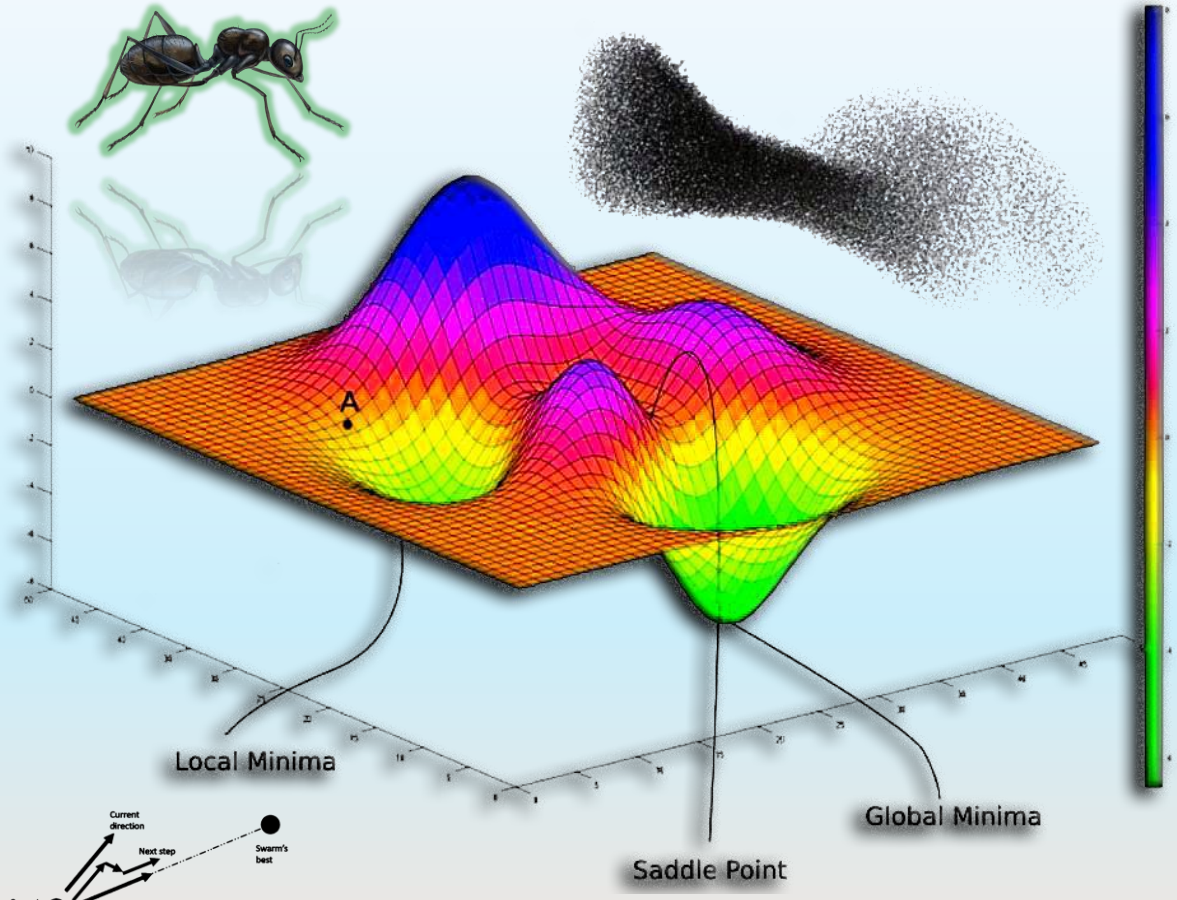
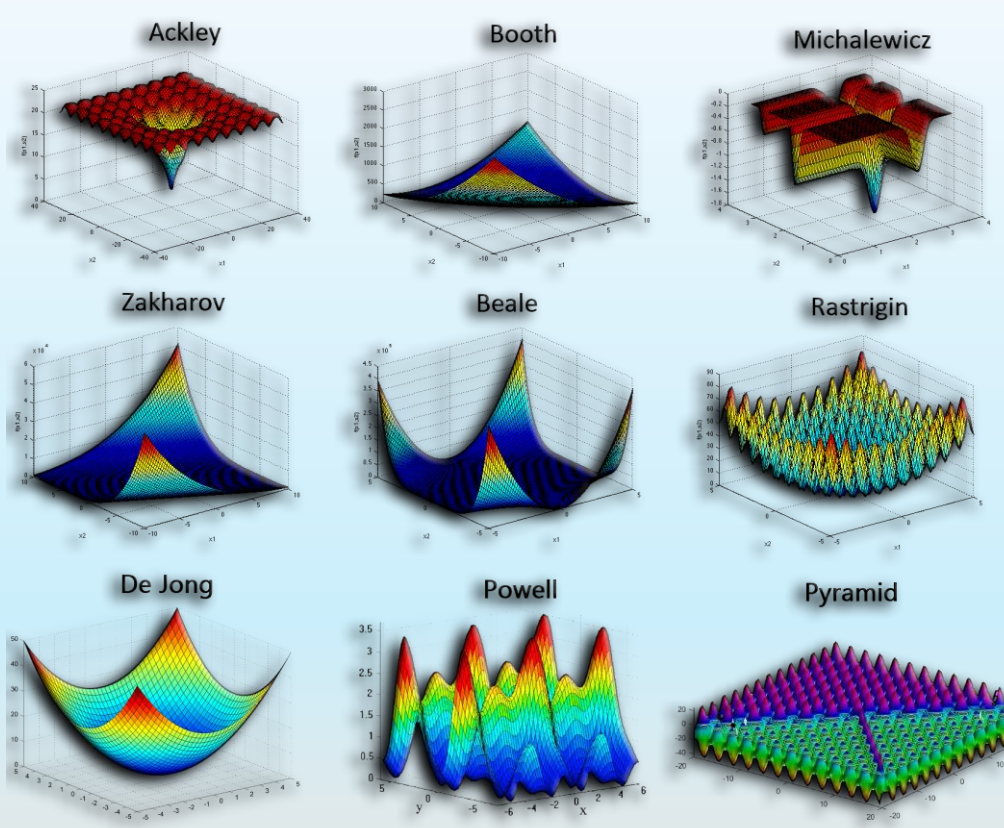
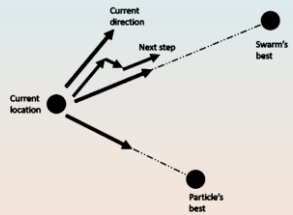
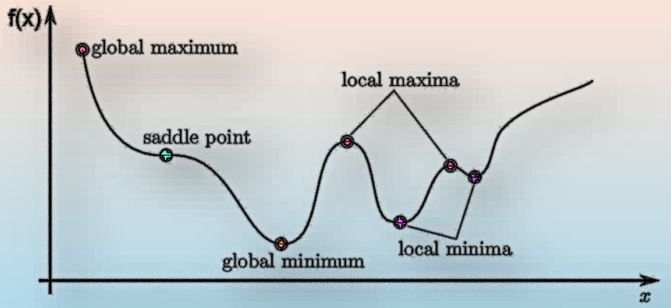






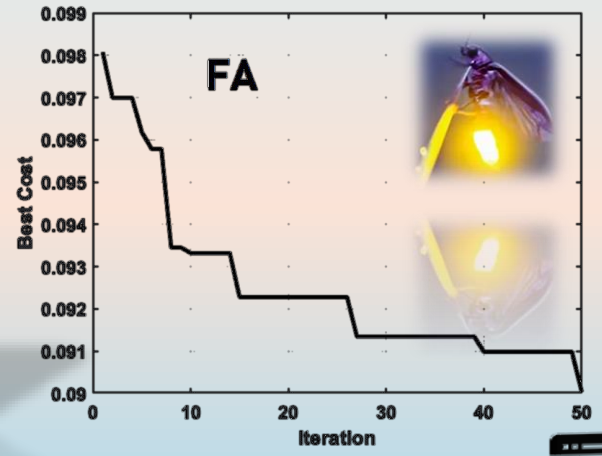


Metaheuristic Optimization: 4 Cutting-Edge Applications

Basics of Optimization

By: Seyed Muhammad Hossein Mousavi

2025

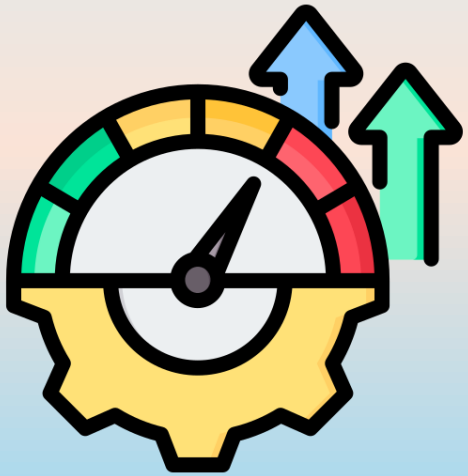


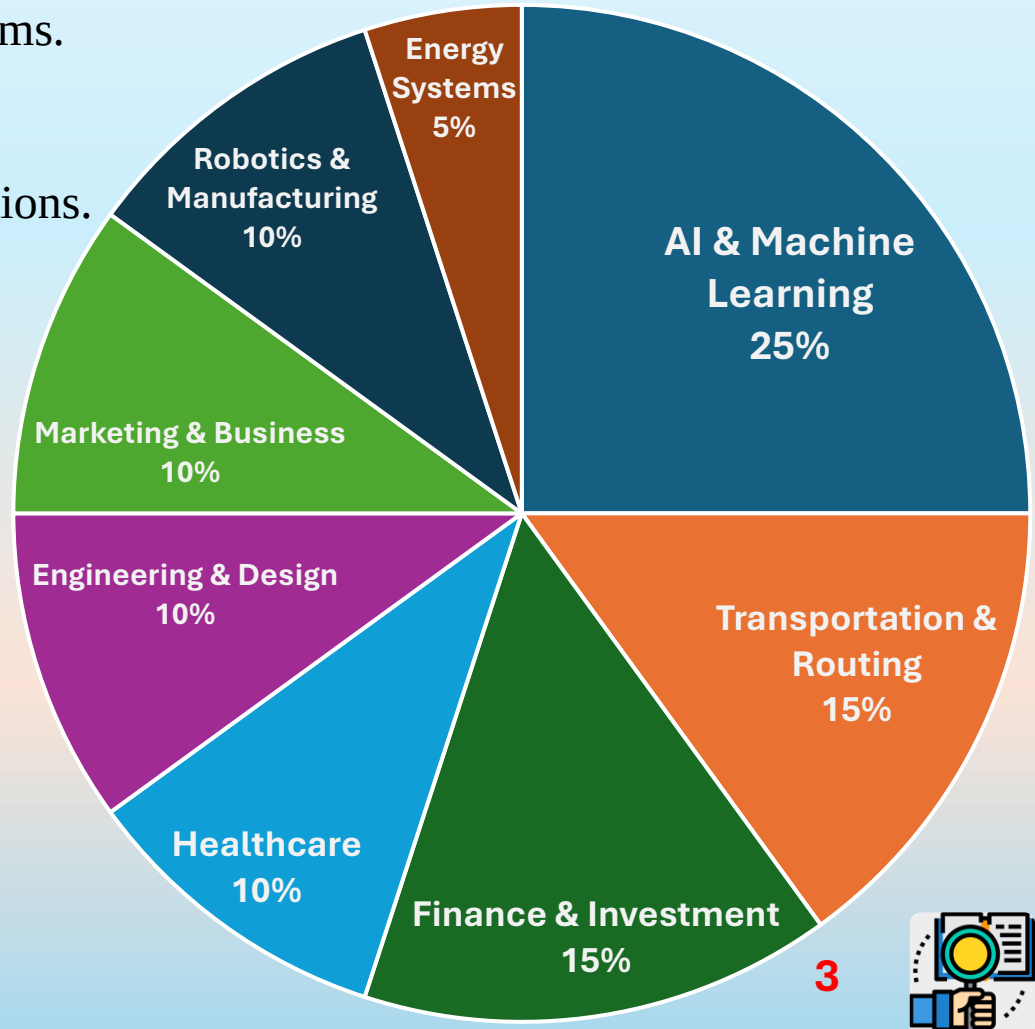
Outline:

- **Optimization**
  - ❖ **Why Optimization?**
  - ❖ **Definitions and Importance**
  - ❖ **Main Types**
  - ❖ **Algorithms**
  - ❖ **Optimization Test Functions**

Book: Kochenderfer, Mykel J., and Tim A. Wheeler. Algorithms for optimization. Mit Press, 2019.

- **Optimization**
  - ❖ **Why Optimization?**
    - Optimization algorithms are used everywhere, from **AI assistants to delivery apps**.
    - Their main goal is to find **the best solution under specific constraints**.
    - They are the **engine of smart decision-making** in many systems.
    - So, they are the **foundation** of intelligent systems.
    - They help you build **smarter, faster, and more efficient** solutions.
    - They power everyday tech: from Spotify, to Google, to Tesla.





- **Optimization**
  - ❖ **Why Optimization?**
    - 🎯 AI & Machine Learning: Tuning deep learning models and more…
    - 🗺️ Navigation & Routing: Finding fastest delivery routes (Google Maps, Uber Eats)
    - 🏥 Healthcare Optimizing: Drug dosages or radiation therapy plans
    - 💰 Finance: Balancing risk vs. return in investment portfolios
    - 🕹️ Game Development: Smoothing character movement or optimizing game AI
    - 📈 Marketing & Business: Selecting top-performing ads or pricing strategies

Hyper Parameter Tuning	Feature Selection	Clustering	Image Segmentation	Image Quantization	Minimum spanning Tree	Hub Location Allocation
Quadratic Assignment Problem	Regression	Economic Dispatching	Parallel Machine Scheduling	Control Systems	Natural Language Processing	Portfolio Optimization
Supply Chain	Signal Denoising	Bin Packing Problem	Weight and Bias Optimization	Evolutionary Art	Latent Space Optimization	Resource Allocation
Vehicle Routing Problem	Protein Structure Prediction	Space-Time Warping	Exoplanetary Adaptation Simulation	Evolved Antenna Design	Travelling salesman problem	Parameter Estimation

Some Applications in AI

- **Optimization**
  - ❖ **Definitions and Importance**
    - Optimization is the process of **finding the best solution or outcome** from a **set of possible solutions while satisfying given constraints**.

    - **Constraints** are the limits or conditions that the solution must meet, like time, cost, or resources.

    - It involves **maximizing or minimizing a specific objective function**, such as **cost or fitness**.

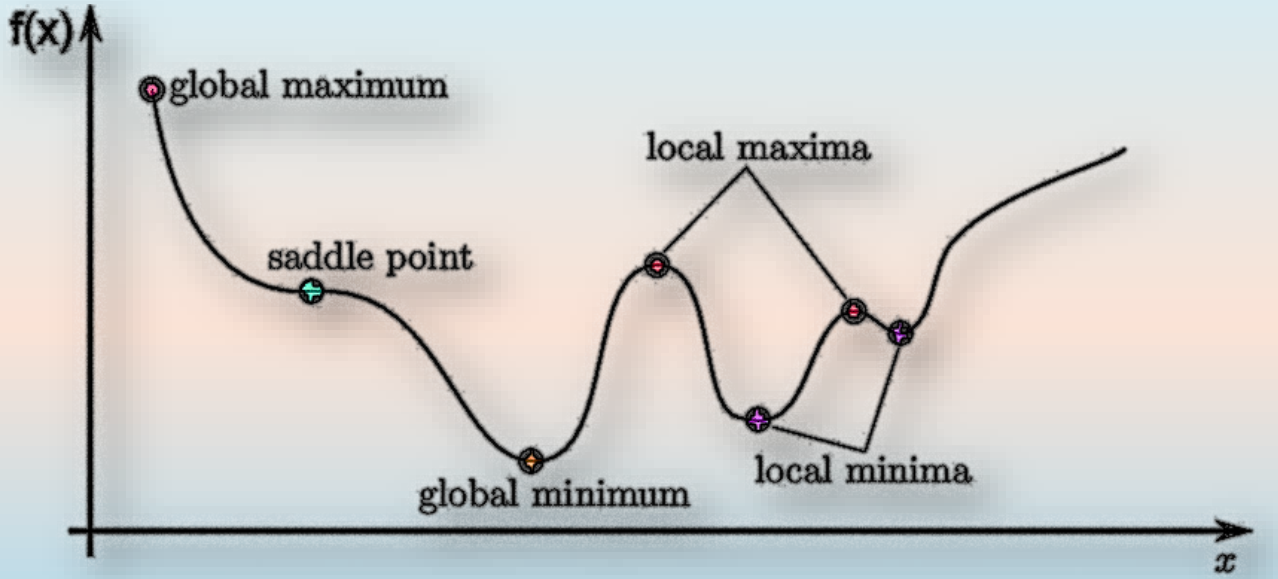
- **Optimization**
  - ❖ **Definitions and Importance**
    - An **objective function** is a mathematical expression that **defines the goal of an optimization problem**.
    - **It evaluates the quality of a solution** and guides the search for the optimal result. For instance:

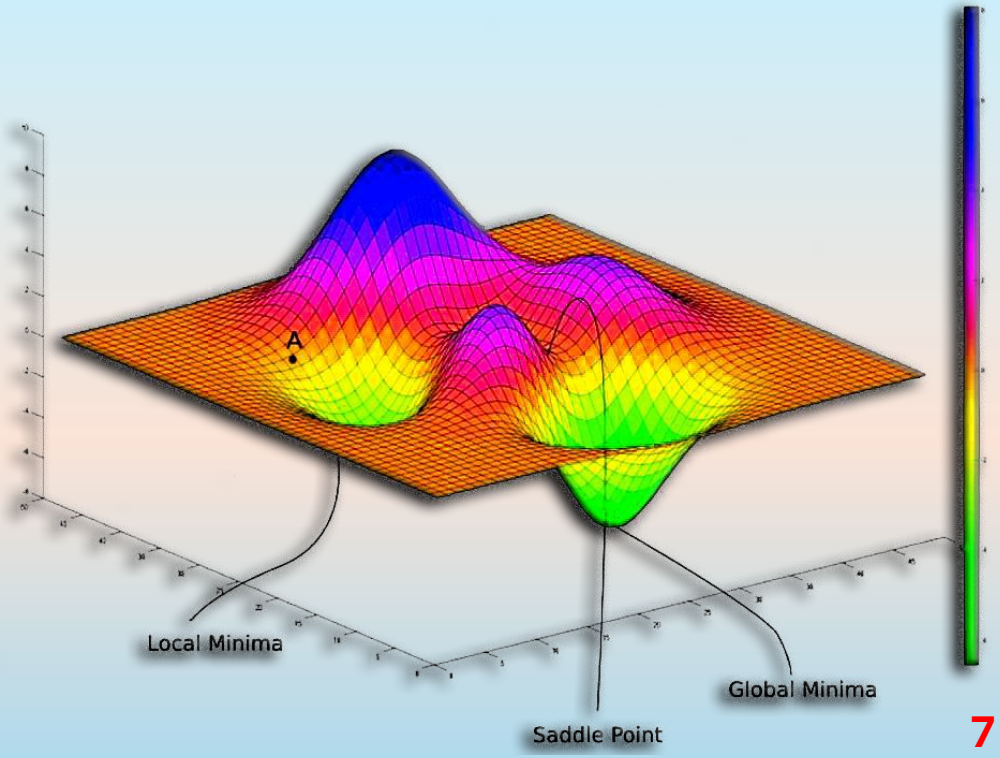
Min f(x) = x^2

- The goal is to minimize the function $f(x) = x^2$.
- This means we are looking for the smallest value of x^2
- The minimum or best solution (if cost) happens at $x=0$, where $f(x)=0$.

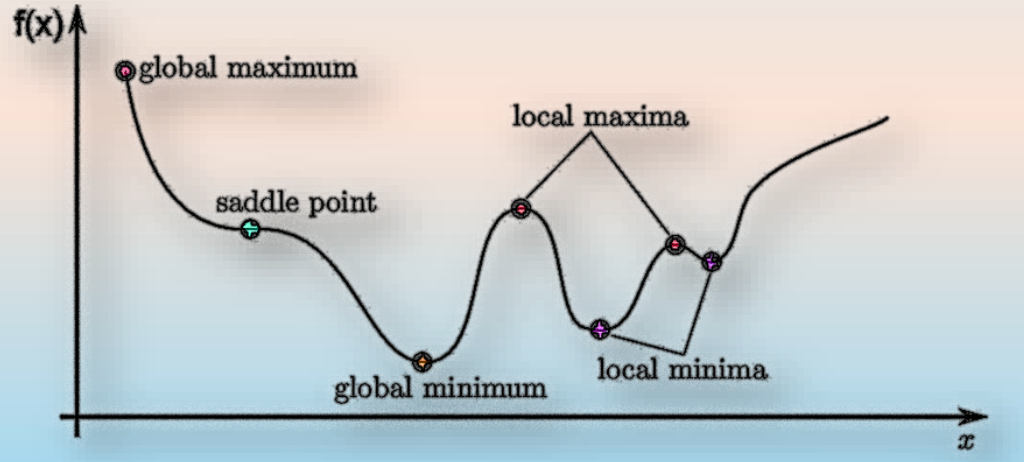
    - **Potential solutions** in optimization problems are called **critical points.**
    - Critical points happen where the gradient of the objective function is **zero or undefined.**
    - This means there is no slope, and the function "**flattens out**" at these points.

- **Optimization**
  - ❖ **Definitions and Importance**
    - Imagine a hill:
      - At the top of a hill, the **slope is zero** → this is a **maximum (local maxima(s) and global maximum)**.
      - At the bottom of a valley, the **slope is zero** → this is a **minimum (local minima(s) and global minimum)**.
      - In between which is not maximum or minimum (**undefined**) → this is called a **saddle point**.

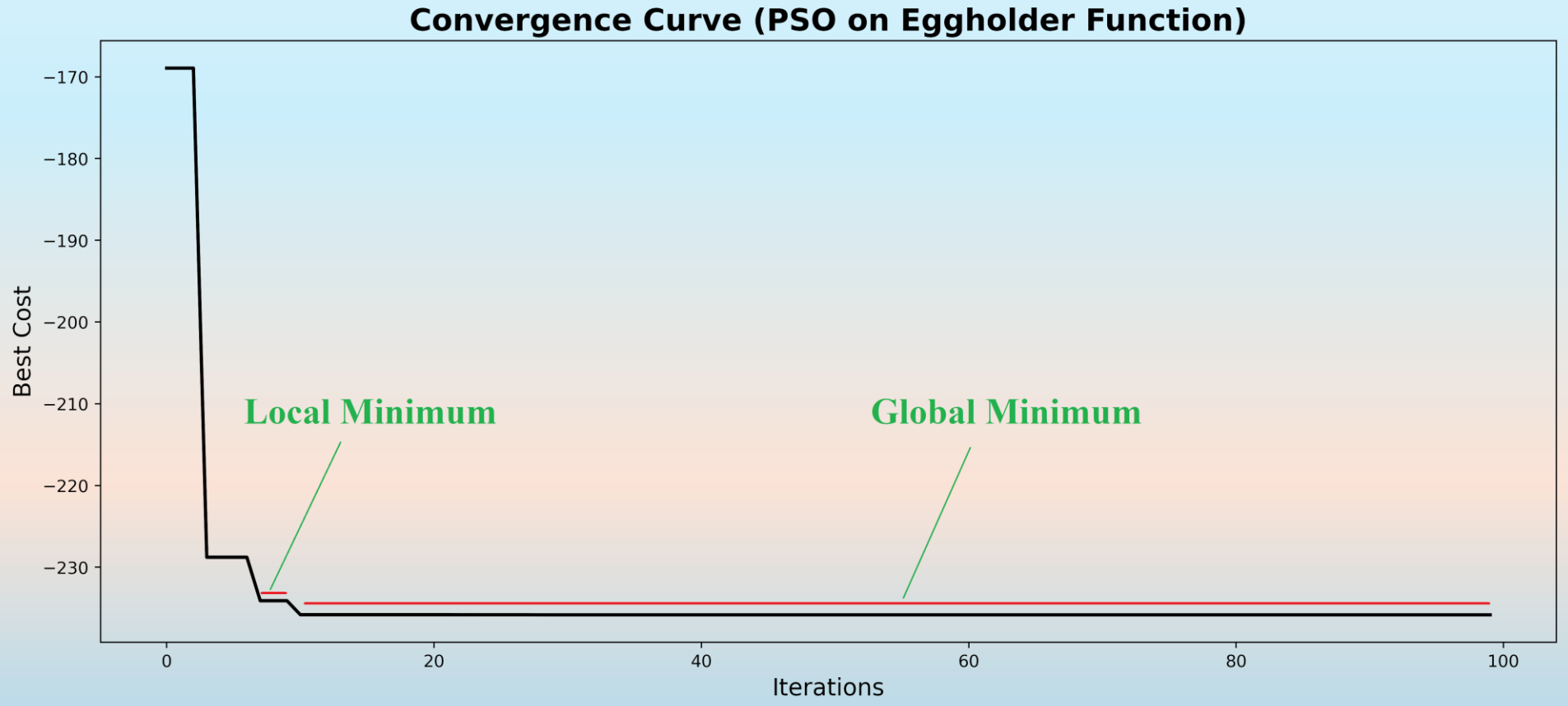




- # Optimization
  - ## ❖ Definitions and Importance
    - Critical points are classified into four main categories:
      - **Global Maximum**: The **highest value** of the objective function across the **entire domain** of the problem (**the best in the case of the fitness function**).
      - **Global Minimum**: The **lowest value** of the objective function across the **entire domain** of the problem (**the best in the case of the cost function**).
      - **In most cases, we are looking for the best value for the cost function.**
      - **Local Maximum**: The highest value of the objective function within a **specific region** or neighborhood.
      - **Local Minimum**: The lowest value of the objective function within a **specific region** or neighborhood.
      - **Saddle Point**: A point where the objective function is **neither a local maximum nor a local minimum.**

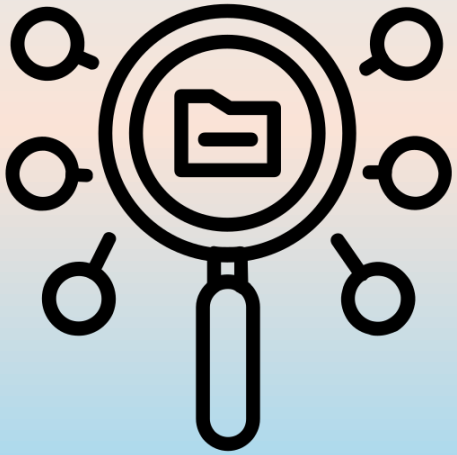


- **Optimization**
  - ❖ **Definitions and Importance**
    - **Convergence**: It is the process of an algorithm gradually approaching a solution, ideally the optimal one.
    - It means that changes in the objective value or parameters become very small over iterations, reaching stability.



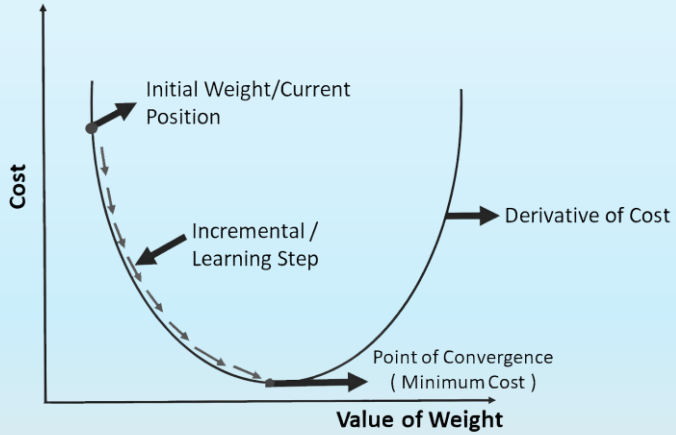
- **Optimization**
  - ❖ **Definitions and Importance**
    - Exploration and Exploitation:
      - ➢ **Exploration** is the **primary phase**, where the **algorithm searches broadly across the solution space to identify promising regions or solutions**. It helps discover areas that might contain the global optimum.

      - ➢ **Exploitation** is the **secondary phase** where the algorithm focuses on **refining or improving solutions within those promising regions** discovered during exploration. It fine-tunes the results to approach the optimum (the best solution).



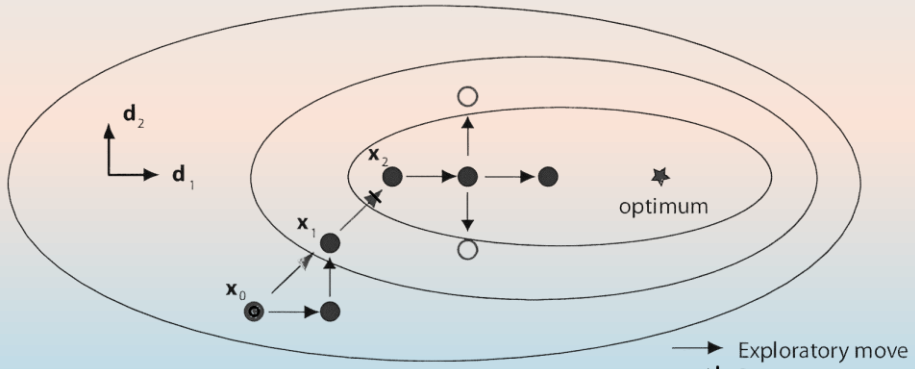
- # Optimization
  - ## ❖ Definitions and Importance
    - **Efficiency Improvement**: Optimization **minimizes resource usage (time, cost, energy)** **while achieving desired outcomes**, and creates more efficiently.
    - **Cost Reduction**: It helps **reduce expenses in industries** by finding the **most cost-effective solutions without compromising quality** or performance.
    - **Performance Enhancement**: Optimization improves system or product performance by **fine-tuning parameters** to achieve the best results under given constraints.
    - **Decision-Making Support**: It provides data-driven, **optimal choices in complex scenarios**, aiding businesses and individuals in making better decisions.
    - **Maximizing Profits**: By **optimizing resource allocation and processes**, organizations can maximize returns, productivity, and profitability.
    - **Sustainability Promotion**: Optimization contributes to **sustainable practices by reducing waste, energy consumption**, and environmental impact in operations.
    - **Innovation and Advancement**: It drives innovation by **finding creative solutions to complex problems** and pushing the boundaries of technology and design.

- **Optimization**
  - ❖ **Main Types**
    - **Gradient-Based Optimization**: Relies on derivatives to iteratively find optima, suitable for smooth and differentiable problems.
      - Gradient Descent (GD)
      - Stochastic Gradient Descent (SGD)
      - Newton's Method



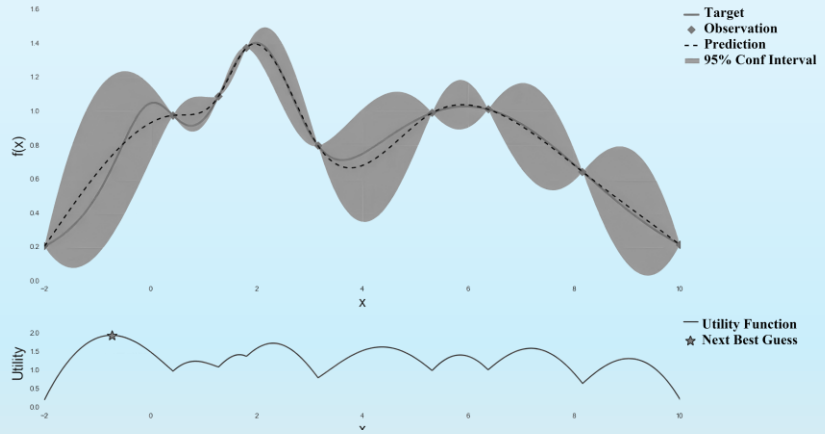
Gradient Descent

    - **Gradient-Free Optimization**: Operates without gradient information, ideal for non-differentiable or noisy problems.
      - Nelder-Mead
      - Powell's Method
      - Pattern Search



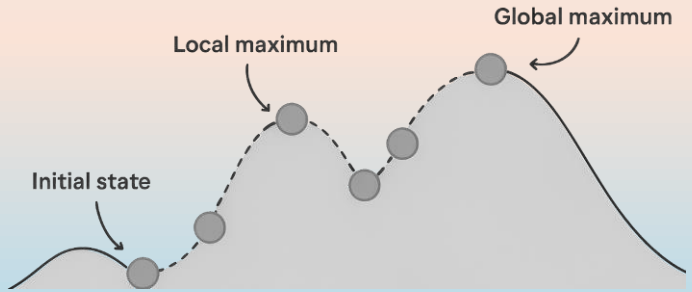
Pattern Search

- **Optimization**
  - ❖ **Main Types**
    - **Bayesian Optimization**: Uses probabilistic models to optimize expensive, black-box functions.
      - Gaussian Process Optimization
      - Tree-structured Parzen Estimator (TPE)



Gaussian Process Optimization

    - **Heuristics**: Problem-specific approaches using simple rules to provide quick, often suboptimal solutions.
      - Greedy Algorithm
      - Hill Climbing
      - Beam Search



Hill Climbing

- # Optimization
  - ## ❖ Main Types
    - **Metaheuristics**: General frameworks for global search, balancing exploration and exploitation, covering diverse problems. (**Here, we concentrate more on Metaheuristics**)
      - **Evolutionary Algorithms**: A Subset of metaheuristics inspired by biological evolution (Like Genetic Algorithms, Differential Evolution).
      - **Swarm Intelligence Algorithms**: A Subset of metaheuristics inspired by collective behaviors in nature (Like Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Victoria Amazonia Algorithm (VAO), and Ant Colony Optimization (ACO)).
      - **Physics-Based Optimization**: Inspired by physical processes like annealing or gravitation to find near-optimal solutions (such as Simulated Annealing (SA), Galaxy Gravity Optimization (GGO), and Harmony Search (HS)).
      - **Human-Based Algorithms**: Inspired by human decision-making and behavior (Like Teaching-Learning-Based Optimization (TLBO), Brain Storm Optimization (BSO), Tabu Search (TS), and Imperialist Competitive Algorithm) (ICA).

- **Optimization**
  - ❖ **Main Types**

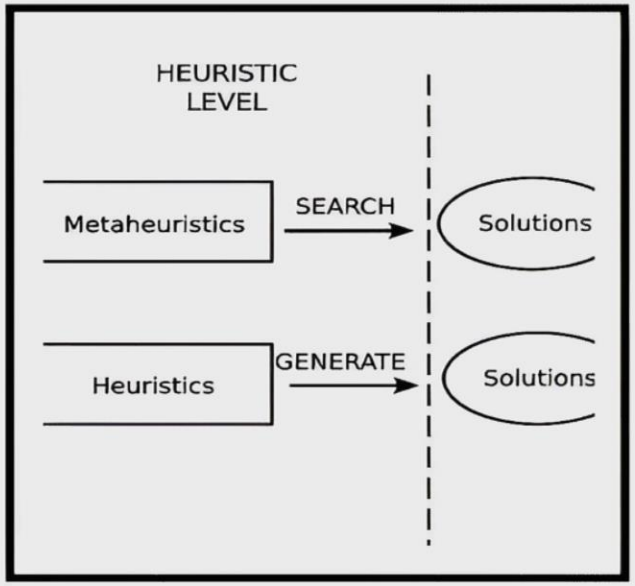
Evolutionary algorithms
- Genetic Algorithm
- Genetic programming
- Differential Evolution
- Evolutionary Strategy

Swarm-based algorithms
- Particle Swarm Optimization
- Ant Colony Optimization
- Firefly Algorithm
- Grey Wolf Optimization

Meta-heuristics

Human-based algorithms
- Tabu Search
- League Championship Algorithm
- Imperialistic Competitive Algorithm
- Exchange Market Algorithm

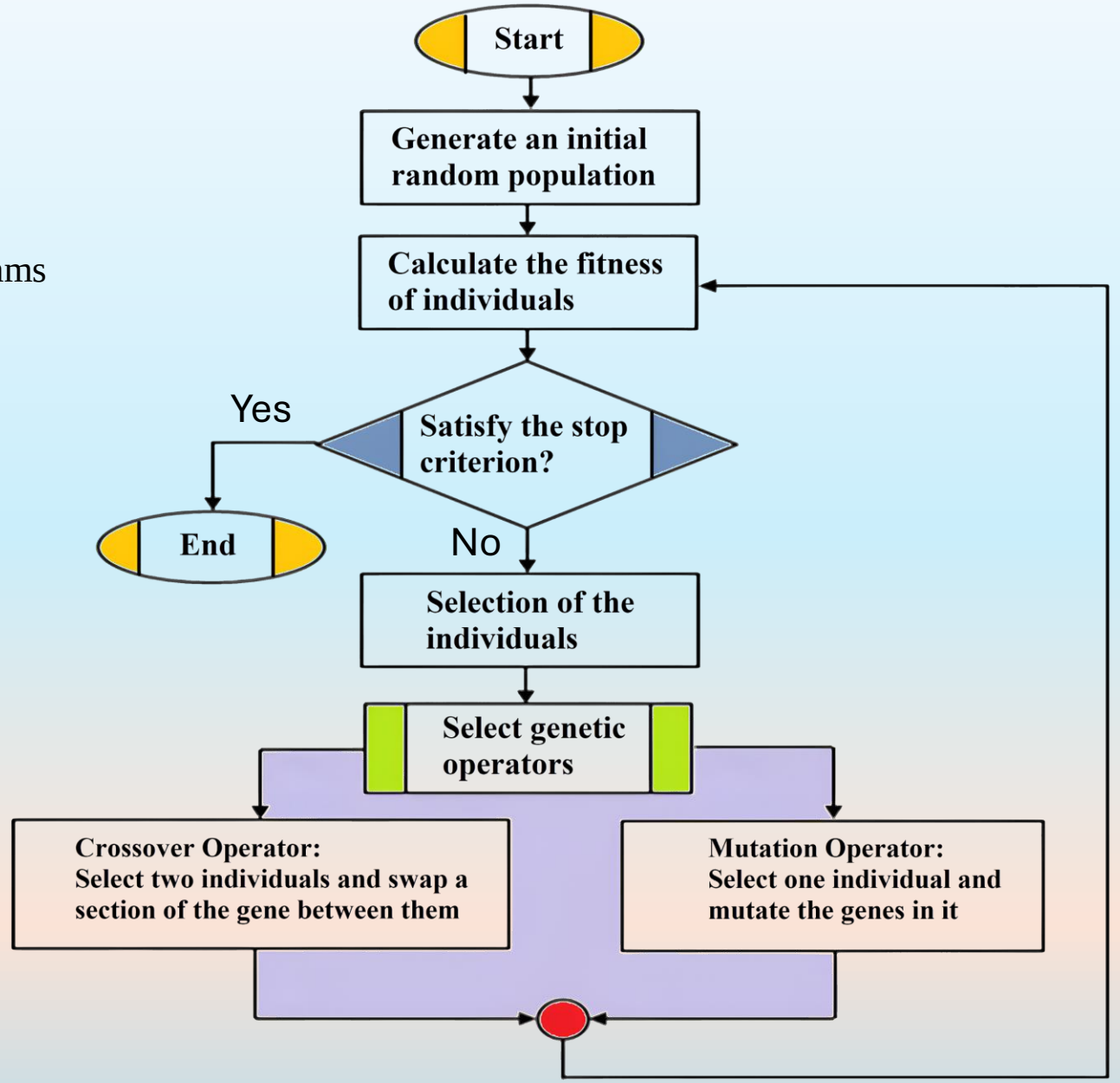
Physics-based algorithms
- Simulated Annealing
- Harmony Search Algorithm
- Gravitational Search Algorithm
- Artificial Chemical Reaction Optimization



Heuristics are often problem-dependent, that is, you define a heuristic for a given problem. Metaheuristics are problem-independent techniques that can be applied to a broad range of problems.

- **Optimization**
  - ❖ **Algorithms**
    - ○ **Metaheuristics**
      - Evolutionary Algorithms

Genetic Algorithm (GA): Mimics natural selection to evolve solutions to optimization problems.



Paper: Holland, John H. "Genetic algorithms." Scientific american 267.1 (1992): 66-73.

- **Optimization**
  - ❖ **Algorithms**
    - **Metaheuristics** - > Evolutionary Algorithms -> GA

Our objective is f(x) = Max (x).
Initialization:
Create a random population of 4 candidates:

Candidate	Binary	Decimal (Fitness)
A	1001	9
B	0110	6
C	1100	12
D	0101	5

Parents:
A = 1001
C = 1100
Swap after the 2nd bit:
A → 10|01
C → 11|00

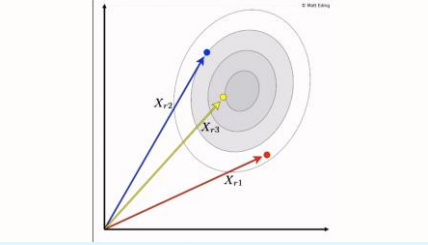
Children:
Child1 = 10|00 → 1000 (8)
Child2 = 11|01 → 1101 (13)
Mutation:
1000 → 1001
New Population: Now we replace the old population with new ones:

Candidate	Binary	Fitness
1	1001	9
2	1101	13
3	0110	6
4	0101	5

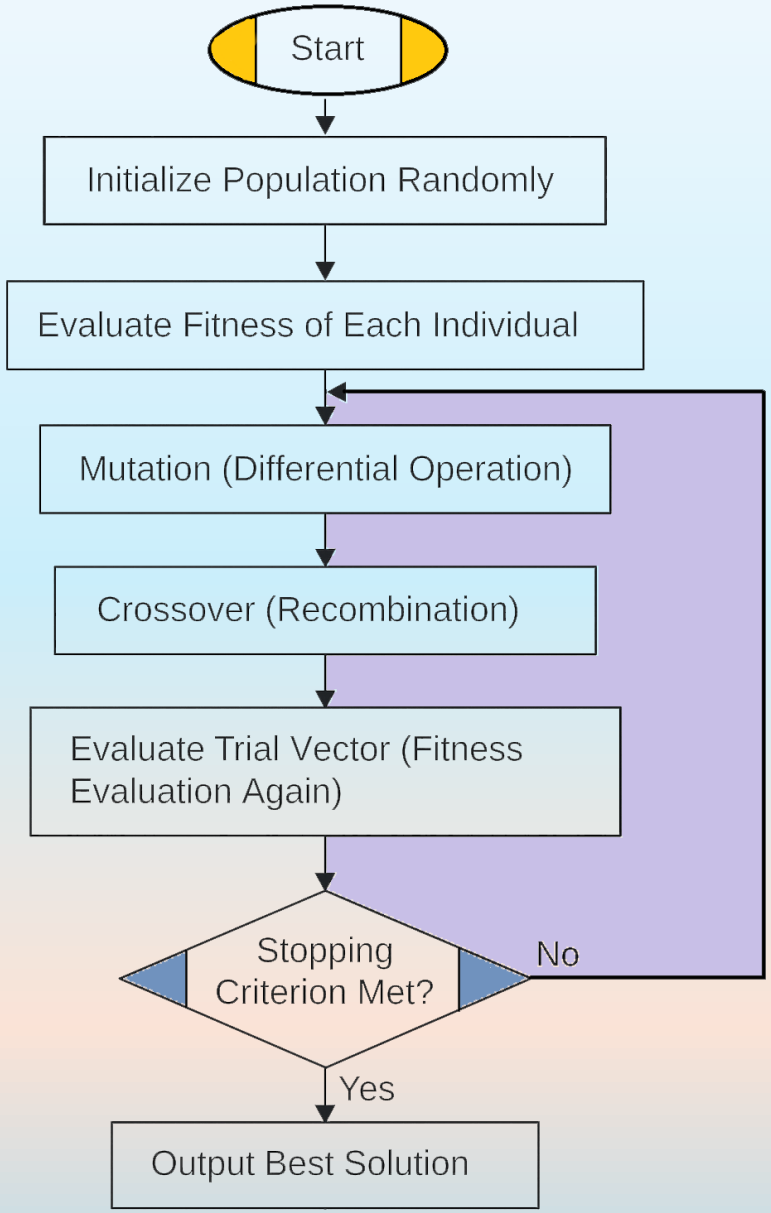
Repeat:
Keep repeating until the best solution is found: ideally, 1111 (fitness = 15) is the best solution.

Paper: Holland, John H. "Genetic algorithms." Scientific american 267.1 (1992): 66-73.

- **Optimization**
  - ❖ **Algorithms**
    - ○ **Metaheuristics**
      - Evolutionary Algorithms

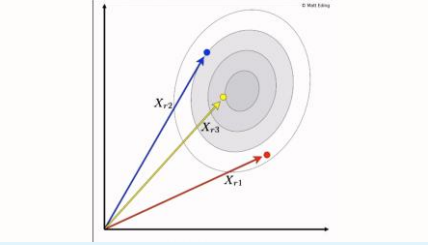


Differential Evolution (DE): Optimizes by evolving candidate solutions based on differences between them.



Paper: Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11.4 (1997): 341-359.

- **Optimization**
  - ❖ **Algorithms**
    - **Metaheuristics - >** Evolutionary Algorithms -> DE



Let's say we want to minimize $f(x)=x^2$.
Initialize population:
3 random solutions:
x1 = 5.2
x2 = -3.8
x3 = 2.1
We start with a random candidate:
x1 = 5.2
We calculate its fitness:
f(5.2) = (5.2)^2 = 27.04
Mutation - new candidate (mutant):
Pick three random ones, and create a mutant:

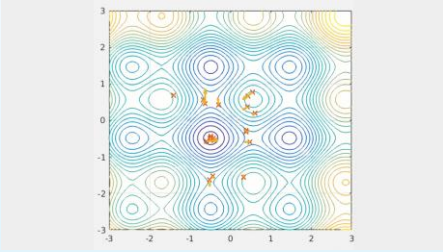
v = x1 + F * (x2 - x3)

= 5.2 + 0.8 * (-3.8 - 2.1)
= 5.2 - 4.72 = 0.48

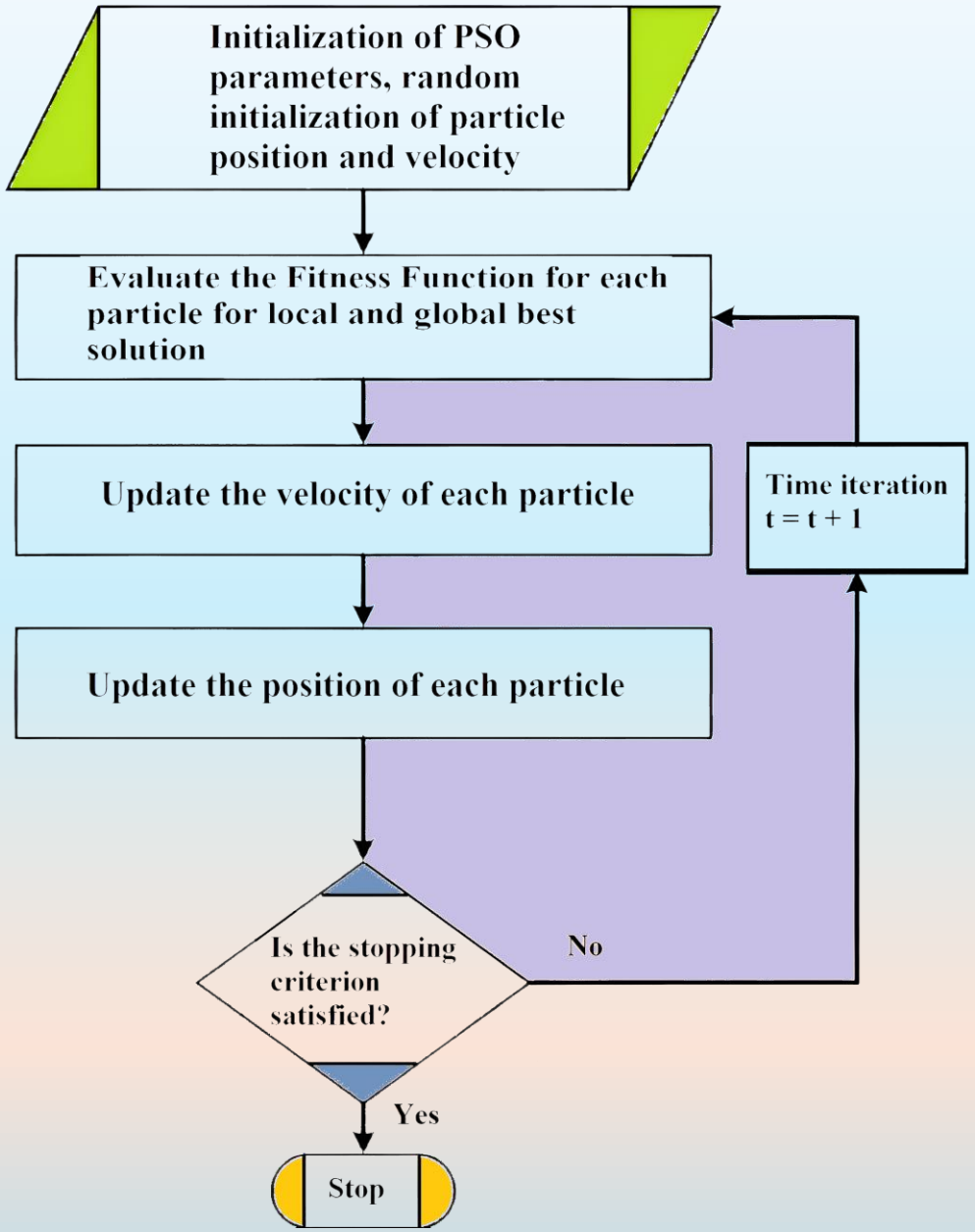
F is our constant parameter. Now is 0.8, here.
That equation is part of the official Differential Evolution algorithm.
Crossover:
Mix mutant v with original target x1.
Selection:
Compare fitness:
Now we compare which one is better:
Old: x1 = 5.2 → f(5.2) = 27.04
New: v = 0.48 → f(0.48) = 0.23
We keep 0.48 for the next generation as it is smaller.
So 0.48 replaces x1 or 5.2 in the next gen.
Then repeat, the numbers converge toward 0.

Paper: Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11.4 (1997): 341-359.

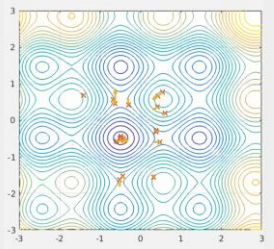
- **Optimization**
  - ❖ **Algorithms**
    - ○ **Metaheuristics**
      - Swarm Intelligence Algorithms



Particle Swarm Optimization (PSO): Simulates the social behavior of swarms to explore the solution space.



Paper: Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. ieee, 1995.



- **Optimization**
  - ❖ **Algorithms**
    - **Metaheuristics** - > Swarm Intelligence Algorithms -> PSO

We have 3 random particles (each is a possible solution):
Objective or fitness : F(x)=Min (x^2)
Particle 1: x = 5.0, velocity = 0.0
Particle 2: x = -3.0, velocity = 0.0
Particle 3: x = 2.0, velocity = 0.0
Compute fitness:
f(5) = 25, f(-3) = 9, f(2) = 4
Personal bests (pbest): same as starting positions.
Global best (gbest): smallest fitness → at x = 2.0.
Update **velocity**:
Each particle adjusts its **velocity** using this equation:

$$v_{new} = w \times v + c1 \times r1 \times (pbest - x) + c2 \times r2 \times (gbest - x)$$

Let's use simple constants for clarity:
w = 0.5 (inertia)
c1 = 1.0 (self-confidence)
c2 = 1.0 (swarm confidence)
r1 = r2 = 0.5 (random values)

For Particle 1 (x = 5.0):
v = 0.5*0 + 1*0.5*(5 - 5) + 1*0.5*(2 - 5)
v = 0 + 0 + 0.5 * (-3)
v = -1.5
Same For Particle 2 (x = -3.0): gives v = 2.5
Same for For Particle 3 (x = 2.0): gives v = 0
Update **position**: Each particle moves:

$$x_{new} = x + v$$

Particle 1: 5.0 + (-1.5) = 3.5
Particle 2: -3.0 + 2.5 = -0.5
Particle 3: 2.0 + 0.0 = 2.0
Evaluate fitness again:
f(3.5) = 12.25
f(-0.5) = 0.25
f(2.0) = 4.0
Update pbests: Particle 1: now 3.5 (better than)Particle 2: now -0.5 (better than -3.0)Particle 3: stays 2.0

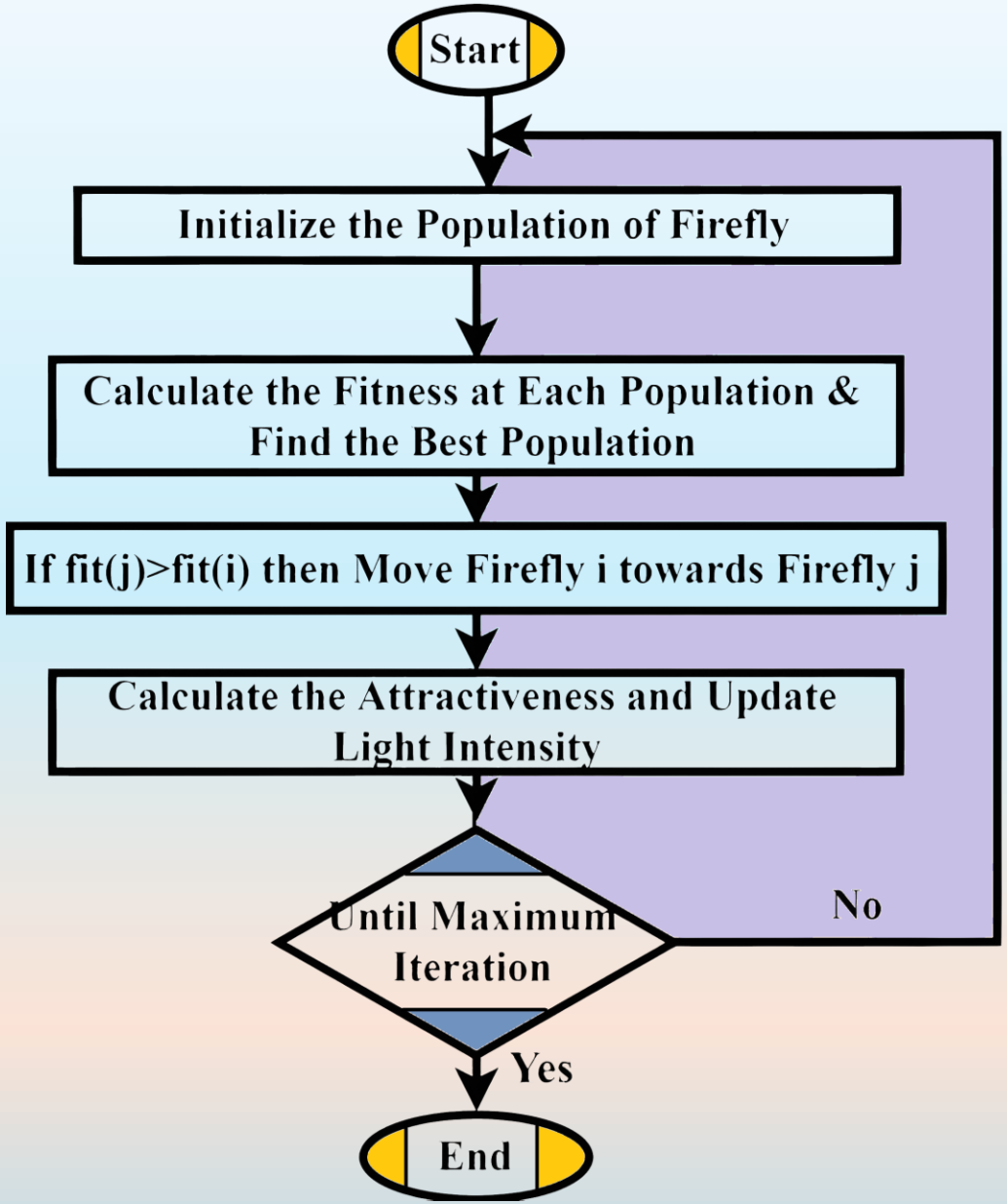
Update gbest:Best fitness = 0.25 → at x = -0.5
Particles will now move toward position x = -0.5, refining step-by-step until they all converge near 0

pbest = particle's own best
gbest = swarm's overall best

Paper: Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. ieee, 1995.

- **Optimization**
  - ❖ **Algorithms**
    - ○ **Metaheuristics**
      - Swarm Intelligence Algorithms

Firefly Algorithm (FA): Mimics the flashing behavior of fireflies, using light intensity to attract others and guide the search for optimal solutions.



Paper: Yang, Xin-She. "Firefly algorithms for multimodal optimization." International symposium on stochastic algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

- **Optimization**
  - ❖ **Algorithms**
    - **Metaheuristics** - > Swarm Intelligence Algorithms -> FA

Objective or fitness : F(x)=Min (x^2)
We have 3 fireflies (candidate solutions):
Firefly 1: x = 5.0
Firefly 2: x = -3.0
Firefly 3: x = 2.0
Evaluate fitness (brightness = here, lower is better):
f(5.0) = 25.0
f(-3.0) = 9.0
f(2.0) = 4.0
Use the Firefly movement equation (attractiveness):

$$xi=xi+\beta 0\cdot e^{-\gamma r2}\cdot(xj-xi)$$

Constant: $\beta_0 = 1$
Constant: $\gamma = 1$
Only move if another firefly is brighter (lower f(x))
Firefly 1 → moves toward Firefly 3

r = |5.0 - 2.0| = 3.0
β = e^(-3^2) = e^(-9) ≈ 0.000123
$x_1 = 5.0 + 0.000123 \times (2.0 - 5.0)$
$x_1 \approx 5.0 - 0.000369 = 4.99963$
Firefly 2 → moves toward Firefly 3
r = |-3.0 - 2.0| = 5.0
β = e^(-5^2) = e^(-25) ≈ 1.39e-11
$x_2 = -3.0 + 1.39e\text{-}11 \times (2.0 - (-3.0))$
$x_2 \approx -3.0 + 1.39e\text{-}11 \times 5 \approx$ -3.0 (no visible change)
Firefly 3 → no movement (already best)
Evaluate new positions that are new generation
Firefly 1: x = 4.99963 → f ≈ 24.996
Firefly 2: x ≈ -3.0 → f = 9.0
Firefly 3: x = 2.0 → f = **4.0**

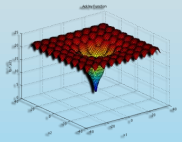
Paper: Yang, Xin-She. "Firefly algorithms for multimodal optimization." International symposium on stochastic algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

- # Optimization

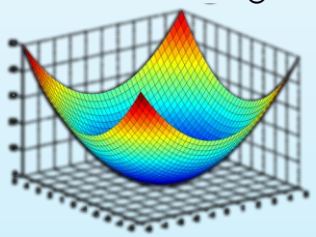
❖ Optimization Test Functions:

- Test functions in optimization are mathematical functions used to **evaluate and compare the performance** of optimization algorithms.
- These functions are carefully designed to represent **different kinds of challenges** that optimization algorithms might face, such as **non-convexity, multimodality (multiple local optima), or rugged landscapes**.
- They allow researchers to compare the performance of different optimization algorithms on the same problems **under specific conditions.**
- They **simulate real-world problem complexities** to analyze the robustness and adaptability of optimization techniques.
- They are generally categorized into the categories of unimodal and multimodal.
- Unimodal, which has a single global optimum and is used to test the convergence speed and accuracy of an algorithm such as sphere or Rosenbrock functions.
- Multimodal, which has **multiple local optima and at least one global optimum,** and is used to test an **algorithm's ability to escape local optima and find the global optimum**. Such as Ackley or Rastrigin functions.

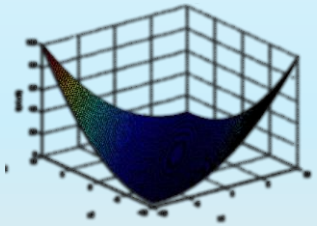
Helpful Link: https://www.sfu.ca/~ssurjano/optimization.html



- # Optimization
  - ## ❖ Optimization Test Functions:
    - **Unimodal convex/quadratic (simple bowl-shaped, only one minimum)**

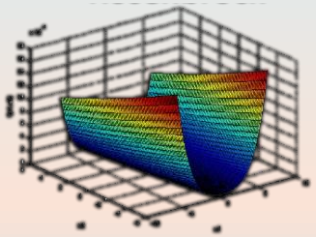


➢ **Sphere:** A simple quadratic function. It is a basic benchmark for testing an algorithm's **convergence speed.**

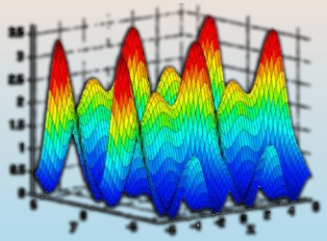


➢ **Matyas:** A symmetric, bowl-shaped function. Used to test optimization methods in **smooth, low-dimensional landscapes.**

- **Unimodal but non-convex / narrow valleys**

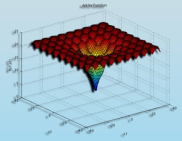


➢ **Rosenbrock:** A well-known function with a narrow valley leading to the global minimum. It tests an algorithm's **convergence efficiency**.



➢ **Powell:** A **complex function with flat regions**. Tests an algorithm’s performance in scenarios **where gradients are small**.

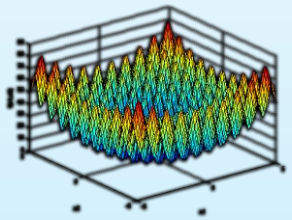
Helpful Link: https://www.sfu.ca/~ssurjano/optimization.html



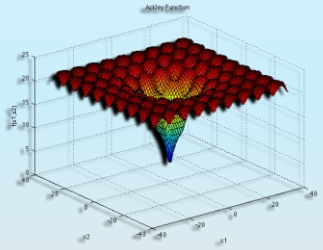
• Optimization

❖ Optimization Test Functions:

- **Multimodal (many local minima, periodic)**

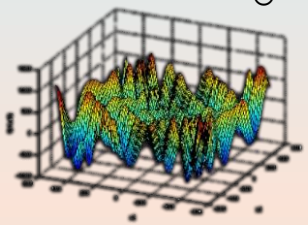


➢ **Rastrigin:** Contains **many regularly spaced local minima**. It is widely used to test optimization algorithms in **high-dimensional spaces**.

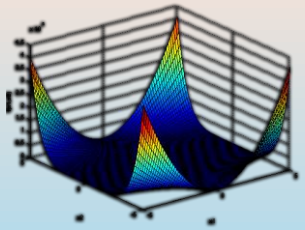


➢ **Ackley**: Features **many local minima but one global minimum**. It is used to test an algorithm's ability to converge to the global minimum in a noisy, multimodal landscape.

- **Highly irregular/complex landscapes (deceptive or rugged)**

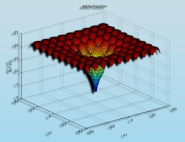


➢ **Egg holder:** A **highly non-convex**, multimodal function. Used to test **algorithms' exploration capabilities.**

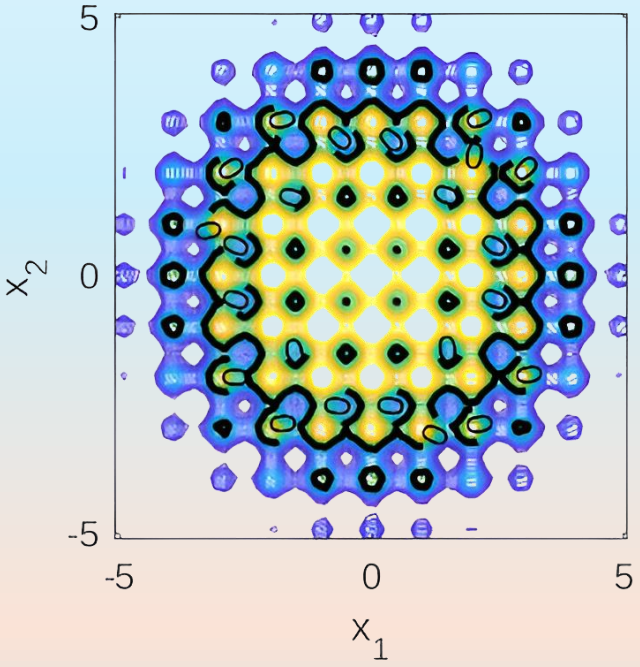


➢ **Beale**: A smooth function with a **single global minimum**. It evaluates an algorithm's performance in smooth, nonlinear spaces.

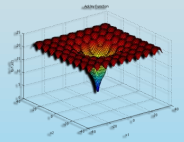
Helpful Link: https://www.sfu.ca/~ssurjano/optimization.html



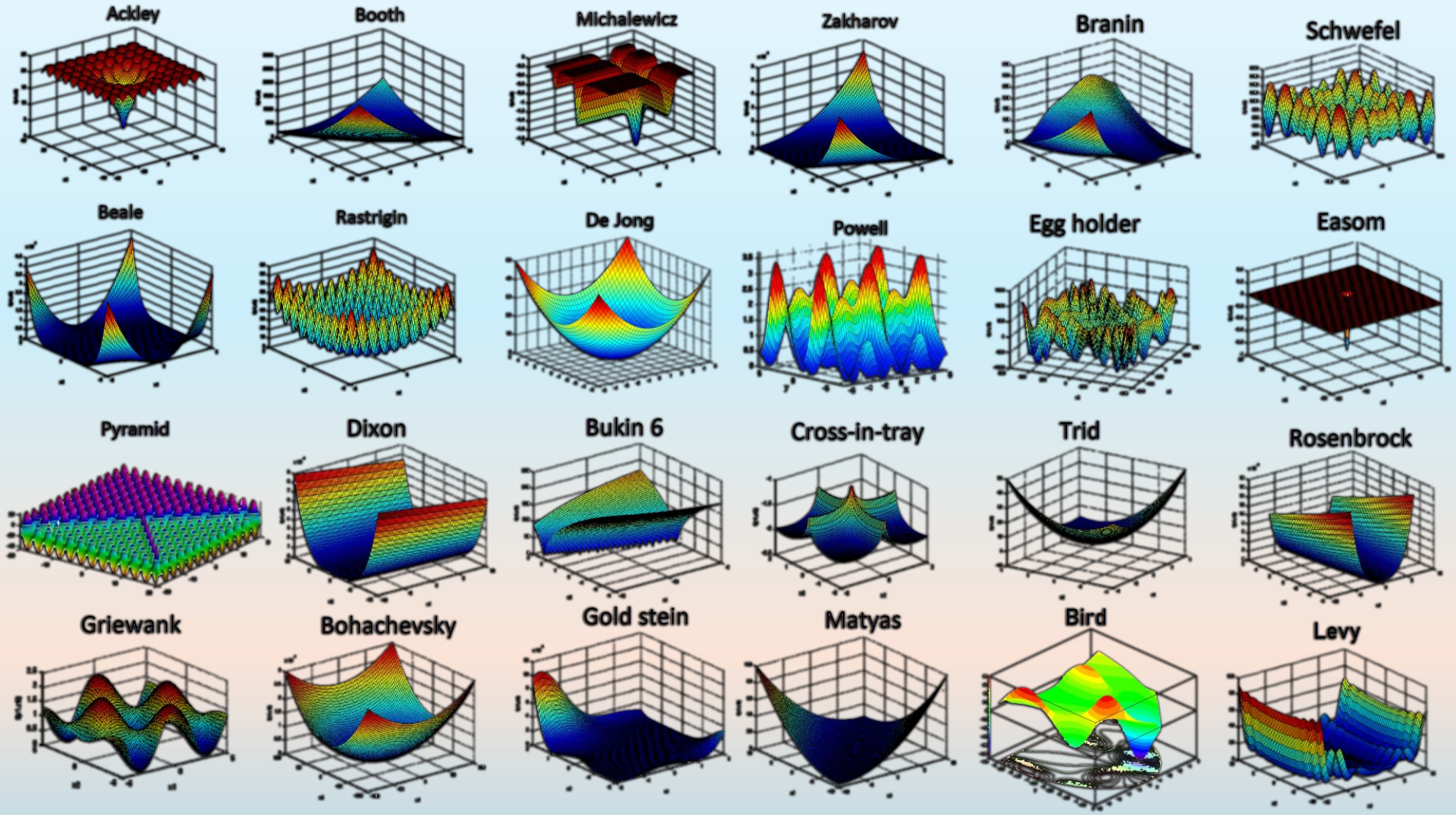
- **Optimization**
  - ❖ **Optimization Test Functions: FA on Rastrigin Function:**



Helpful Link: https://www.sfu.ca/~ssurjano/optimization.html



- **Optimization**
  - ❖ **Optimization Test Functions:**



Helpful Link: https://www.sfu.ca/~ssurjano/optimization.html