



# Optimization (GAs)

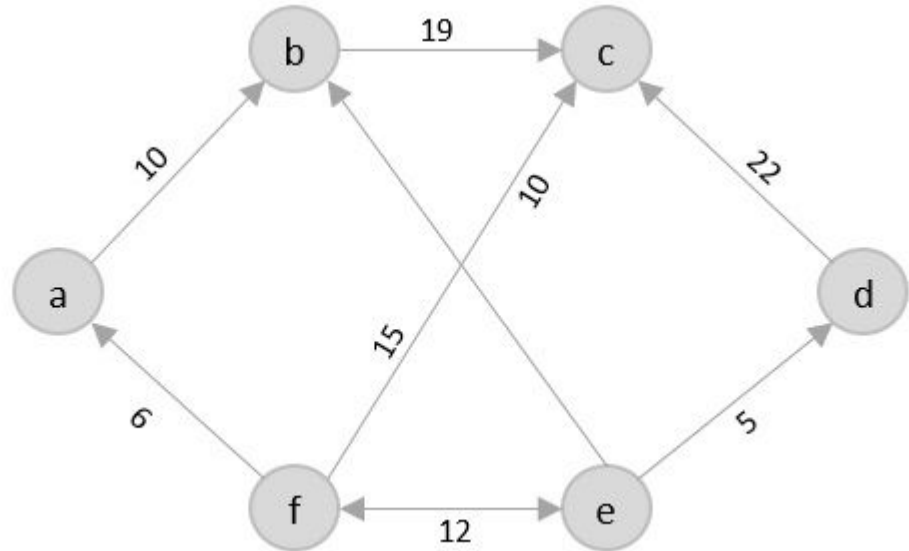


# Optimization problem

- Goal: Finding the best solution from all possible solutions.
- It is not always feasible... some problem may require too much computation time.
- Sometimes being close to the optimum is enough.

# Optimization problem

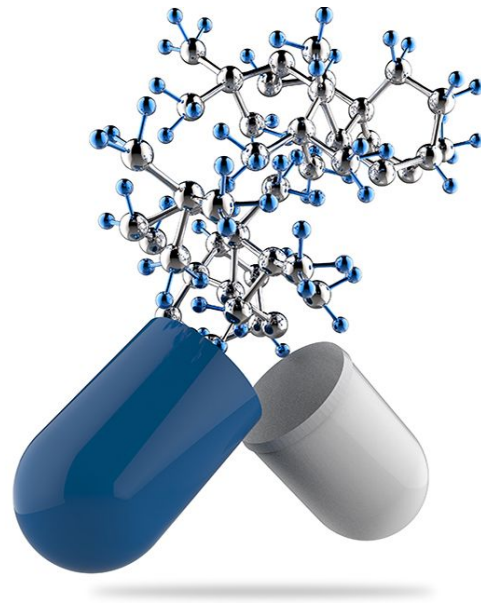
- Traveling Salesman Problem (TSP).





# Optimization problem

- Traveling Salesman Problem (TSP).
- Drug design.





# Optimization problem

- Traveling Salesman Problem (TSP).
- Drug design.
- Developing new materials.



# Optimization problem

- Traveling Salesman Problem (TSP).
- Drug design.
- Developing new materials.
- Scheduling.

Large Lecture Room Timetabling v1.0 - Microsoft Internet Explorer

Source: Úpravy Zobrazení Sdílení Naposledy | Adresa: <https://www.mos.purdue.edu/teaching/index.jsp>

**Timetable**

☒ Purdue Timetable

☒ Input Configuration

☐ Buildings

☐ Rooms

☐ Instructors

☐ Classes

☐ Constraints

☒ Timetable

☐ Solver

☐ Bundled LS Solver

☐ Alone LS Solver

☐ Conf. Statistics

☐ Administration

☐ Users

☐ Versions

☐ Export Input Cfg

☐ Timings

☐ Debug

☐ Data Configuration

Login: muller

Name: Tomas Muller

Date: FALL 2004 v93 r/o

Server: v1.0 aloha build149

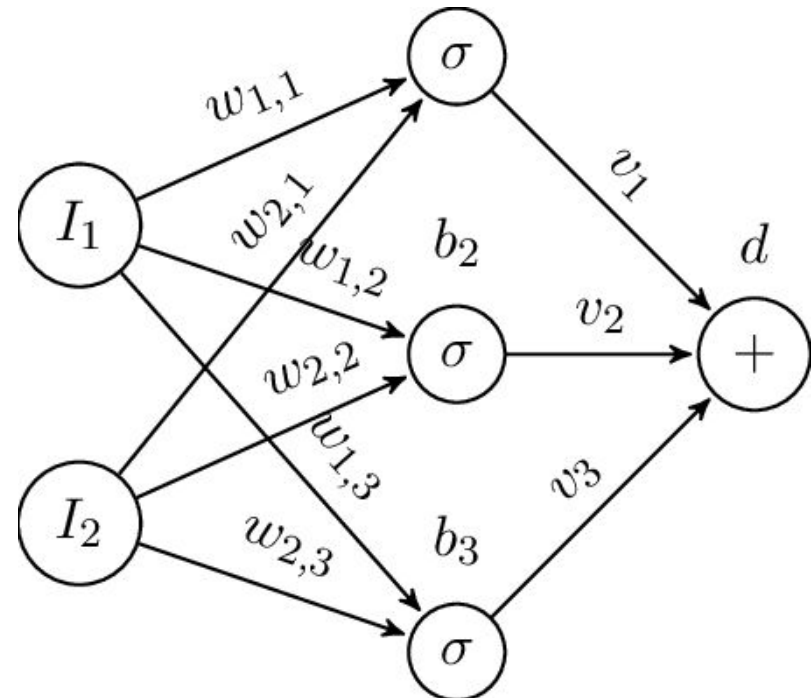
Fri, 11 Jun 2004

	7:30a	8:30a	9:30a	10:30a	11:30a	12:30a	1:30p	2:30p	3:30p
Mon		NGE228 1001 0, 1, 1	ECET209 1001 0, 1, 1	AUSL227 1001 2	CSR331 1001 0, 1, 0	MSE230 2001 0, 2, 0	HIST104 4001 2	ECE321 1001 0, 2, 0	PHPR202 1001 0, 0, 0
Tue	CPT365 1001 0, 1, 1	C E283 1001 0, 0, 0		PSY235 2001 0, 0, 0	HIST151 2001 0, 0, 0		ECE311 1001 0, 2, 0		ECE311 1001 0, 1, 0
Wed	ECET214 1001 0, 0, 0	MSE230 1001 0, 1, 0	ECET209 1001 0, 1, 0	AUSL227 1001 2	CSR331 1001 0, 0, 0	MSE230 2001 0, 2, 0	HIST104 4001 2	ECE321 1001 0, 2, 0	PHPR202 1001 0, 0, 0
Thu	CPT365 1001 0, 1, 1	C E283 1001 0, 0, 0		PSY235 2001 0, 0, 0	HIST151 2001 0, 0, 0		ECE311 1001 0, 2, 0		ECE311 1001 0, 1, 0
Fri	ECET214 1001 0, 0, 0		ECET209 1001 0, 1, 0	AUSL227 1001 2	CSR331 1001 0, 0, 0		HIST104 4001 2	ECE321 1001 0, 2, 0	PHPR202 1001 0, 0, 0
Mon	NUCL273 1001 0, 0, 0	NUCL300 1001 0, 0, 0	HTM181 1001 0, 0, 0	IE230 1001 1	C 8352 1001 0, 0, 0	KA161 3001 0, 0, 0	ECET278 1001 0, 0, 0	PSY200 2001 0, 0, 0	NUCL300 1001 0, 0, 0
Tue		EDPR235 1001 0, 0, 0	PHIL206 1001 2, 2	EA8221 1001 5	HIST103 4001 1		F&M202 1001 1		ECET278 1001 0, 0, 0
Wed	NUCL273 1001 0, 0, 0	NUCL300 1001 0, 0, 0	HTM181 1001 0, 0, 0	IE230 1001 1	C 8352 1001 0, 0, 0	KA161 3001 0, 0, 0	ECET278 1001 0, 0, 0	PSY200 2001 0, 0, 0	ECET198 1001 0, 0, 0
Thu		EDPR235 1001 0, 0, 0	PHIL206 1001 2, 2	EA8221 1001 5	HIST103 4001 1		F&M202 1001 1		ECET278 1001 0, 0, 0
Fri	NUCL273 1001 0, 0, 0	NUCL300 1001 0, 0, 0	HTM181 1001 0, 0, 0	IE230 1001 1	C 8352 1001 0, 0, 0	KA161 3001 0, 0, 0	ECET278 1001 0, 0, 0	PSY200 2001 0, 0, 0	ECET198 1001 0, 0, 0
Mon	CSR342 1001 0, 0, 0	MA161 1001 0, 0, 0	ENGR108 1001 0, 0, 0	PSY120 4001 0, 0, 0	MA161 2001 0, 0, 0	KA162 1001 0, 0, 0	PHIL330 1001 2	AGRY320 1001 5, 2, 2	MA162 2001 0, 1, 1
Tue	AGEC217 2001 0, 0, 0		ENGR108 2001 0, 0, 0	HIST152 2001 1	ECEN251 2001 0, 0, 0	PSY120 1001 1, 2		PSY120 5801 3, 1, 1	
Wed	CSR342 1001 0, 0, 0	MA161 1001 0, 0, 0	ENGR108 3001 0, 0, 0	PSY120 4001 0, 0, 0	MA161 2001 0, 0, 0	KA162 1001 0, 0, 0	PHIL330 1001 2	AGRY320 1001 5, 2, 2	MA162 2001 0, 1, 1

Internet

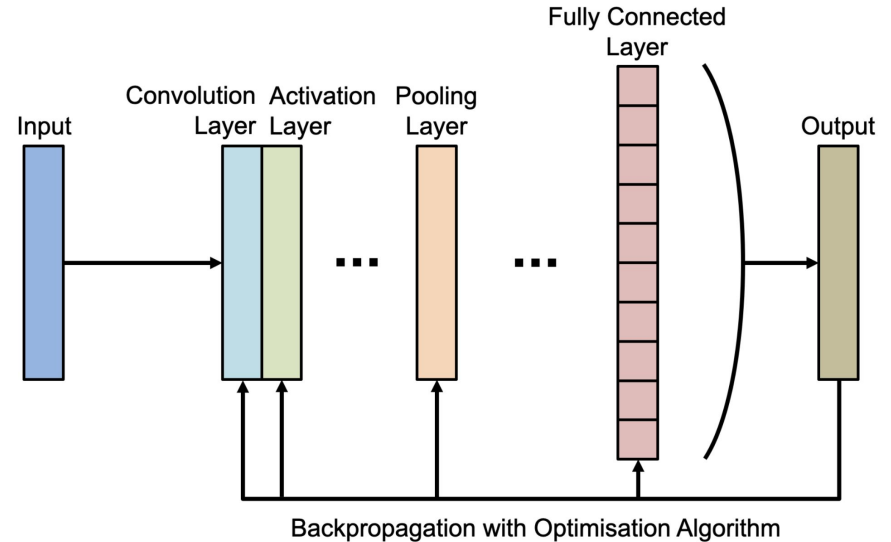
# Optimization problem

- Traveling Salesman Problem (TSP).
- Drug design.
- Developing new materials.
- Scheduling.
- Optimize weights.



# Optimization problem

- Traveling Salesman Problem (TSP).
- Drug design.
- Developing new materials.
- Scheduling.
- Optimize weights.
- NN architecture.







# Famous algorithms

- Linear programming (LP)
  - Simplex

Consider the following linear programming (LP):

$$\text{Max.} \quad z = 2x_1 + 3x_2$$

$$\text{Such that} \quad 2x_1 + x_2 \leq 4$$

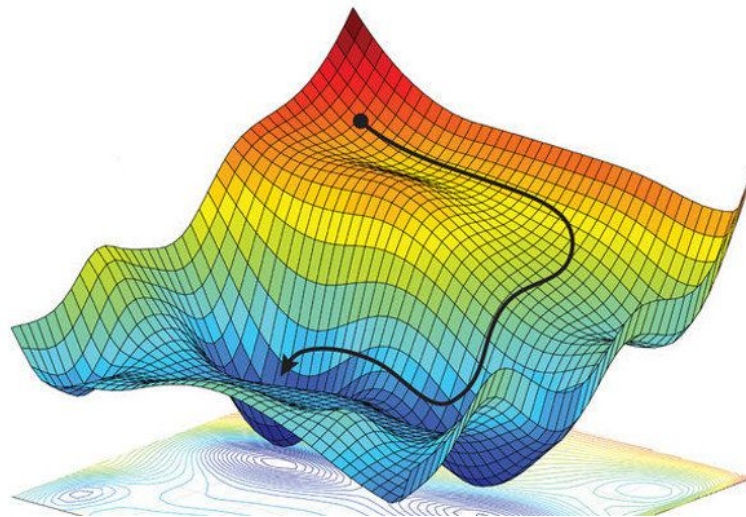
$$x_1 + 2x_2 \leq 5$$

$$x_1, x_2 \geq 0$$

The optimum value of the LP is

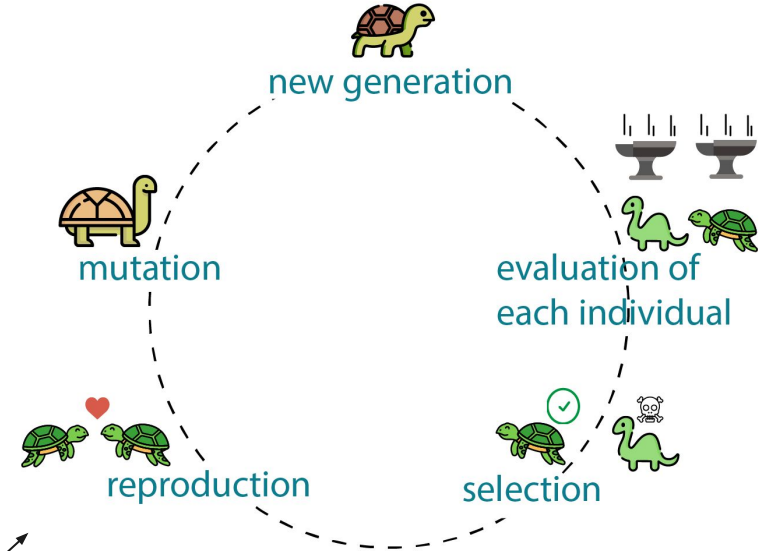
# Famous algorithms

- Linear programming (LP)
  - Simplex
- Nonlinear programming
  - Gradient descent



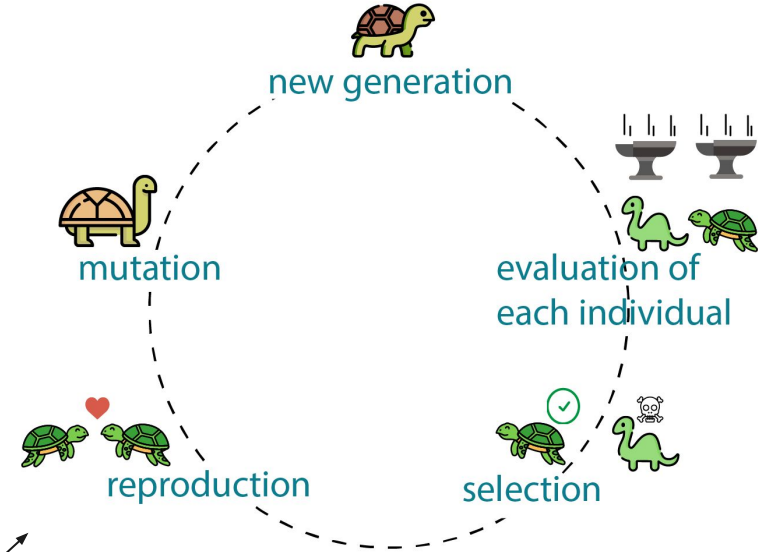
# Famous algorithms

- Linear programming (LP)
  - Simplex
- Nonlinear programming
  - Gradient descent
- Heuristic
  - Genetic algorithms (GAs)
  - Simulated annealing (SA)
- ...



# Famous algorithms

- Linear programming (LP)
  - Simplex
- Nonlinear programming
  - Gradient descent
- Heuristic
  - Genetic algorithms (GAs)
  - Simulated annealing (SA)
- ...





# GAs

- Inspired by the process of natural selection and genetic mechanisms.



# GAs

- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.



# GAs

- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.



# GAs

- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.
- Powerful in exploring large and complex search spaces.





# GAs

- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.
- Powerful in exploring large and complex search spaces.
- Can provide near-optimal solutions where other methods fail.



# GAs

- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.
- Powerful in exploring large and complex search spaces.
- Can provide near-optimal solutions where other methods fail.
- No guarantee of finding the global optimum solution.



# GAs

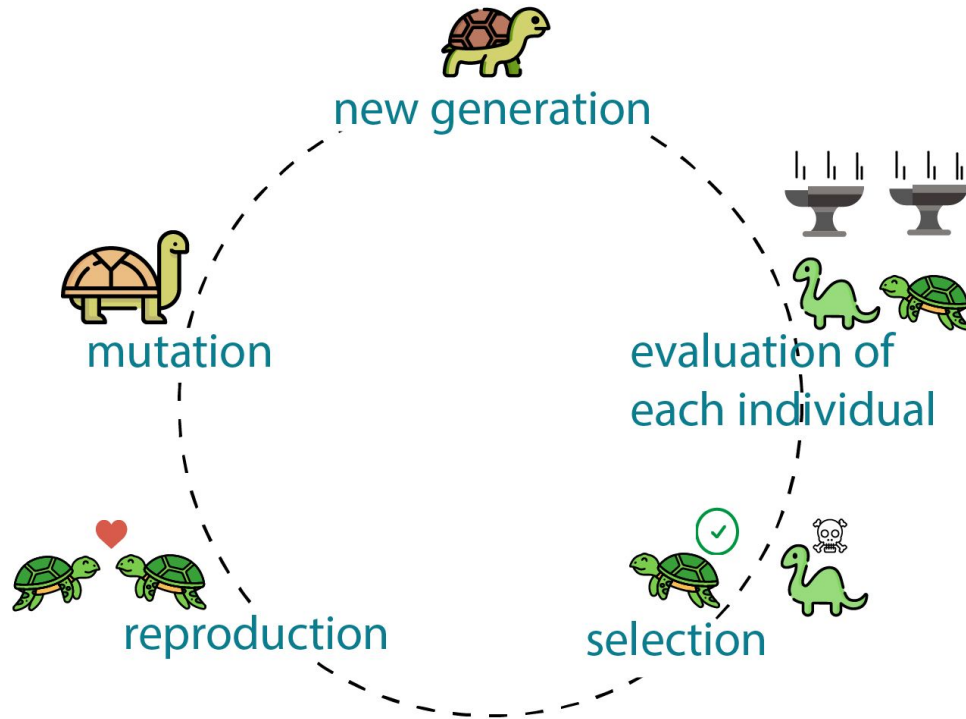
- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.
- Powerful in exploring large and complex search spaces.
- Can provide near-optimal solutions where other methods fail.
- No guarantee of finding the global optimum solution.
- May require careful tuning of parameters and can be computationally intensive.



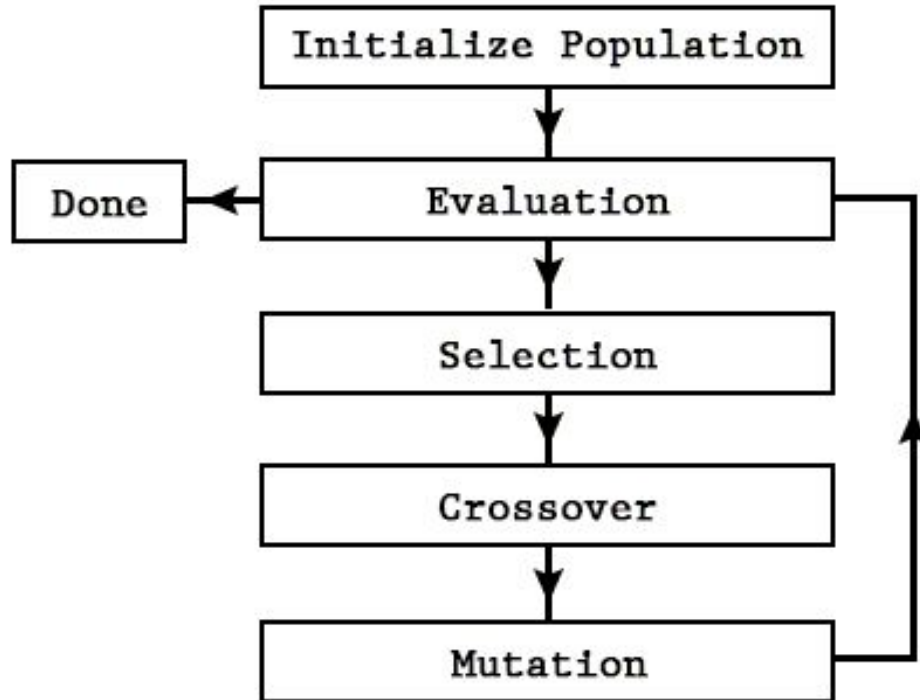
## Why GAs?

- Robustness.
- Versatility.
- Parallelism.
- Global search capability.
- Creativity.
- Incorporation of domain knowledge.

# GAs



## GENETIC ALGORITHM FLOW CHART





## Problem example

- Maximize the number of 1's in a binary sequence.



# Individual

- Decide how to encode each individual.





# Individual

- Decide how to encode each individual.
- In our problem:

1	0	0	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---



# Initialize population

- Techniques:
  - Randomly.
  - Heuristic-based with domain knowledge.
  - Diverse (ensure diversity).
  - Hybrid.



## Evaluation/Fitness

- Decide how to evaluate each individual.



## Evaluation/Fitness

- Decide how to evaluate each individual.
- In our problem:
  - Sum the number of 1's in the individual.
  - Higher evaluations mean better solutions.

1	0	0	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---

Fitness = 4





# Selection

- Decide how individuals are selected.
- Techniques:
  - **Roulette wheel:** Higher fitness gives a higher chance of being selected.



# Selection

- Decide how individuals are selected.
- Techniques:
  - **Roulette wheel:** Higher fitness gives a higher chance of being selected.
  - **Tournament:** A set of individuals are randomly chosen. The fittest is selected.



# Selection

- Decide how individuals are selected.
- Techniques:
  - **Roulette wheel:** Higher fitness gives a higher chance of being selected.
  - **Tournament:** A set of individuals are randomly chosen. The fittest is selected.
  - **Elitism:** Fittest individuals are automatically carried over to the next generation, ensuring that the best traits persist.



# Crossover

- Decide how individuals are merged/reproduced.
- Techniques:
  - **One-Point:** A single crossover point is selected. The genes beyond that point are swapped between two parent individuals.





# Crossover

- Decide how individuals are merged/reproduced.
- Techniques:
  - **One-Point:** A single crossover point is selected. The genes beyond that point are swapped between two parent individuals.
  - **Two-Point:** Two crossover points are chosen. The genes between these points are swapped between the parent individuals.



# Crossover

- Decide how individuals are merged/reproduced.
- Techniques:
  - **One-Point:** A single crossover point is selected. The genes beyond that point are swapped between two parent individuals.
  - **Two-Point:** Two crossover points are chosen. The genes between these points are swapped between the parent individuals.
  - **Uniform:** Each gene is independently considered. For each gene, one of the two parent genes is randomly selected for the offspring.



# Mutation

- Decide how offspring are mutated.
- Techniques:
  - **Uniform:** Iterate through each gene in an individual and decide, based on a predefined mutation rate, whether to mutate that gene.



# Mutation

- Decide how offspring are mutated.
- Techniques:
  - **Uniform:** Iterate through each gene in an individual and decide, based on a predefined mutation rate, whether to mutate that gene.
  - **Bit Flip:** In binary representation, flip a bit with a certain mutation probability.



# Mutation

- Decide how offspring are mutated.
- Techniques:
  - **Uniform:** Iterate through each gene in an individual and decide, based on a predefined mutation rate, whether to mutate that gene.
  - **Bit Flip:** In binary representation, flip a bit with a certain mutation probability.
  - **Scramble:** A subset of genes is chosen randomly and their values are shuffled.