Lecture 2: Canonical Genetic Algorithms

Suggested reading: D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison Wesley Publishing Company, January 1989

What Are Genetic Algorithms?

- Genetic algorithms are optimization algorithm inspired from natural selection and genetics
- A candidate solution is referred to as an ***individual***
- Process
  - Parent individuals generate offspring individuals
  - The resultant offspring are evaluated for their **fitness**
  - The fittest offspring individuals survive and become parents
  - The process is repeated

History of Genetic Algorithms

- In 1960's
  - Rechenberg: "***evolution strategies***"
    - Optimization method for real-valued parameters
  - Fogel, Owens, and Walsh: "***evolutionary programming***"
    - Real-valued parameters evolve using random mutation

- In 1970's
  - John Holland and his colleagues at University of Michigan developed "***genetic algorithms*** (GA)"
  - Holland's1975 book "*Adaptation in Natural and Artificial Systems*" is the beginning of the GA
  - Holland introduced "schemas," the framework of most theoretical analysis of GAs.

- In 1990's
  - John Koza: "***genetic programming***" used genetic algorithms to evolve programs for solving certain tasks

- It is generally accepted to call these techniques as *evolutionary computation*

- Strong interaction among the different evolutionary computation methods makes it hard to make strict distinction among GAs, evolution strategies, evolutionary programming and other evolutionary techniques

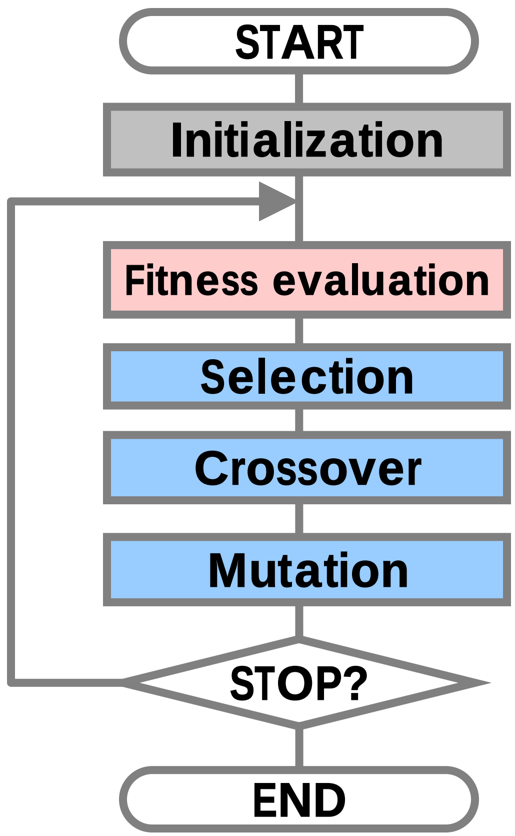
Differences Between GAs and Traditional Methods

- GAs operate on encodings of the parameters values, not necessarily the actual parameter values
- GAs operate on a population of solutions, not a single solution
- GAs only use the fitness values based on the objective functions
- GAs use probabilistic computations, not deterministic computations
- GAs are efficient in handling problems with a discrete or mixed search spaces

The Canonical GA

Canonical GA

- The **canonical genetic algorithm** refers to the GA proposed by John Holland in 1965



Gene Representation

- Parameter values are encoded into **binary** strings of **fixed** and **finite** length
  - Gene: each bit of the binary string {0, 1}
  - Chromosome: a binary string
  - Individual: a set of one or multiple chromosomes, a prospective solution to the given problem
  - Population: a group of individuals
- Longer string lengths
  - Improve resolution
  - Requires more computation time

Binary Representation

- Suppose we wish to maximize $f(x)$
- Where $x \in \Omega; \quad \Omega = \lfloor x_{min,} x_{max} \rfloor$
- Binary representation $x_{binary} \in [b_l \ b_{l-1} \cdots b_2 \ b_1]$
- We map $\lfloor x_{min,} x_{max} \rfloor$ to $\lfloor 0, 2^l - 1 \rfloor$
- Thus $x = x_{min} + \dfrac{x_{max} - x_{min}}{2^l - 1} \sum_{i=1}^{l} b_i \, 2^{i-1}$

Example

- Let l = 5, Ω = [-5, 20]
- Then

$$x_{\text{binary}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow x = -5$$

$$x_{\text{binary}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \Rightarrow x = 20$$

$$x_{\text{binary}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \end{bmatrix} \Rightarrow x = -5 + (2^4 + 2^1 + 2^0)\frac{20-(-5)}{2^5 - 1} = 10.3226$$

Fitness Evaluation

- Each individual ***x*** is assigned with a fitness value *f*(***x***) as the measure of performance
- It is assumed that the fitness value is positive and the better the individual as a solution, the fitness value is more positive
- The objective function can be the fitness function itself if it is properly defined

Example 1

- Consider the problem

$$\max g(x) = -x^2 + 4x, \quad x \in [1, 5]$$

- A fitness function

$$f(x) = -g(x)+100 = -x^2 + 4x + 100$$

Example 2

- Consider the problem

$$\min g(x) = x^2, \quad x \in [-10, 10]$$

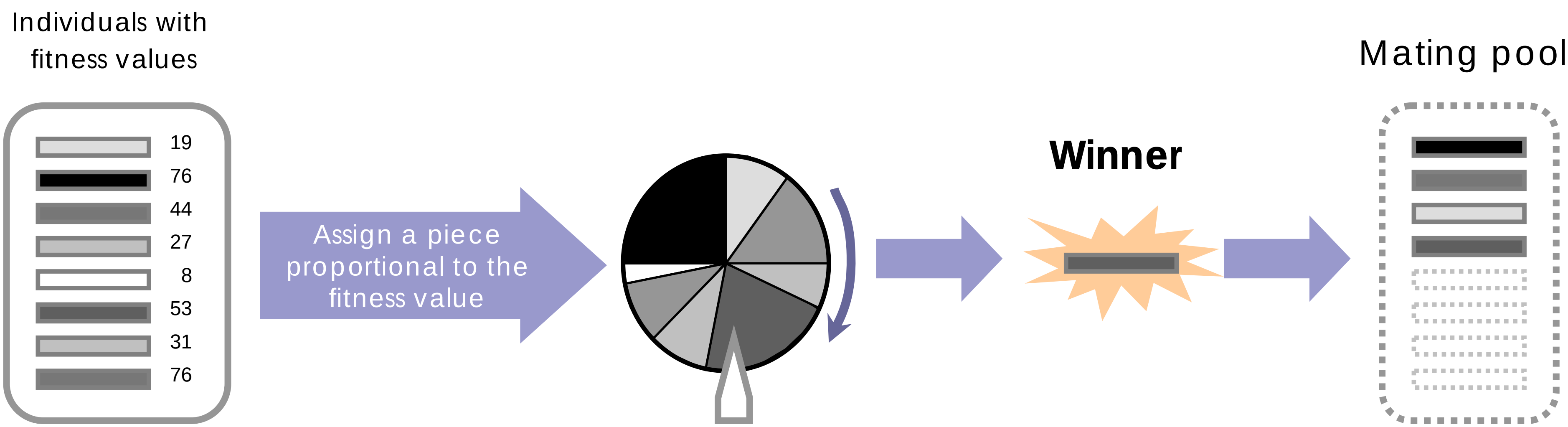
- A fitness function

$$f(x) = 1/(g(x) + 0.1) = 1/(x^2 + 0.1)$$

Selection

- Chooses individuals from the current population to constitute a mating pool for reproduction
- Fitness proportional selection methods
  - Each individual ***x*** is selected and copied in the mating pool with the probability proportional to fitness ($f(x) / \Sigma f(x)$)

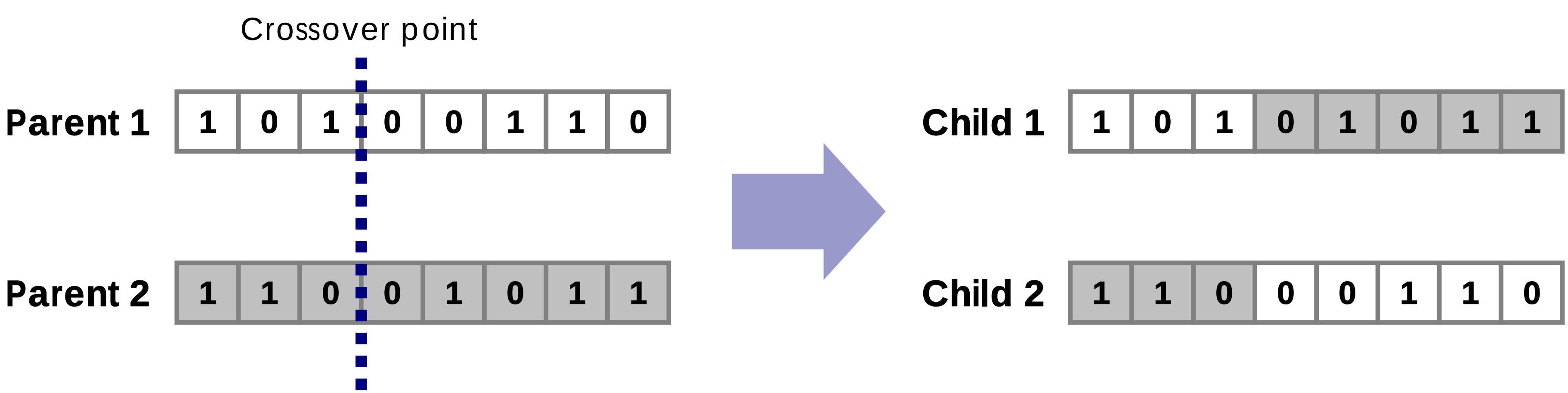
Roulette Wheel Selection



Crossover

- **Single-point crossover** is assumed
- Two parent individuals are selected from mating pool
- Crossover operation is executed with the probability p_c
- Crossover point is randomly chosen and the strings are swapped with respect the crossover point between the two parents

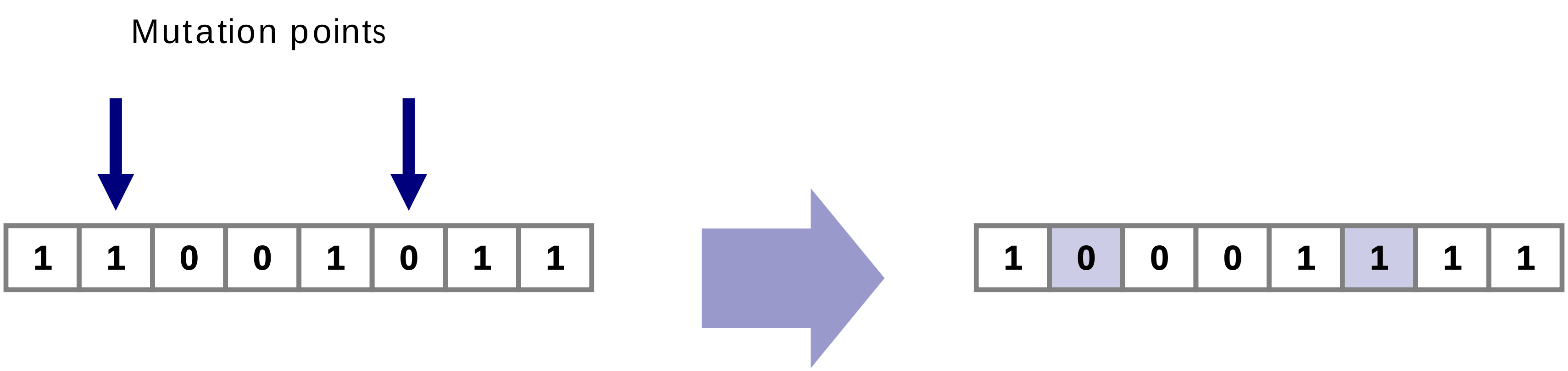
Single Point Crossover



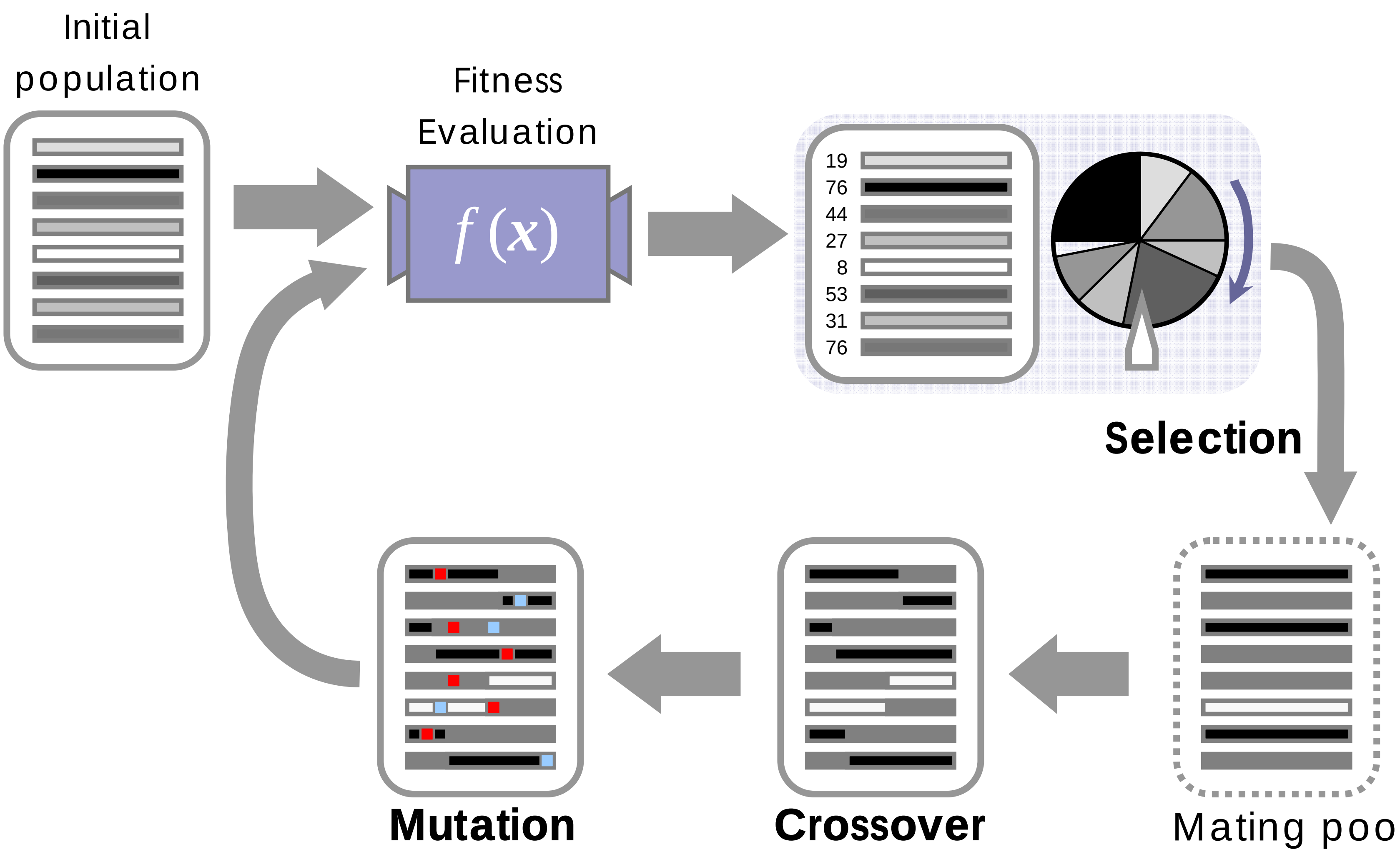
Mutation

- Mutation operator is applied **gene-wise**, that is, each gene undergoes mutation with the probability p_m
- When the mutation operation occurs to a gene, its gene value is flipped

$$1 \longrightarrow 0$$
$$0 \longrightarrow 1$$



Overview of Canonical GA



Summary of Canonical GA

- Binary representation
- Fixed string length
- Fitness proportional selection operator
- Single-point crossover operator
- Gene-wise mutation operator

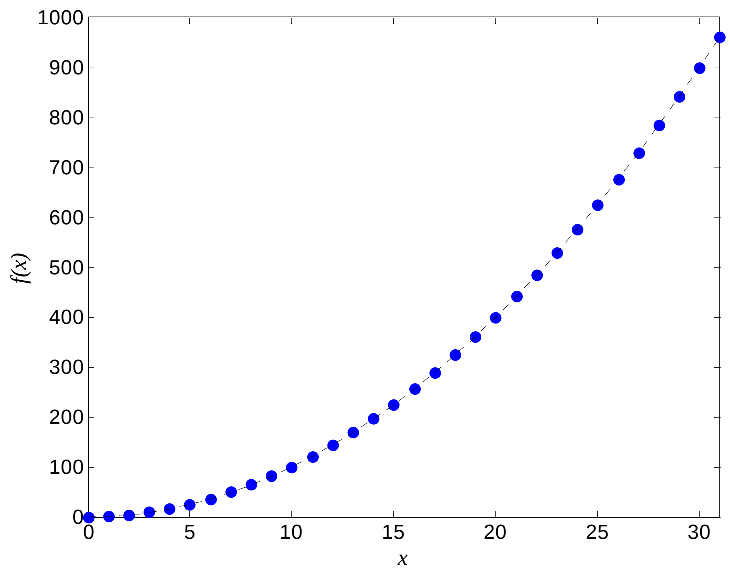
A Manual Example Using Canonical GAs

Problem Description

- Consider the following maximization problem

$$\max \quad f(x) = x^2$$

where x is an integer between 0 and 31



$f(x) = x^2$ has its maximum value 961 at $x = 31$

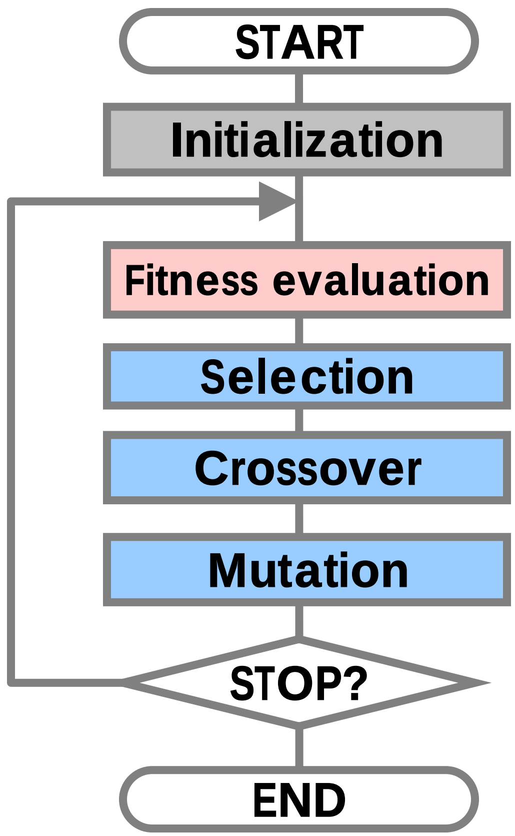
Gene Representation

- Before applying GA, a representation method must be defined
- Use unsigned binary integer of length 5

10110 ⇨ $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = \mathbf{22}$

- A five-bit unsigned binary integer can have values between 0(0000) and 31(11111)

Canonical GA Execution Flow



Initialization

- Initial populations are randomly generated
- Suppose that the population size is 4
- An example initial population

Individual No.	Initial population	x value
1	**0 1 1 0 1**	13
2	**1 1 0 0 0**	24
3	**0 1 0 0 0**	8
4	**1 0 0 1 1**	19

Fitness Evaluation

- Evaluate the fitness of initial population
- The objective function $f(x)$ is used as the fitness function
- Each individual is decoded to integer and the fitness function value is calculated

Fitness Evaluation

- Decoding

01000 $\xrightarrow{\text{Decode}}$ $x = 8$ $\xrightarrow{\text{Evaluation}}$ $f(8) = 8^2 =$ **64**

divide fi and average f

$f_i \leftrightarrow \bar{f}.$

- Results

Individual No.	Initial population	x value	$f(x)$	$f_i / \Sigma f$	Expected number
1	**0 1 1 0 1**	13	169	0.14	0.56
2	**1 1 0 0 0**	24	576	0.49	1.96
3	**0 1 0 0 0**	8	64	0.06	0.24
4	**1 0 0 1 1**	19	361	0.31	1.24

4

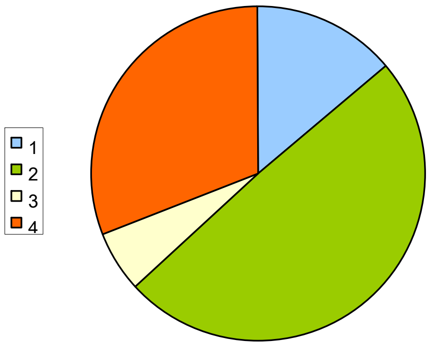
Selection

- Select individuals to the mating pool
- Selection probability is proportional to the fitness value of the individual
- Roulette wheel selection method is used

Individual No.	Initial population	x value	$f(x)$	$f_i / \Sigma f$	Expected number
1	**0 1 1 0 1**	13	169	0.14	0.56
2	**1 1 0 0 0**	24	576	0.49	1.96
3	**0 1 0 0 0**	8	64	0.06	0.24
4	**1 0 0 1 1**	19	361	0.31	1.24

Selection

- Roulette wheel



- Outcome of Roulette wheel is 1, 2, 2, and 4
- Resulting mating pool

No.	Mating pool
1	**0 1 1 0 1**
2	**1 1 0 0 0**
3	**1 1 0 0 0**
4	**1 0 0 1 1**

Crossover

- Two individuals are randomly chosen from mating pool
- Crossover occurs with the probability of $p_c = 1$
- Crossover point is chosen randomly

Mating pool	Crossover point	New population	x	$f(x)$
0 1 1 0 | 1	4	**0 1 1 0 0**	12	144
1 1 0 0 | 0		**1 1 0 0 1**	25	625
1 1 | 0 0 0	2	**1 1 0 1 1**	27	729
1 0 | 0 1 1		**1 0 0 0 0**	16	256

Mutation

- Applied on a bit-by-bit basis
- Each gene mutated with probability of $p_m = 0.001$

Before Mutation	After Mutation	x	$f(x)$
0 1 1 0 0	**0 1 1 0 0**	12	144
1 1 0 0 1	**1 1 0 0 1**	25	625
1 1 0 1 1	**1 1 0 1 1**	27	729
1 0 0 0 0	**1 0 0 1 0**	18	324

Fitness Evaluation

- Fitness values of the new population are calculated

Old population	x	$f(x)$	New population	x	$f(x)$
0 1 1 0 1	13	169	**0 1 1 0 0**	12	144
1 1 0 0 0	24	576	**1 1 0 0 1**	25	625
0 1 0 0 0	8	64	**1 1 0 1 1**	27	729
1 0 0 1 1	19	361	**1 0 0 1 0**	18	324
	sum	1170		sum	1756
	avg	293		avg	439
	max	576		max	729