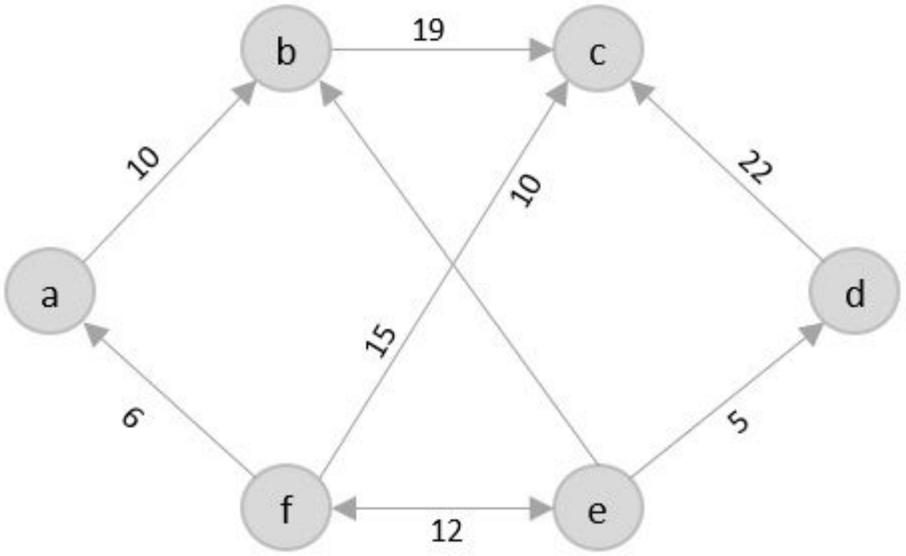
Optimization (GAs)

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.

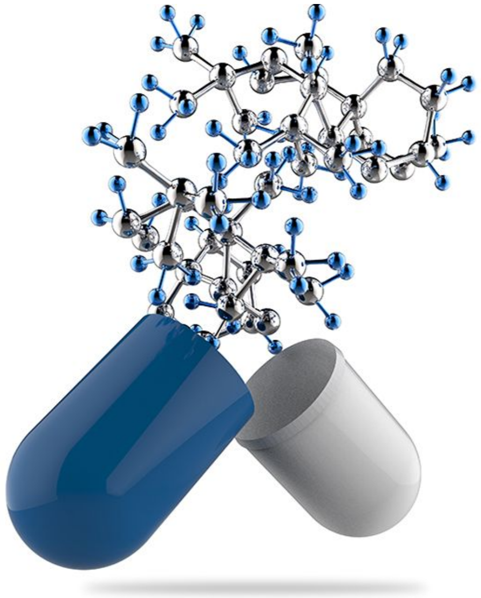
Optimization problem

- Traveling Salesman Problem (TSP).



Optimization problem

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

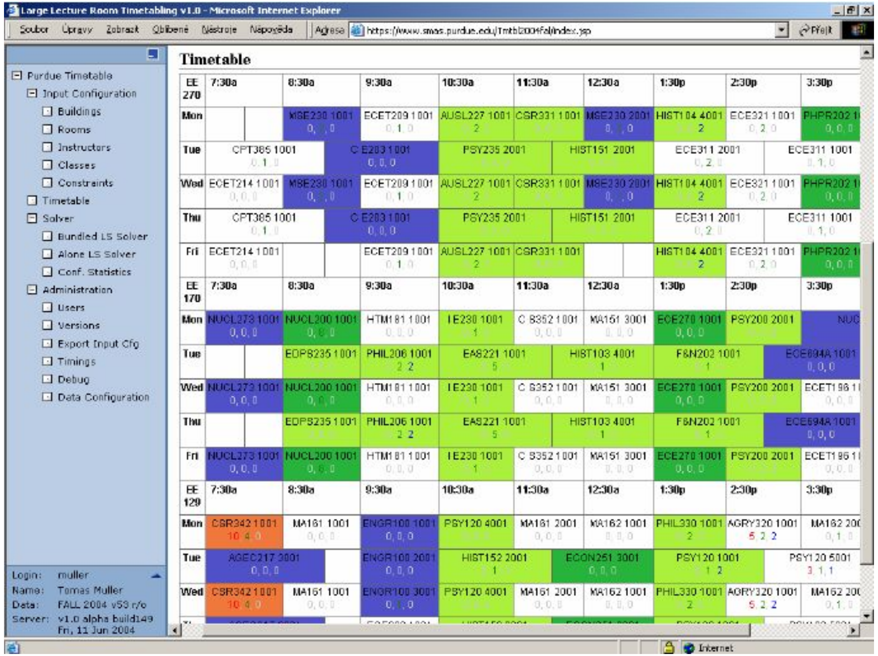


Optimization problem

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

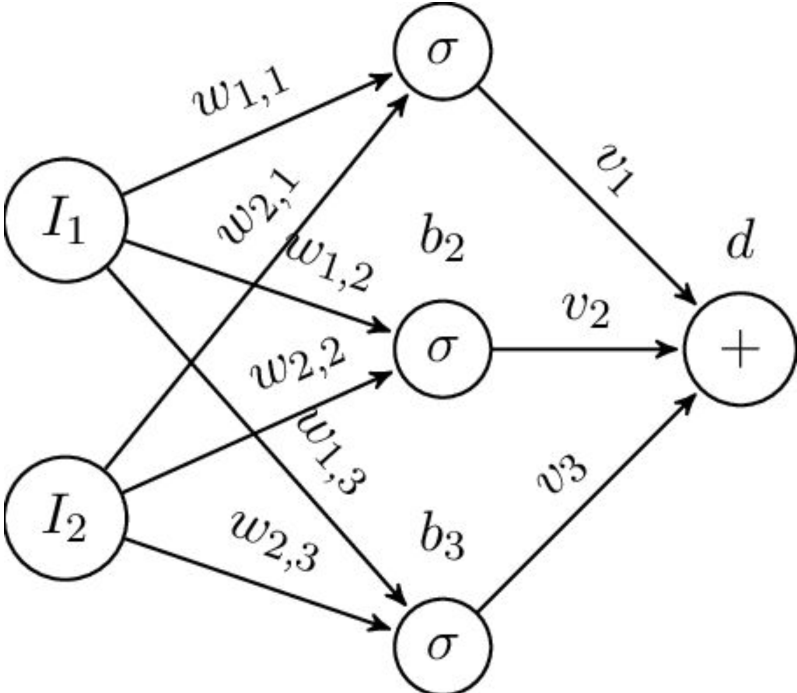
Optimization problem

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



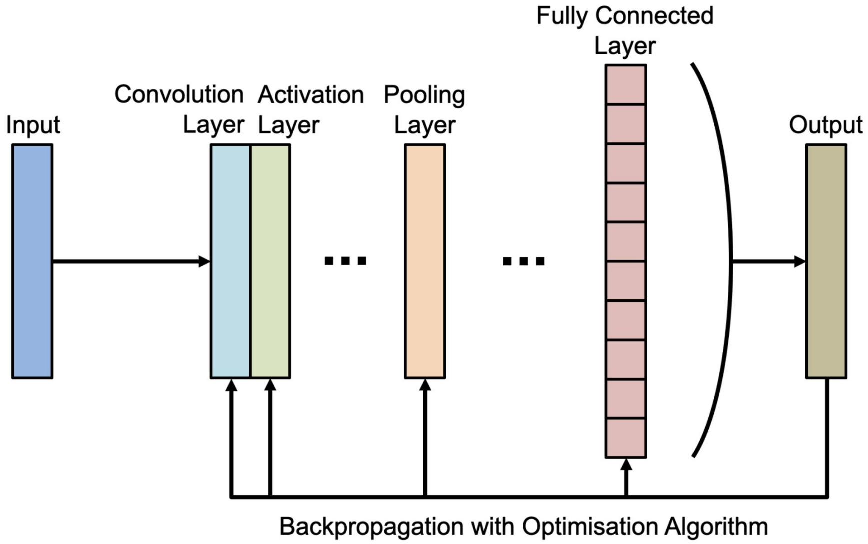
Optimization problem

- 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

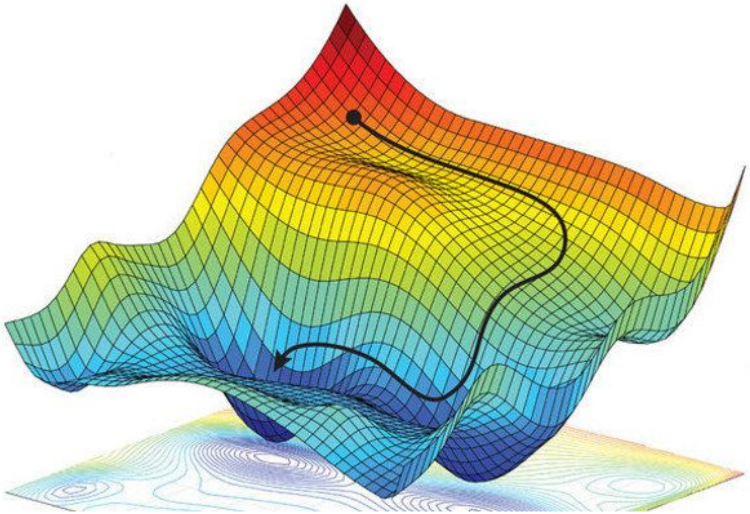
Consider the following linear programming (LP):

$$\begin{array}{ll} \text{Max.} & z = 2x_1 + 3x_2 \\ \text{Such that} & 2x_1 + x_2 \leq 4 \\ & x_1 + 2x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{array}$$

The optimum value of the LP is

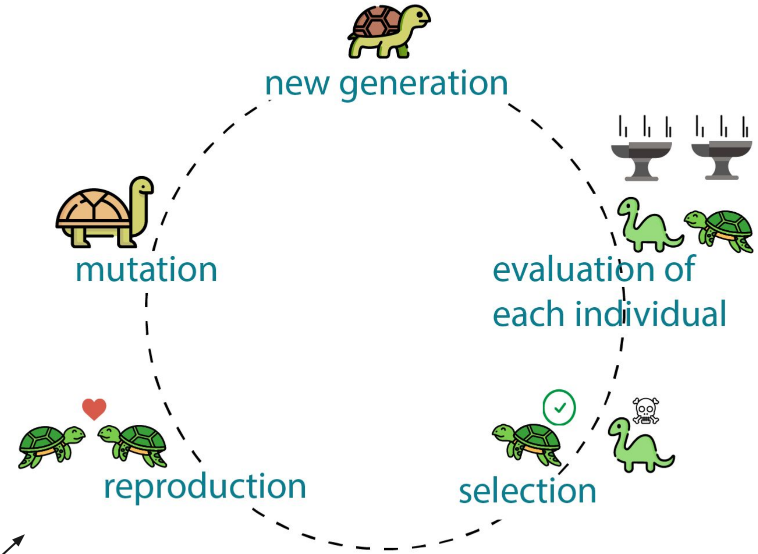
Famous algorithms

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



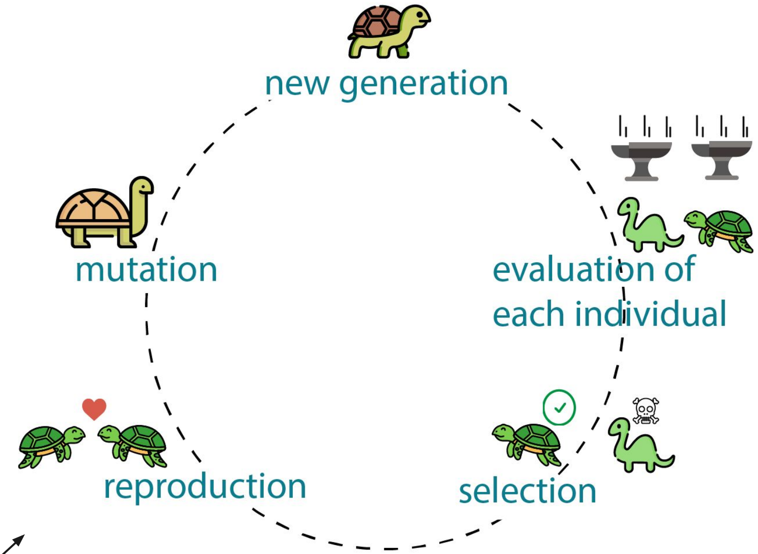
Famous algorithms

- 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.

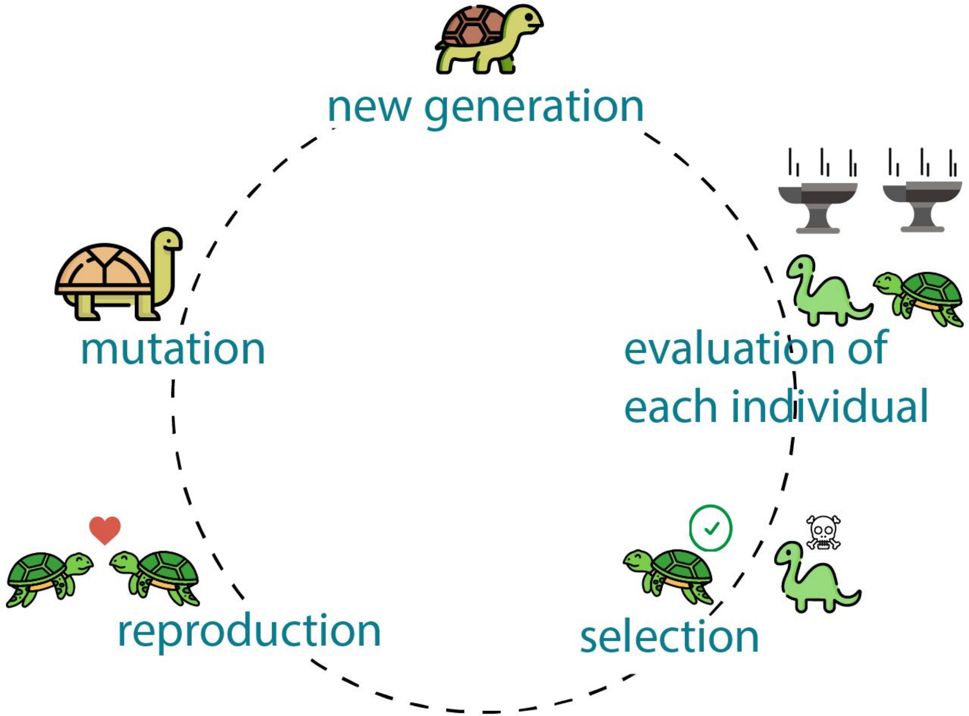
GAs

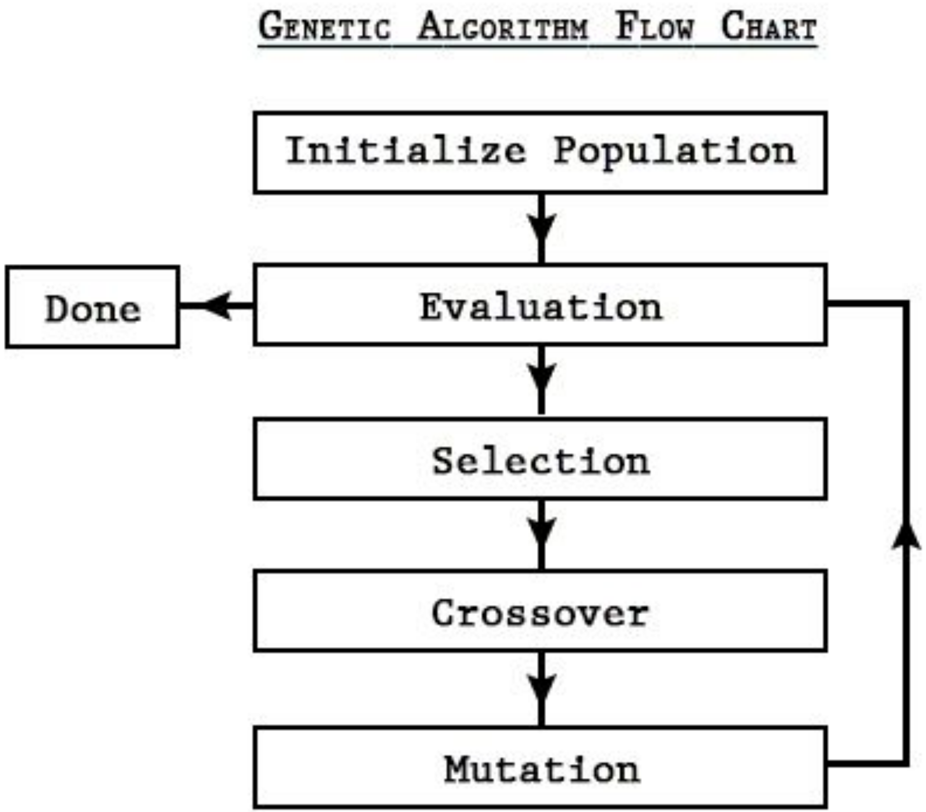
- Inspired by the process of natural selection and genetic mechanisms.
- Developed to solve complex optimization and search problems.
- They are a subset of a broader class of algorithms known as evolutionary algorithms.
- Powerful in exploring large and complex search spaces.
- Can provide near-optimal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


new generation
mutation
evaluation of each individual
reproduction
selection


GENETIC ALGORITHM FLOW CHART
Initialize Population
Evaluation
Done
Selection
Crossover
Mutation

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.

Fitness = 4

1	0	0	1	0	0	0	1	1	0

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.