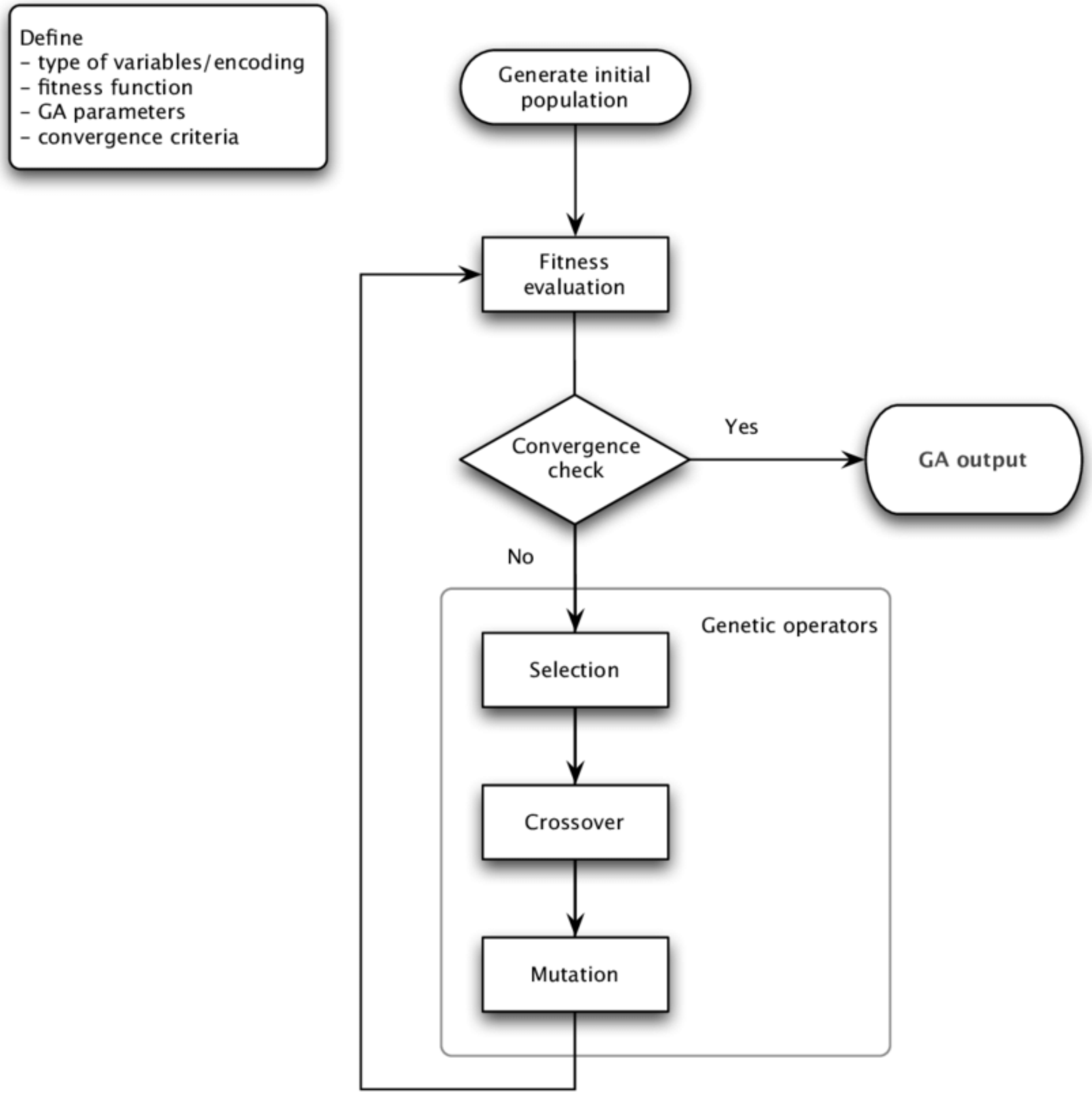
Generic Evolutionary Algorithm

Algorithm 8.1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create and initialize an n_x-dimensional population, $\mathcal{C}(0)$, to consist of n_s individuals;
while *stopping condition(s) not true* **do**
  Evaluate the fitness, $f(\mathbf{x}_i(t))$, of each individual, $\mathbf{x}_i(t)$;
  Perform reproduction to create offspring;
  Select the new population, $\mathcal{C}(t+1)$;
  Advance to the new generation, i.e. $t = t + 1$;
end

- Terminate when no improvement is observed over a number of consecutive generations.
- Terminate when there is no change in the population.
- Terminate when an acceptable solution has been found.

Source: A.P. Engelbrecht, Computational Intelligence, An Introduction, Second Edition, 2007.



Genetic Algorithm

Binary genetic representation

Design Components:

- Number of binary bits
- Population size
- Population generating mechanism
- Fitness function
- Selection mechanism
- Genetic operators
  - Crossover
  - Mutation
- Genetic operator rate
- Stopping Criteria

GA in Function Optimization

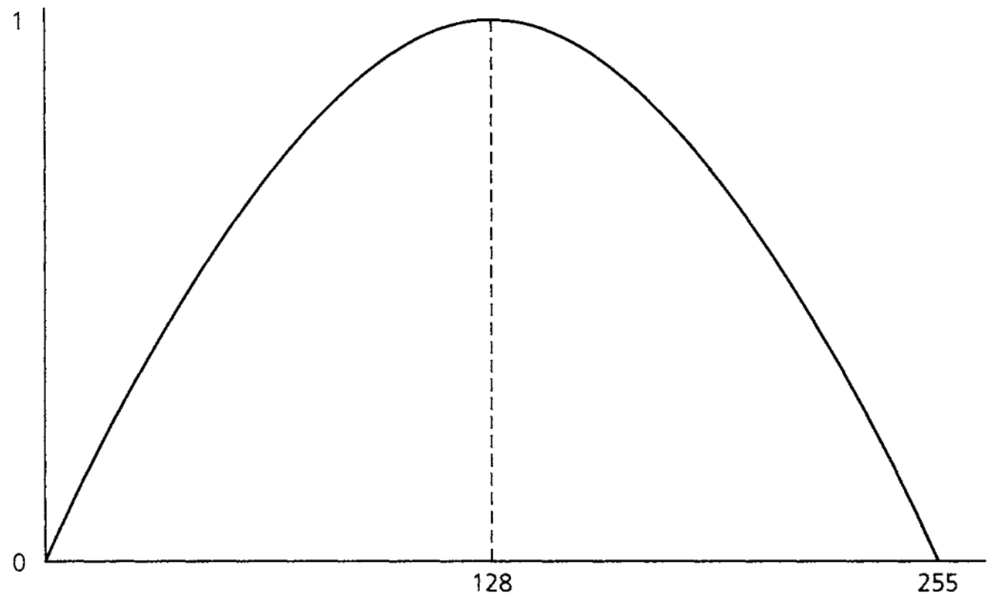


Figure 3.1 Function to be optimized in example problem.

$$f(x) = \sin\left(\frac{\pi x}{256}\right)$$

$$0 \leq x \leq 255$$

Where x is restricted to integers

Each individual can be represented as **8-bit binary string**

Individual in the population can be varied from

0 0 0 0 0 0 0 0 to **1 1 1 1 1 1 1 1**

Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

Population Initialization

Randomise population with probability of 0.5

Population Size = 8

Individuals	x	$f(x)$	f_{norm}	cumulative f_{norm}
1 0 1 1 1 1 0 1	189	0.733	0.144	0.144
1 1 0 1 1 0 0 0	216	0.471	0.093	0.237
0 1 1 0 0 0 1 1	99	0.937	0.184	0.421
1 1 1 0 1 1 0 0	236	0.243	0.048	0.469
1 0 1 0 1 1 1 0	174	0.845	0.166	0.635
0 1 0 0 1 0 1 0	74	0.788	0.155	0.790
0 0 1 0 0 0 1 1	35	0.416	0.082	0.872
0 0 1 1 0 1 0 1	53	0.650	0.128	1.000

$\Sigma f(x) = 5.083$

Fitness Evaluation

Fitness value for each individual

Function space is identical to fitness space, so the fitness function is the function to be optimized.

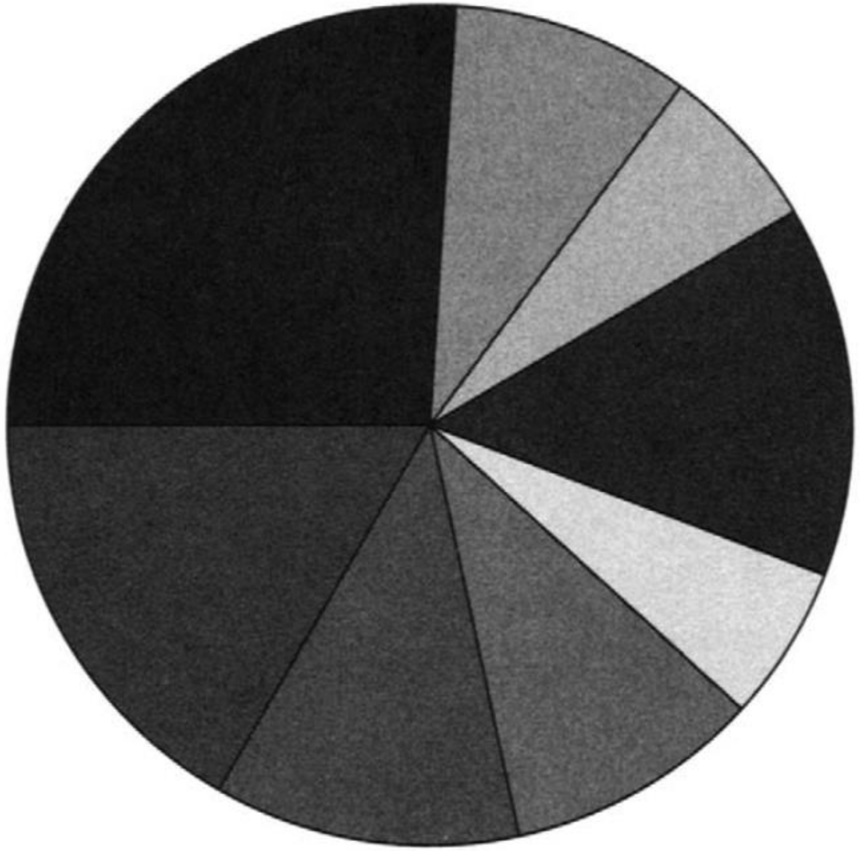
$$f(x) = \sin\left(\frac{\pi x}{256}\right)$$

Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

Reproduction Process

Roulette wheel selection

The size of the roulette wheel wedge for each population member, which reflects the probability of the individual being selected, is proportional to its normalized fitness value.



The probability that an individual is selected is thus proportional to that individual's fitness value.

Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

Individuals	x	$f(x)$	f_{norm}	cumulative f_{norm}
1 0 1 1 1 1 0 1	189	0.733	0.144	0.144
1 1 0 1 1 0 0 0	216	0.471	0.093	0.237
0 1 1 0 0 0 1 1	99	0.937	0.184	0.421
1 1 1 0 1 1 0 0	236	0.243	0.048	0.469
1 0 1 0 1 1 1 0	174	0.845	0.166	0.635
0 1 0 0 1 0 1 0	74	0.788	0.155	0.790
0 0 1 0 0 0 1 1	35	0.416	0.082	0.872
0 0 1 1 0 1 0 1	53	0.650	0.128	1.000
		$\Sigma f(x) = 5.083$		

The roulette wheel is "spun" to generate **eight random number** between 0 and 1, which are:

0.293, **0.971**, **0.160**, **0.469**, **0.664**, **0.568**, **0.371**, and **0.109**,

The initial population member numbers

3, **8**, **2**, **5**, **6**, **5**, **3**, and **1**

Are chosen to make up the new population after reproduction

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

Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

Crossover Operation

Let the **crossover rate** to be at the probability of **0.75**

Crossover is the process of exchanging portions of the strings of two "parent" individuals.

Two crossover points are selected at random

(a)	Individuals (b)	x (c)	$f(x)$ (d)
1 2			
0 1 1 | **0 0 0** | 1 1	0 1 1 1 0 1 1 1	119	0.994
0 0 1 | **1 0 1** | 0 1	0 0 1 0 0 0 0 1	33	0.394
1 2			
1 | **1 0 1 1** | 0 0 0	1 0 1 0 1 0 0 0	168	0.882
1 | **0 1 0 1** | 1 1 0	1 1 0 1 1 1 1 0	222	0.405
2 1			
0 1 | 0 0 1 0 1 | **0**	1 0 0 0 1 0 1 0	138	0.992
1 0 | 1 0 1 1 1 | **0**	0 1 1 0 1 1 1 0	110	0.976
0 1 1 0 0 0 1 1	0 1 1 0 0 0 1 1	99	0.937
1 0 1 1 1 1 0 1	1 0 1 1 1 1 0 1	189	0.733

Population before crossover showing crossover points (a); after crossover (b); and values of x (c) and $f(x)$ (d) after crossover.

Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

Mutation Operation

Crossover in genetic algorithm is plainly
Consists of flipping bits at random.

0 1 1 0 1 1 1 0

1 1 1 0 1 1 1 0

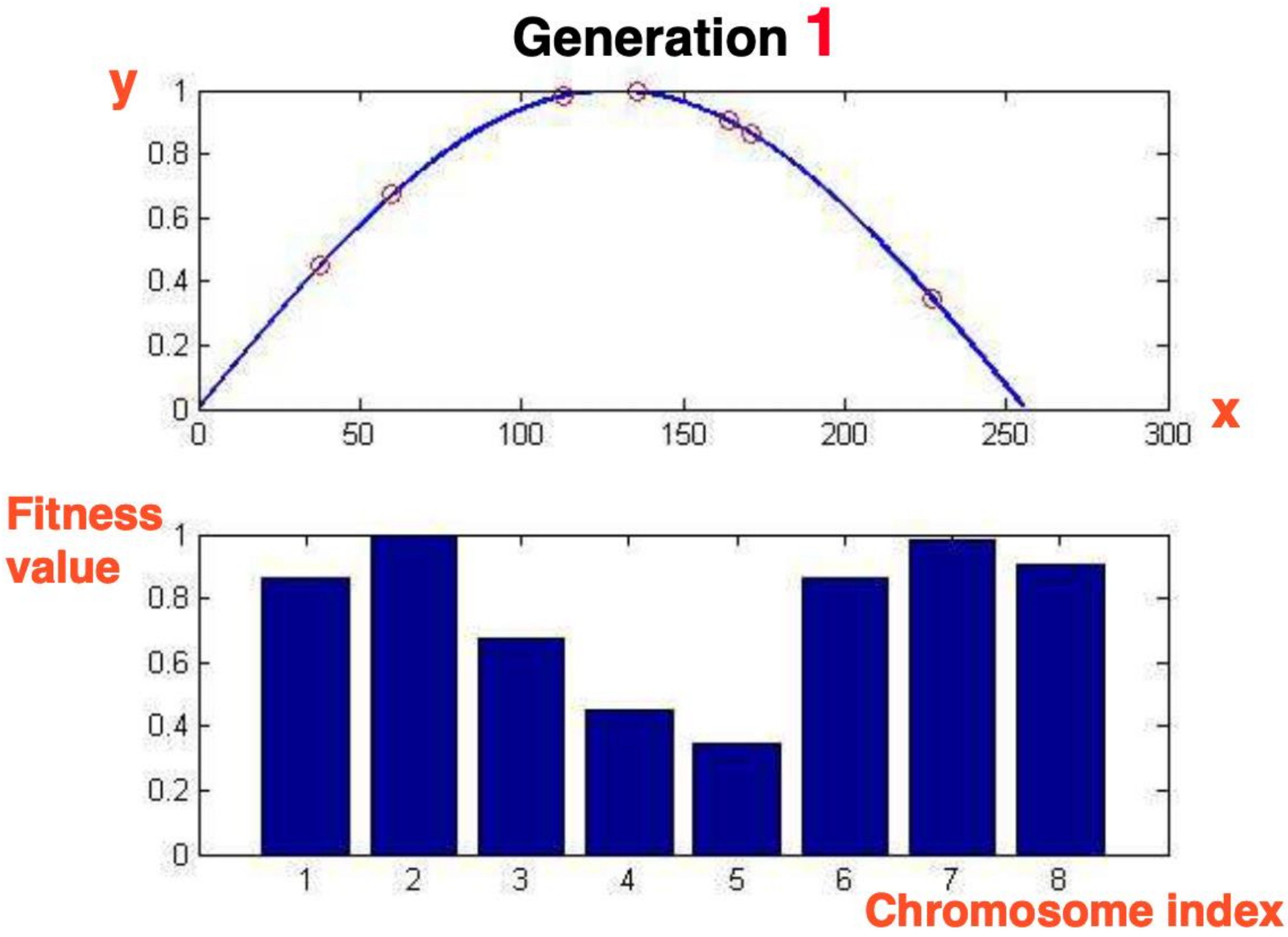
The **probability of mutation** can vary widely according to the application and the preference of the researcher

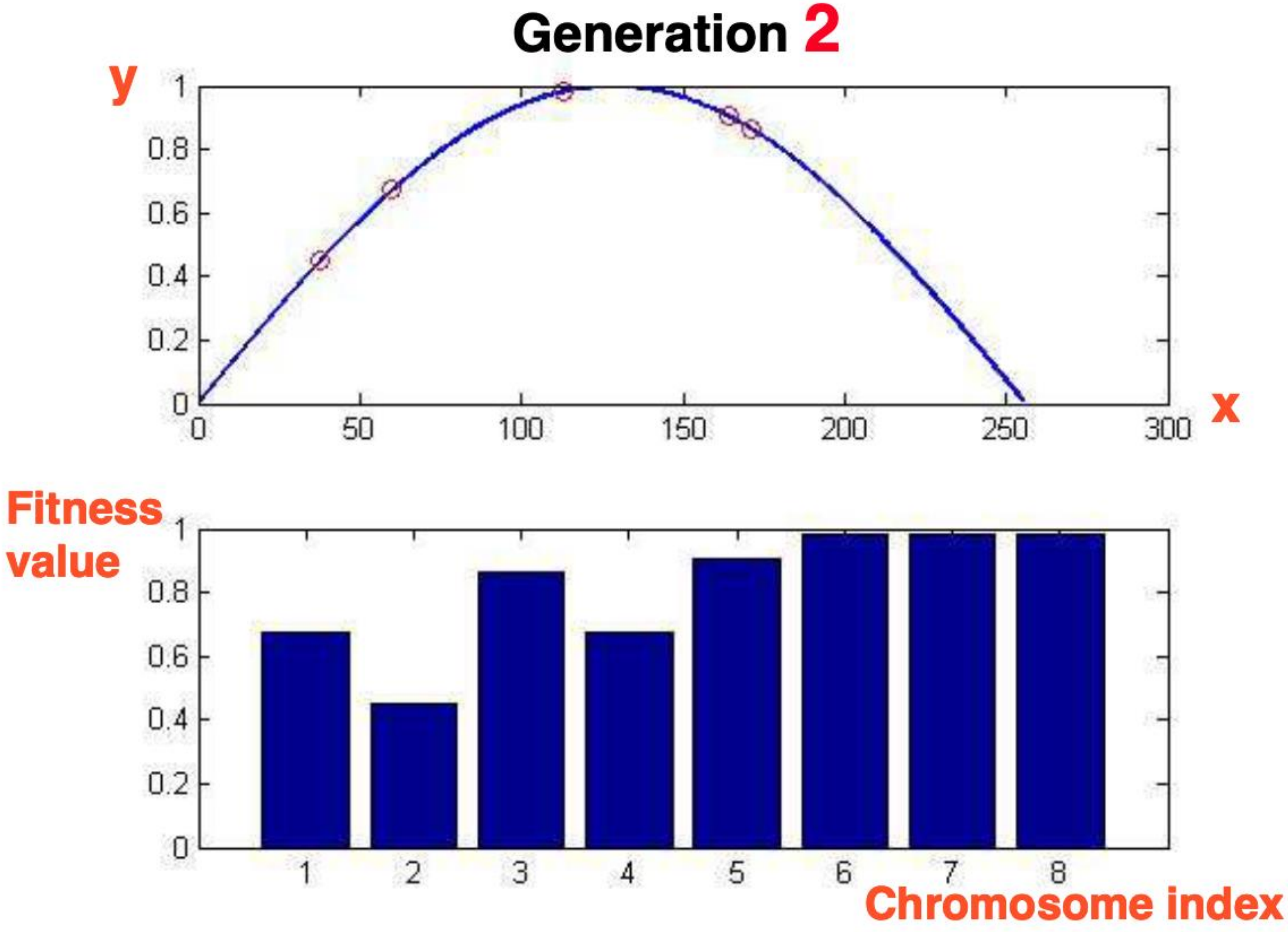
One fixed value is used for each generation and often maintained for an entire run.

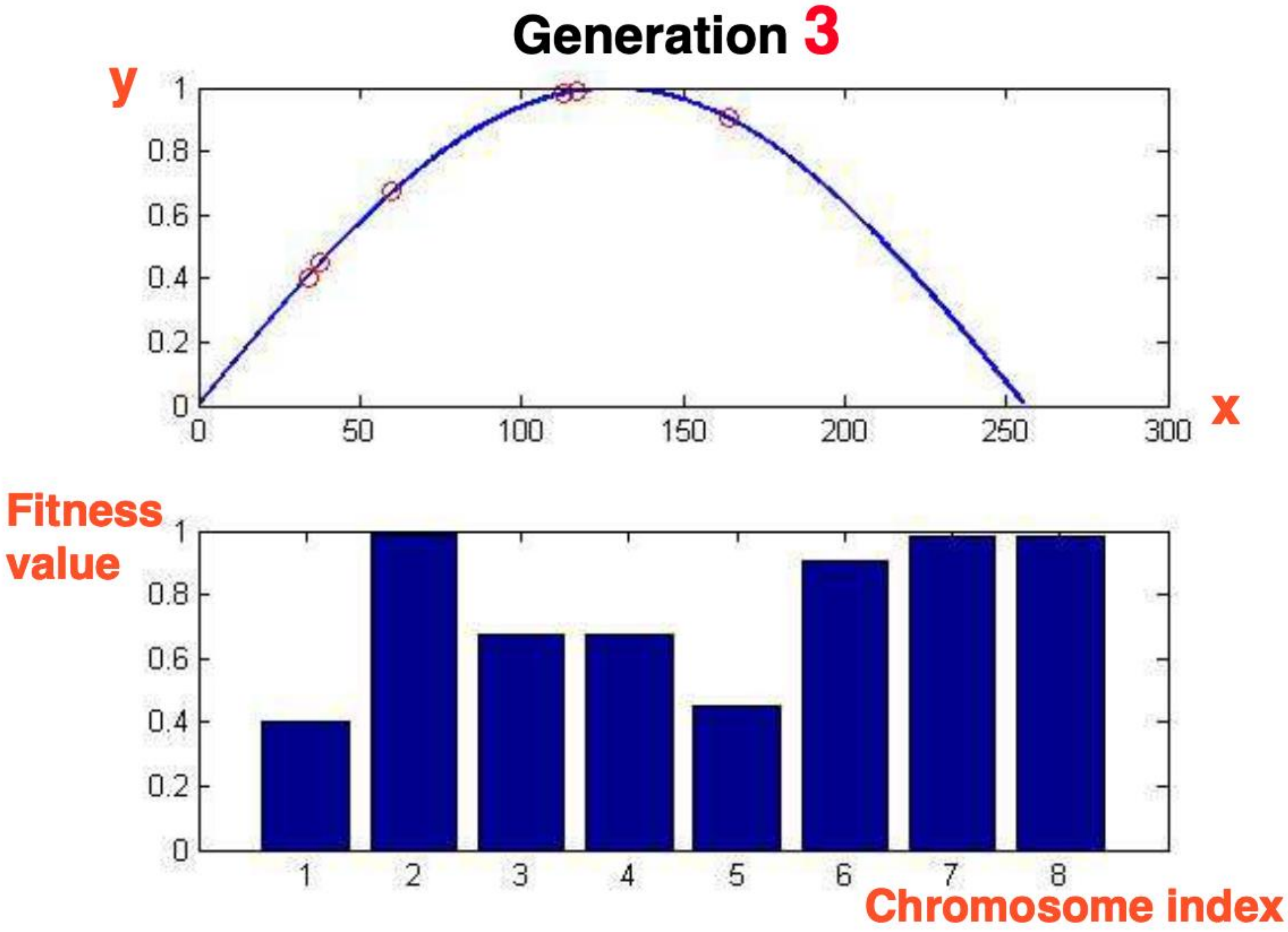
Going through the entire GA procedure one time is said to produce a new generation. Therefore, the new population represents the first generation of the initial randomized population.

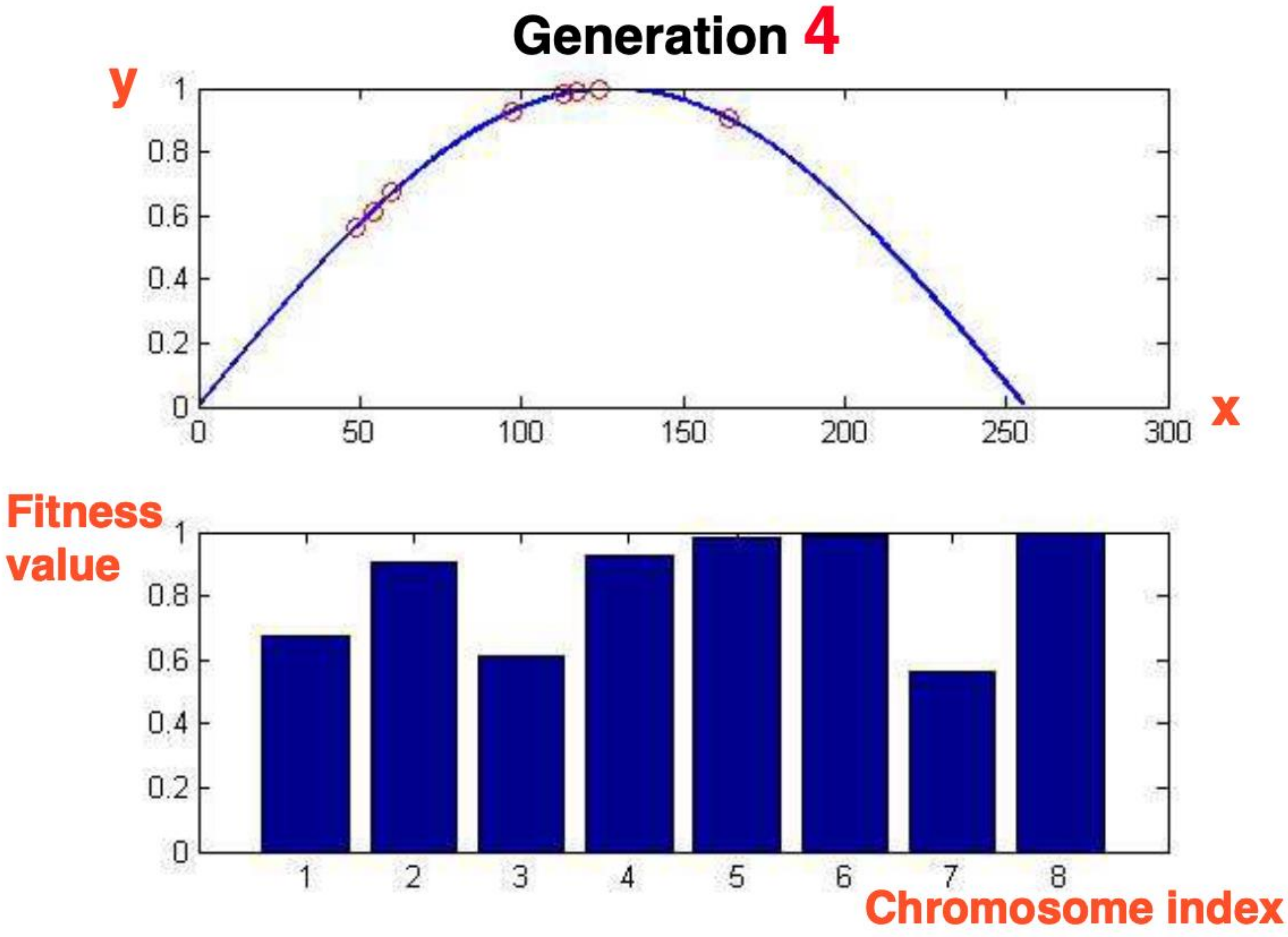
Source: R. C. Eberhart and Y. Shi, Computational Intellignece: concepts to implementations, 2007

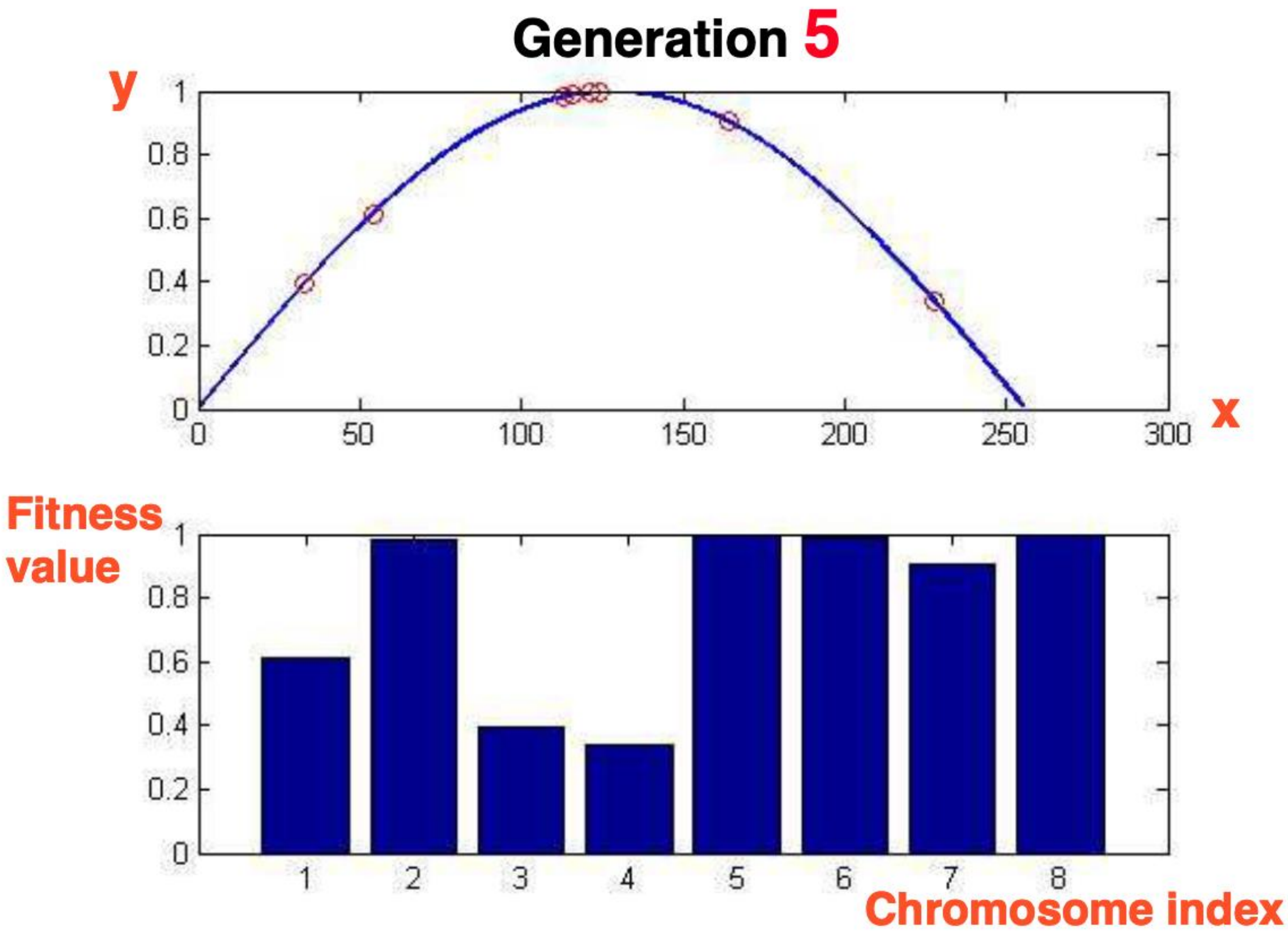
Demonstration on Example of Using Genetic Algorithm

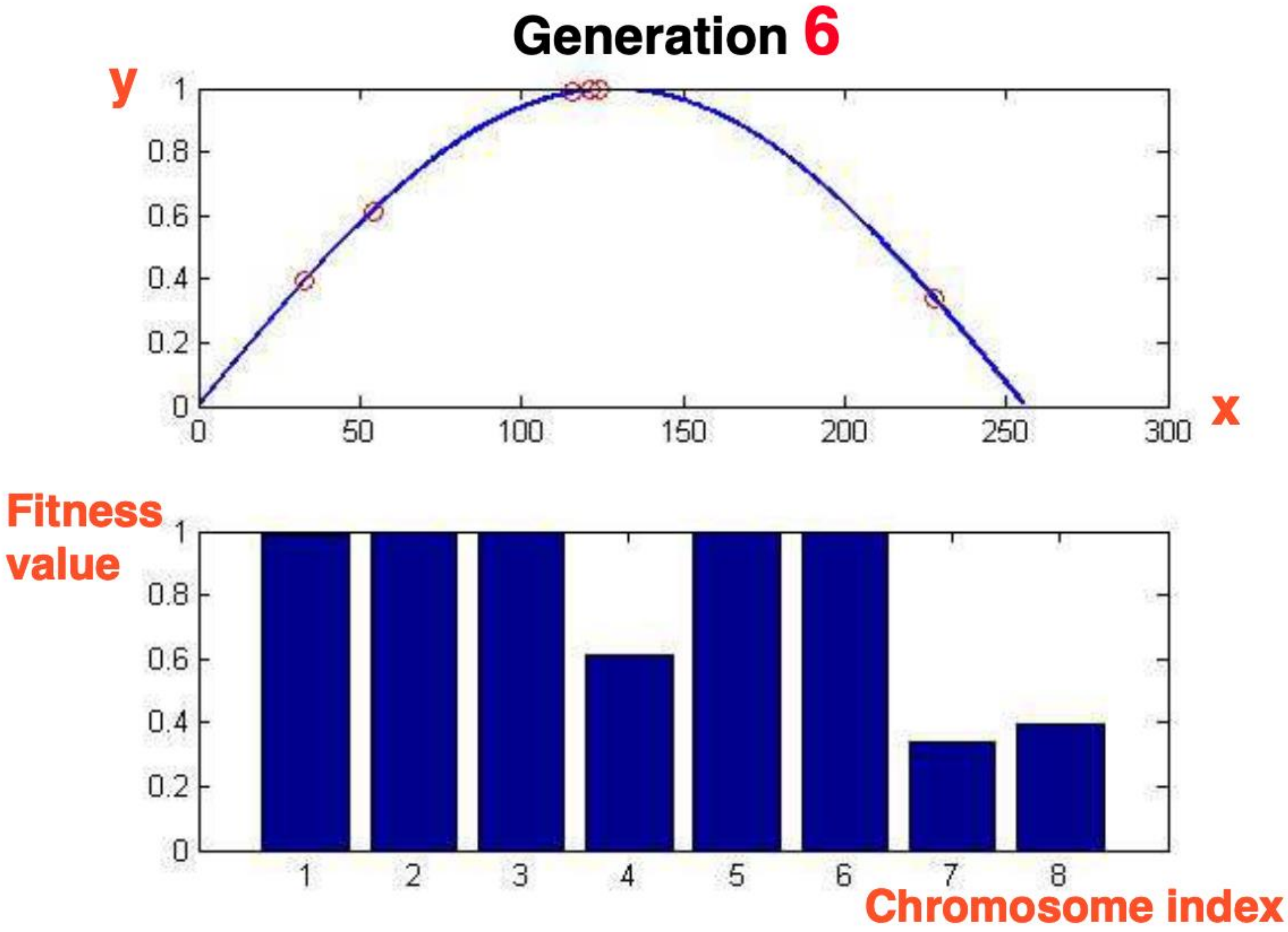

Generation 1
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index

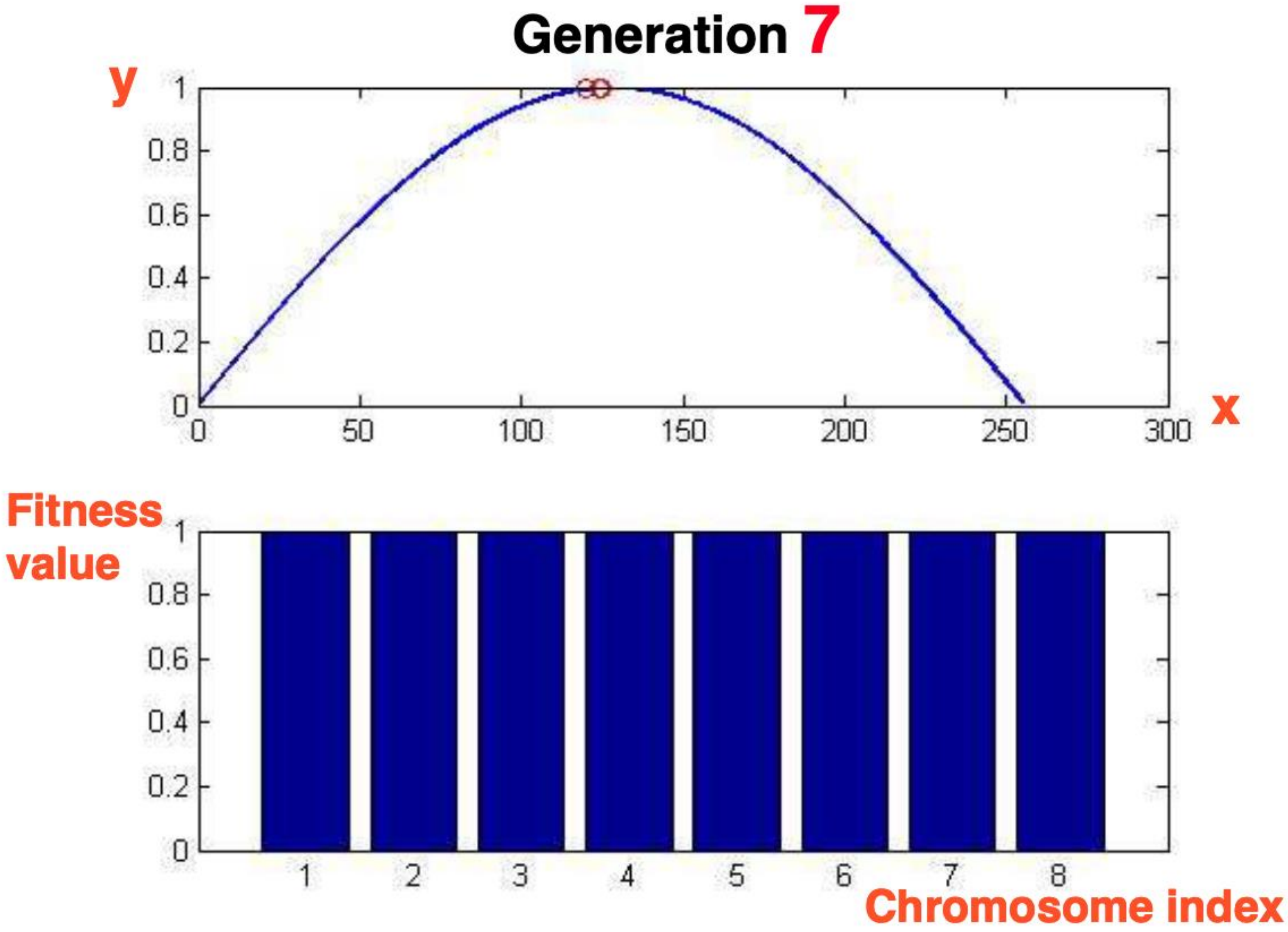

Generation 2
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index

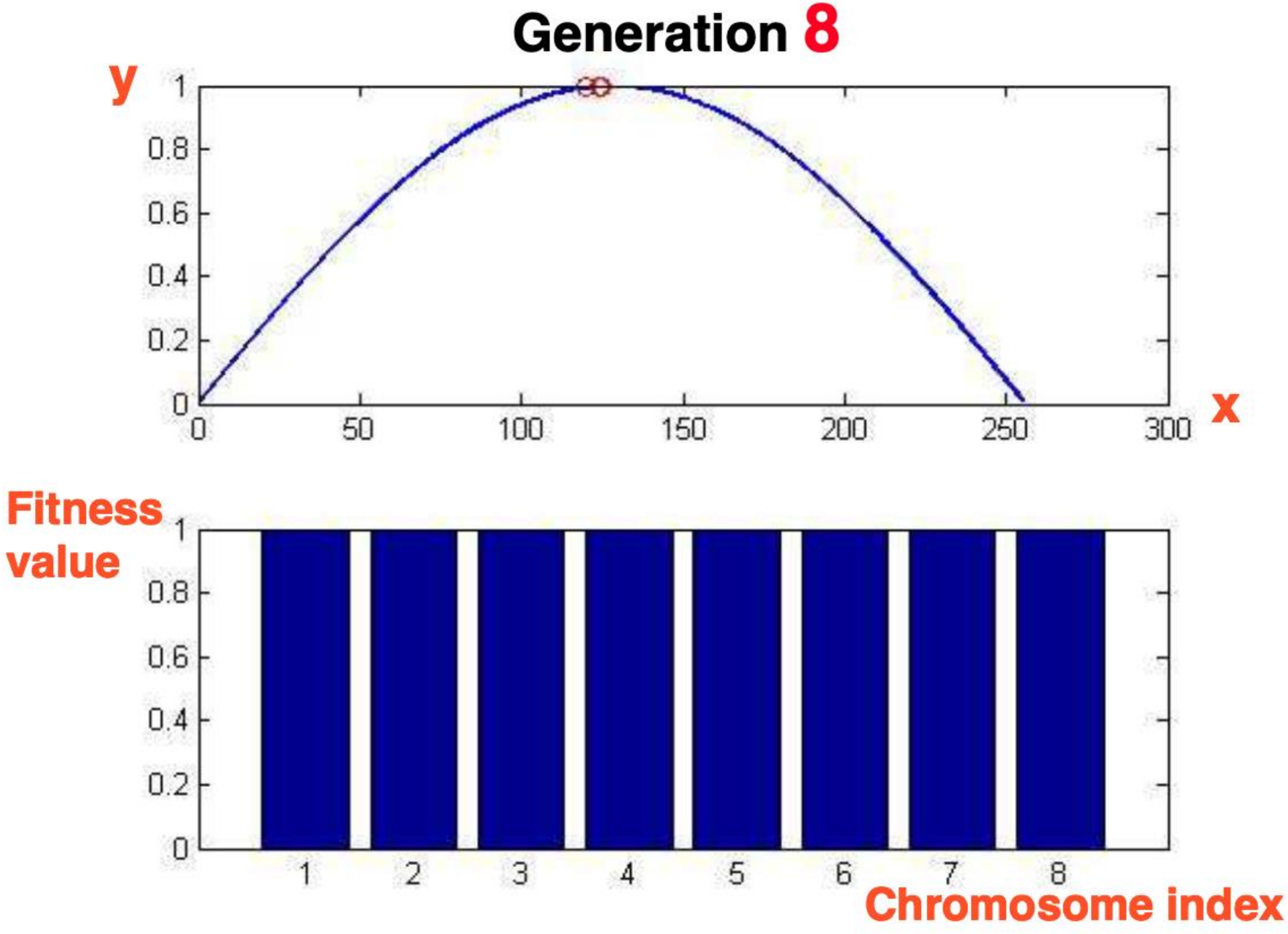

Generation 3
y
1
0.8
0.6
0.4
0.2
0
0 50 100 150 200 250 300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1 2 3 4 5 6 7 8
Chromosome index

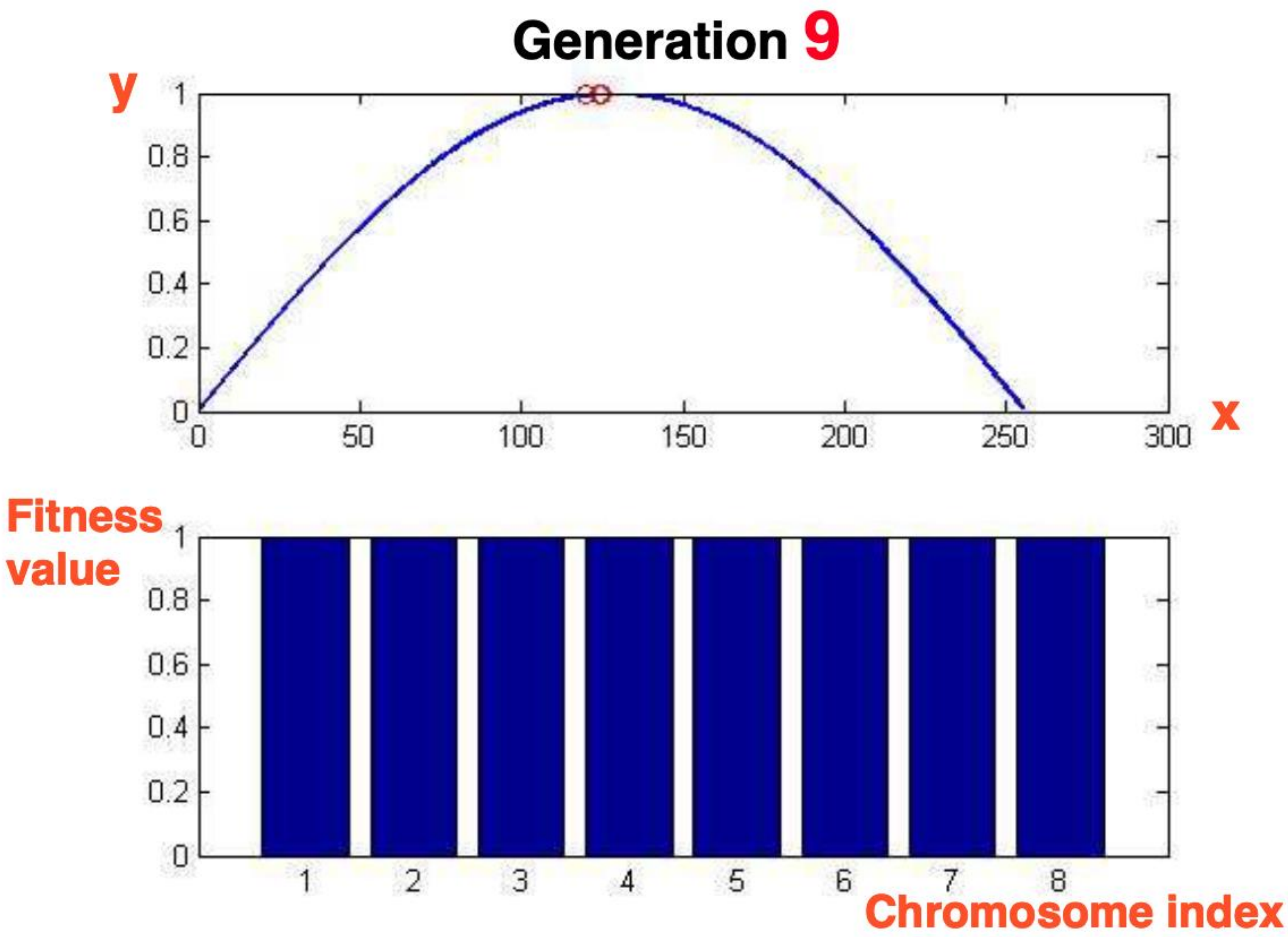

Generation 4
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index

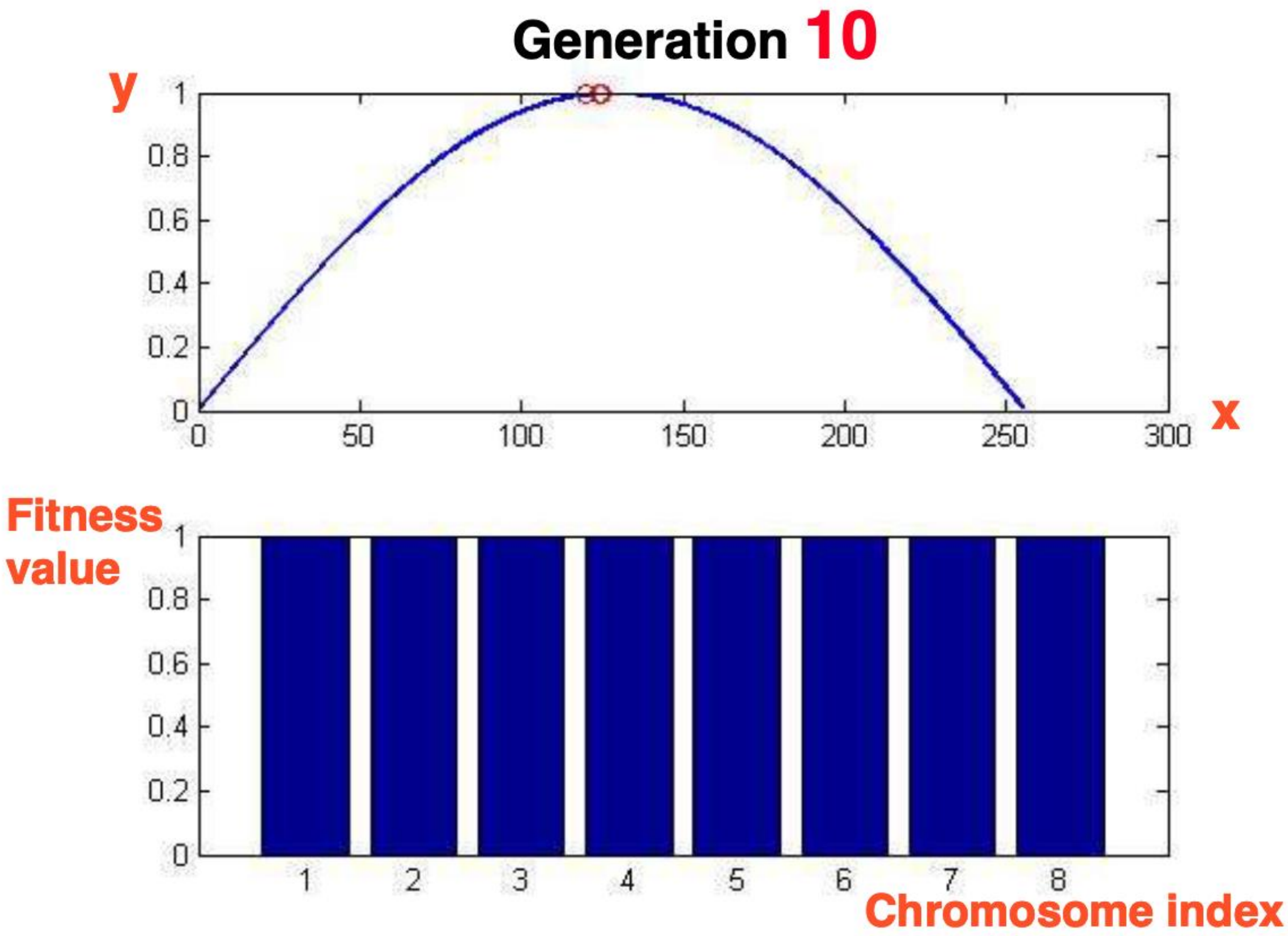

Generation 5
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index

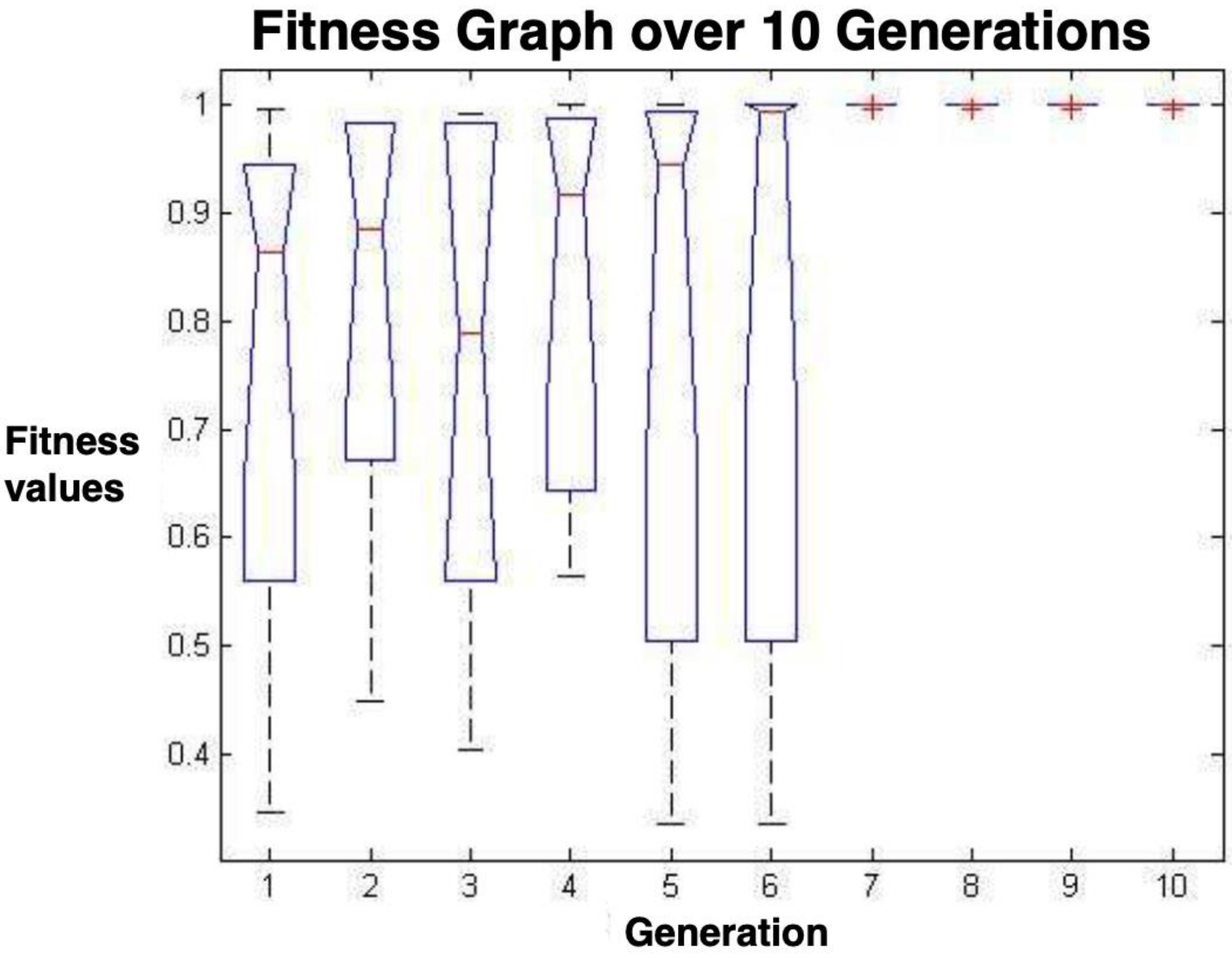

Generation 6
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index


Generation 7
y
x
0
0.2
0.4
0.6
0.8
1
0
50
100
150
200
250
300
Fitness value
Chromosome index
1
2
3
4
5
6
7
8


Generation 8
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index


Generation 9
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index


Generation 10
y
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
x
Fitness value
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
Chromosome index


Fitness Graph over 10 Generations
Fitness values
1
0.9
0.8
0.7
0.6
0.5
0.4
1
2
3
4
5
6
7
8
9
10
Generation

Major Evolutionary Algorithms

- **Genetic Algorithms (GA)** – *Holland, 1975*

  Binary Genotypes, Crossover and Mutation

- **Genetic Programming (GP)** – *Koza, 1992*

  Tree-based Genotypes, Crossover and Mutation

- **Evolutionary Programming (EP)** - *Fogel et al., 1966*

  Real-valued Genotypes, Mutations, Tournaments, Gradual Population Replacement

- **Evolutionary Strategies (ES)** – *Rechenberg, 1973*

  As EP + Mutation range ecoded in genotype of individual

- **Island GA** – *Whitley et al., 1998*

  Parallel evolving populations with rare migration of individuals

- **Differential Evolution** – *Storn and Price, 1995*

  Stochastic, Population-based Search Strategy for Continuous-valued Landscapes.

🥳

End of the Lecture

Please don’t hesitate to raise your hand and ask questions if you're curious about anything!