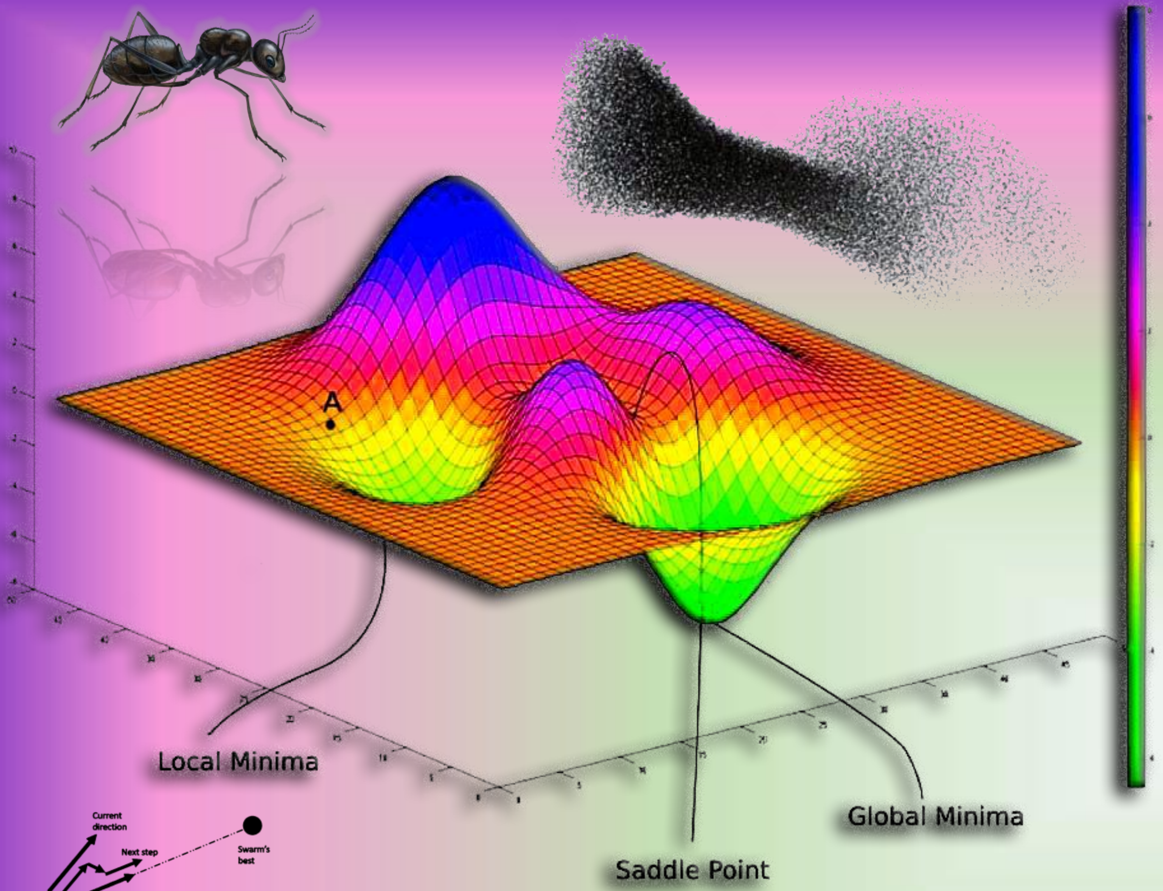
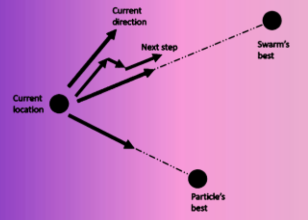
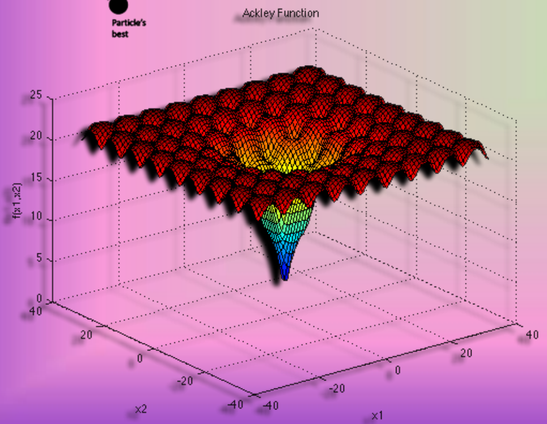
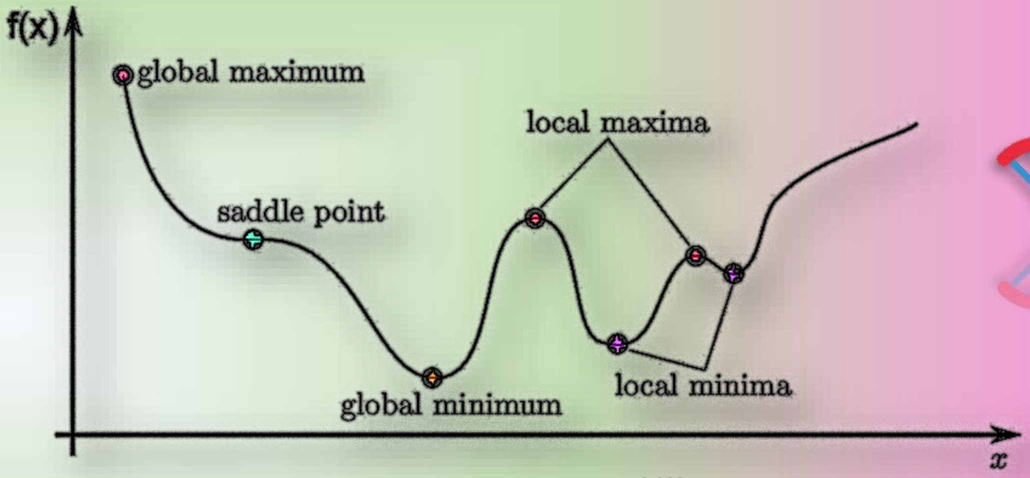
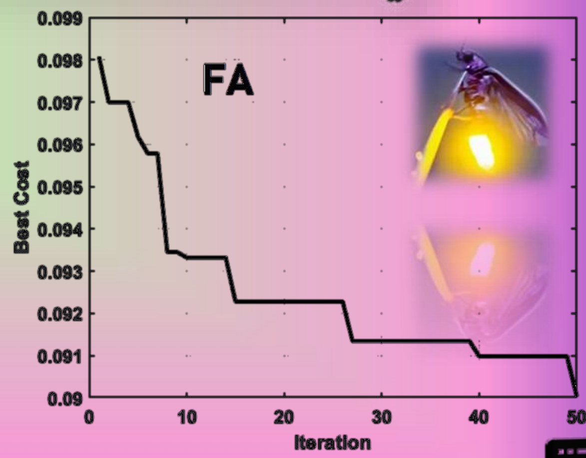






Complexity Example Results for Economic Dispatching Problem by Bess Algorithm:
Best Cost: 16202.85
Convergence time (seconds): 1.5400832176208496
Memory used (MB): 0.18390625
Number of operations performed: 2135
Complexity Class: O(n_scout_bees * max_iter * nPlant)
Complexity Name: Linearithmic (O(n * log n))



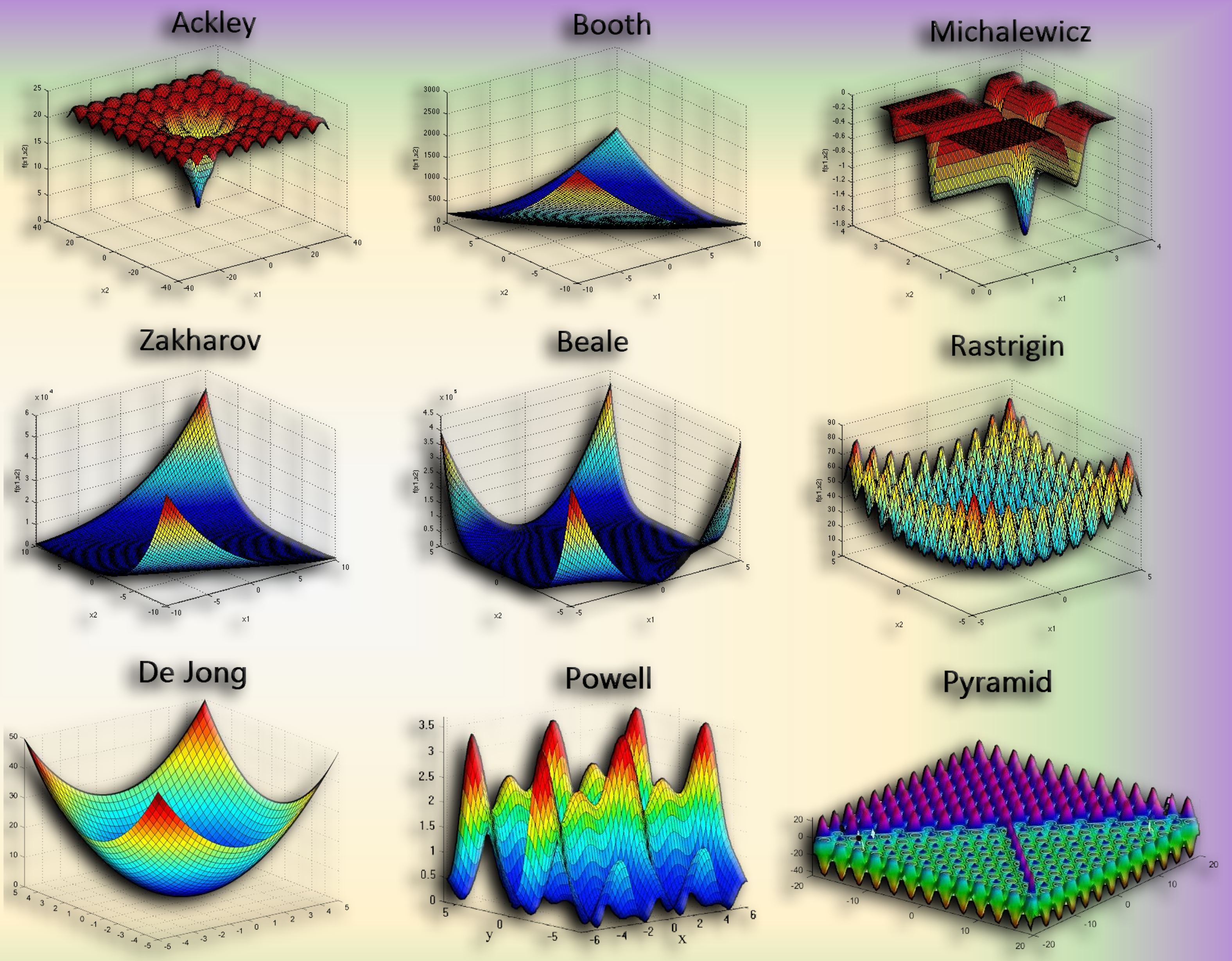


Optimization

By: Seyed Muhammad Hossein Mousavi
2024-2025

Outline:

- **Optimization**
  - ❖ **Definitions**
  - ❖ **Importance**
  - ❖ **Main Types**
  - ❖ **Algorithms**
  - ❖ **Optimization Test Functions**
  - ❖ **Optimization Problems**



• Optimization

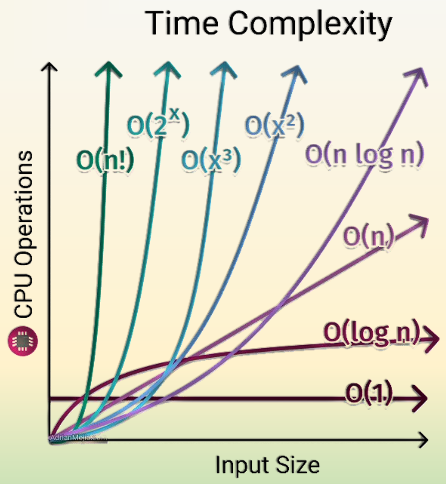
❖ Definitions

- Optimization is the process of **finding the best solution or outcome** from a **set of possible possibilities while satisfying given constraints**.
- It involves **maximizing or minimizing a specific objective function**, such as cost or fitness.
- 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.
- **Potential solutions** in optimization problems called **critical points.**
- Critical points occur where the derivative (gradient) of the objective function is zero or undefined.
- This means there is no slope, and the function "**flattens out**" **at these points**.
- 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.**

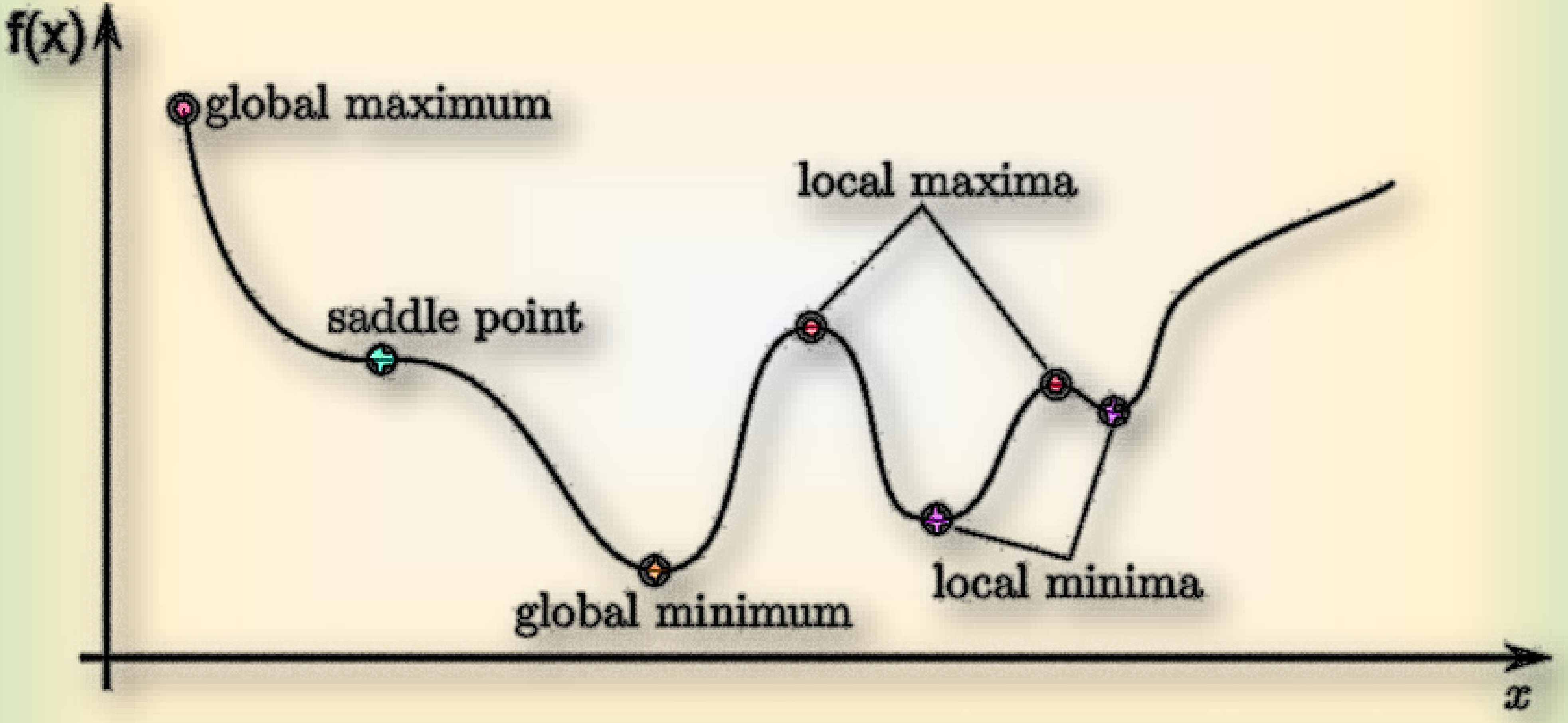
• Optimization

❖ Definitions

- 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.
- A **constraint** is a condition or rule that solutions must satisfy (e.g., budget limits, resource capacities).
- **Complexity** is the complexity of an algorithm is the amount of resources required to run the algorithm.
- It goes from the fastest/less complex of O(1) to the slowest/most complex of O(n!).

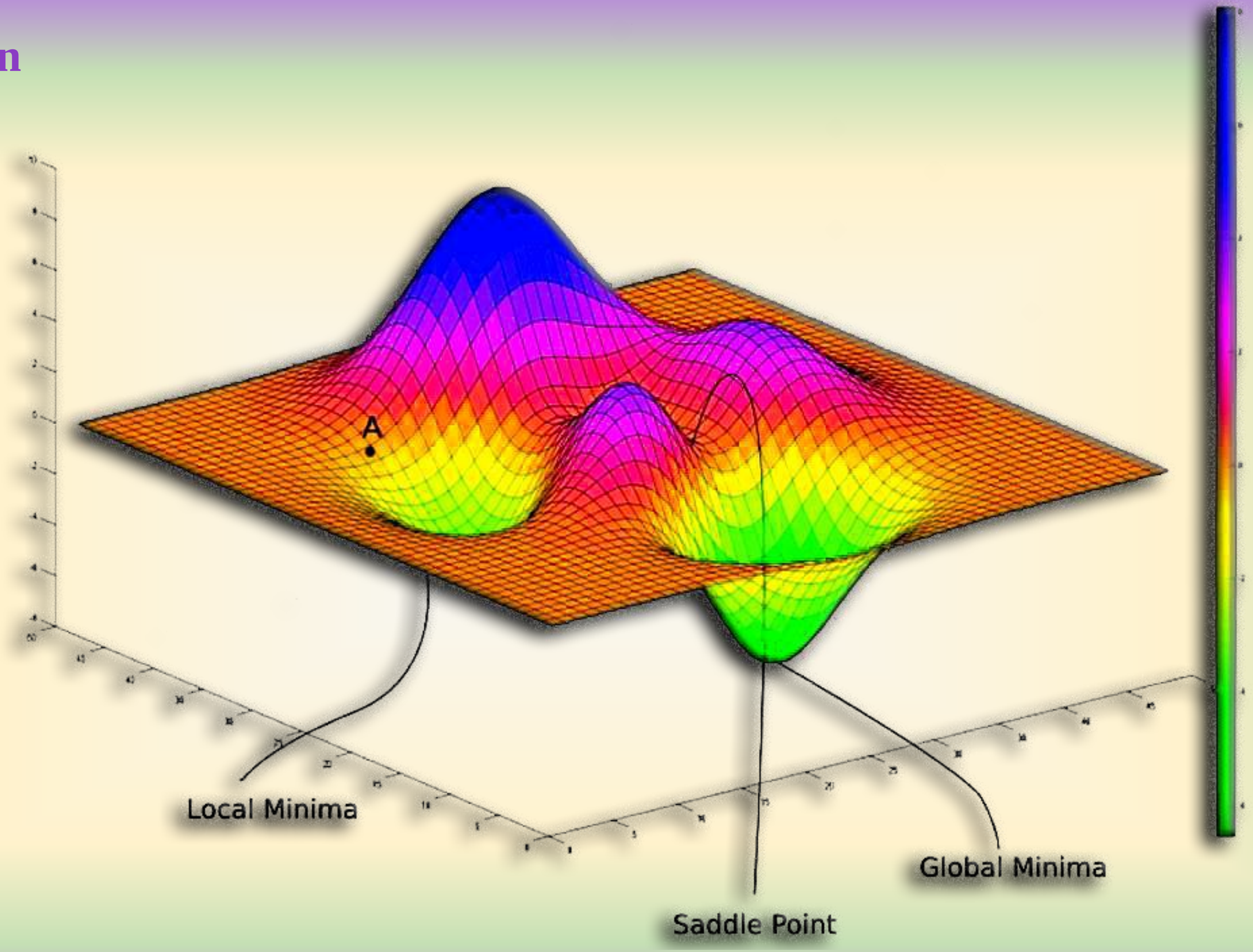


- # Optimization
  - ## Definition



Visualization of Critical Points in Optimization

- **Optimization**
  - **Definition**



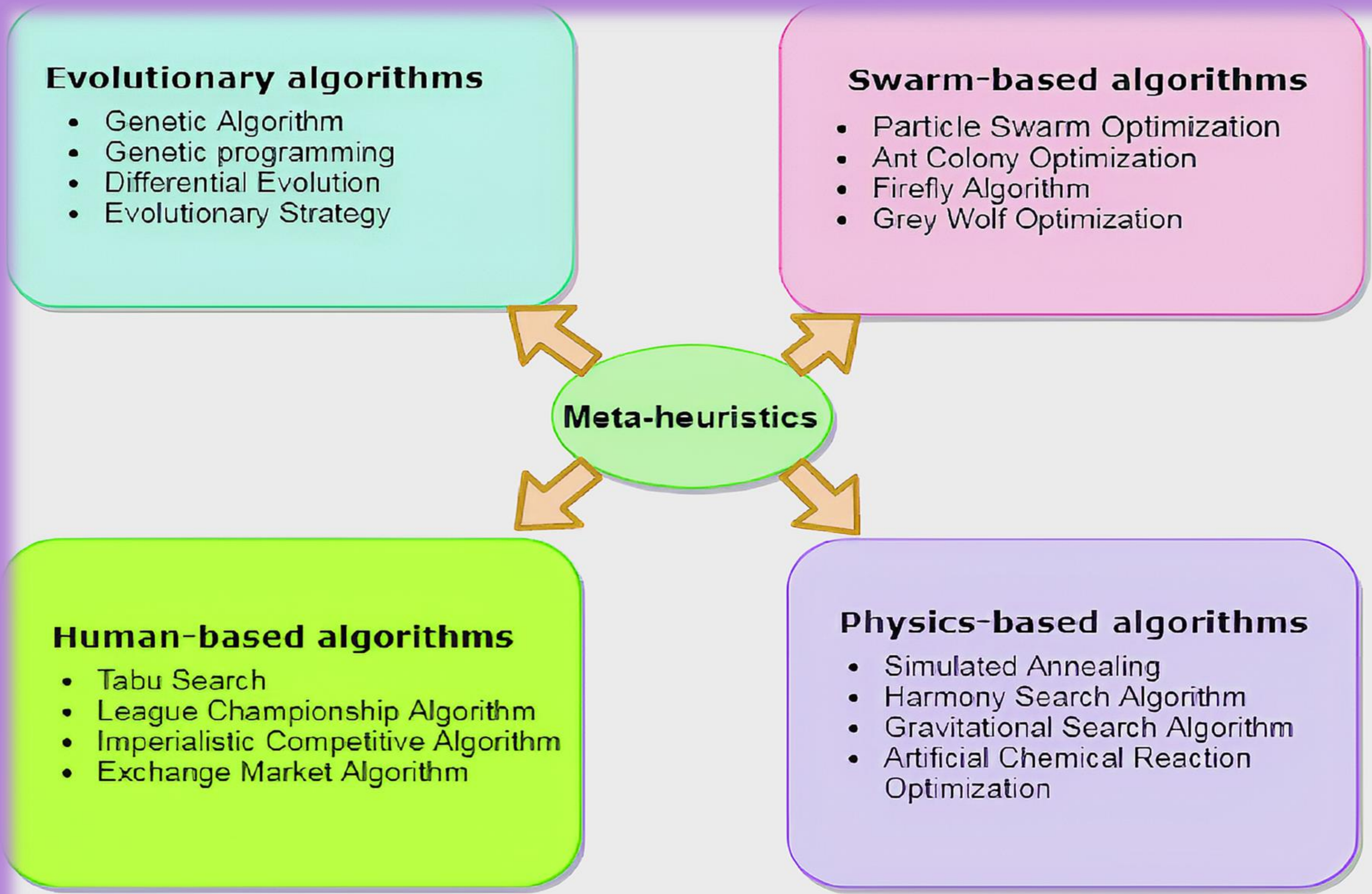
3-D Visualization of Critical Points in Optimization on a Test Function

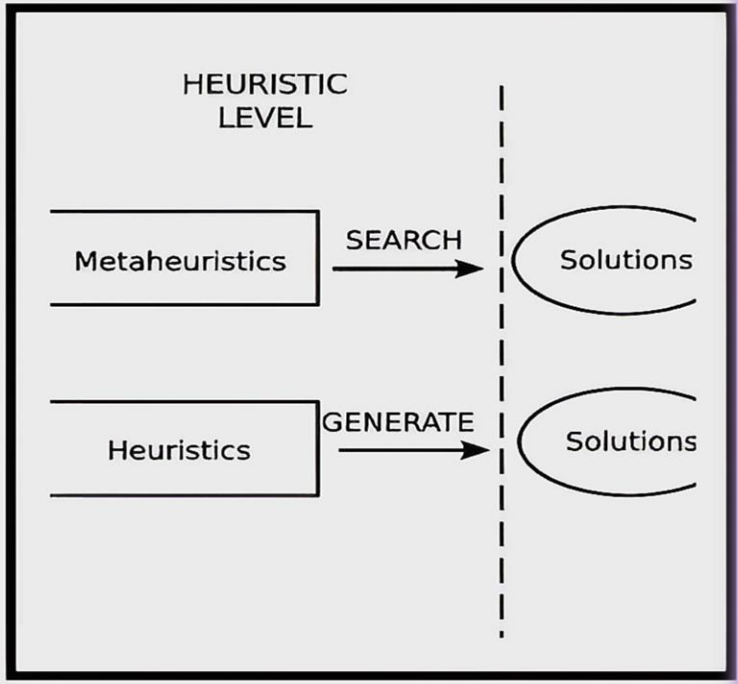
Optimization

Importance

- **Efficiency Improvement**: Optimization **minimizes resource usage (time, cost, energy) while achieving desired outcomes**, enabling processes to operate 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-Free Optimization**: Operates without gradient information, ideal for non-differentiable or noisy problems.
    - **Heuristics**: Problem-specific approaches using simple rules to provide quick, often suboptimal solutions.
    - **Metaheuristics**: General frameworks for global search, balancing exploration and exploitation, covering diverse problems. (**Here, we concentrate more on Metaheuristics**)
      - **Evolutionary Algorithms**: Subset of metaheuristics inspired by biological evolution (e.g., Genetic Algorithms, Differential Evolution).
      - **Swarm Intelligence Algorithms**: Subset of metaheuristics inspired by collective behaviors in nature (e.g., PSO, Firefly Algorithm, and Ant Colony Optimization).
      - **Physics-Based Optimization**: Inspired by physical processes like annealing or gravitation to find near-optimal solutions (such as Simulated Annealing or Harmony Search).
      - **Human-Based Algorithms**: Inspired by human decision-making and behavior (e.g., Teaching-Learning-Based Optimization, Brain Storm Optimization, Tabu Search, and Imperialist Competitive Algorithm).
    - **Bayesian Optimization**: Uses probabilistic models to optimize expensive, black-box functions.





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

- **Gradient-Based Optimization**
  - **Gradient Descent**: Iteratively moves in the direction of the steepest descent to minimize a function.
  - **Newton's Method**: Uses second-order derivatives for faster convergence in smooth problems.
  - **Stochastic Gradient Descent (SGD)**: A variant of Gradient Descent that updates weights using random subsets of data.
- **Gradient-Free Optimization**
  - **Nelder-Mead**: A simplex-based method for finding minima in non-differentiable problems.
  - **Powell's Method**: An iterative optimization method that does not require derivatives.
- **Heuristics**
  - **Hill Climbing**: A greedy algorithm that iteratively improves the solution by moving to better neighbors.
  - **Best-First Search**: Explores a search tree by selecting the node with the lowest estimated cost to the goal, prioritizing the most promising paths first.
- **Bayesian Optimization**
  - **Gaussian Process Optimization**: Builds probabilistic models to optimize expensive black-box functions.
  - **Tree-Structured Parzen Estimator (TPE):** A Bayesian method for hyperparameter tuning in machine learning.

- # Optimization
  - ## Algorithms
    - ### Metaheuristics
      - Evolutionary Algorithms
        - ✓ **Genetic Algorithm (GA):** Mimics natural selection to evolve solutions to optimization problems.
        - ✓ **Differential Evolution (DE):** Optimizes by evolving candidate solutions based on differences between them.
        - ✓ **NSGA-II (Non-dominated Sorting Genetic Algorithm-II):** A fast and elitist evolutionary algorithm for solving multi-objective optimization problems, maintaining diversity.
        - ✓ **Evolutionary Strategy (ES):** Inspired by biological evolution, focusing on mutation and selection to iteratively improve candidate solutions for real-valued optimization problems.
      - Swarm Intelligence Algorithms
        - ✓ **Particle Swarm Optimization (PSO):** Simulates social behavior of swarms to explore the solution space.
        - ✓ **Ant Colony Optimization (ACO):** Mimics the foraging behavior of ants to find optimal paths.
        - ✓ **Firefly Algorithm (FA):** Mimics the flashing behavior of fireflies, using light intensity to attract others and guide the search for optimal solutions.
        - ✓ **Gray Wolf Optimizer (GWO):** Mimics the hunting and leadership hierarchy of gray wolves to balance exploration and exploitation in finding optimal solutions.

- # Optimization
  - ## Algorithms
    - **Metaheuristics**
      - Swarm Intelligence Algorithms (continue)
        - ✓ **Victoria Amazonica Optimization (VAO):** Simulates the growth and leaf arrangement of the Victoria Amazonica water lily, utilizing its efficient spatial coverage to solve optimization problems effectively.
        - ✓ **Weevil Damage Optimization Algorithm (WDOA):** It is inspired by weevil behaviors—specifically their flying, snout, and damaging capabilities—designed to solve complex optimization problems by mimicking these natural processes.
        - ✓ **Bess Algorithm (BA):** Mimics the foraging behavior of bees to explore and exploit the solution space effectively.
        - ✓ **MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition):** Breaks multi-objective problems into simpler subproblems to optimize.
      - Human-Based Algorithms
        - ✓ **Teaching-Learning-Based Optimization (TLBO):** Models the teaching-learning process to refine solutions.
        - ✓ **Brain Storm Optimization (BSO):** Simulates human brainstorming for idea generation and refinement.

Optimization

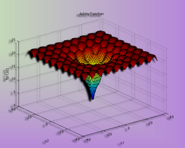
Algorithms

- **Metaheuristics**
  - Human-Based Algorithms (continue)
    - **Tabu Search (TS):** It uses memory structures to avoid revisiting previously explored solutions, enabling efficient exploration of the solution space.
    - **Imperialist Competitive Algorithm (ICA):** A socio-political metaheuristic inspired by imperialistic competition, where countries (solutions) compete to take over colonies and improve their fitness for global optimization problems.
  - Physics-Based Optimization
    - **Simulated Annealing (SA):** Mimics the annealing process in metals to escape local optima.
    - **Gravitational Search Algorithm (GSA):** Models the gravitational attraction between objects to find optima.
    - **Harmony Search (HS):** A music-inspired optimization algorithm that mimics the improvisation process of musicians to find the optimal solution by balancing exploration and exploitation.
    - **Galaxy Gravity Optimization (GGO):** A physics-inspired optimization algorithm that models the gravitational interactions within a galaxy to explore and converge toward optimal solutions effectively.

• Optimization

❖ Optimization Test Functions

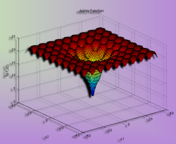
- Test functions in optimization are mathematical functions used to **evaluate and compare the performance** of optimization algorithms.
- These functions are carefully designed to represent **different kinds of challenges** that optimization algorithms might face, such as **non-convexity, multimodality (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 like 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.



Optimization

Optimization Test Functions

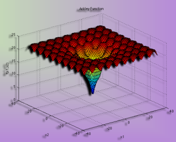
- **Ackley (Multimodal):** Features **many local minima but one global minimum**. It is used to test an algorithm's ability to converge to the global minimum in a **noisy, multimodal landscape**.
- **Booth (Unimodal):** A **simple function with a single global minimum**. Useful for testing optimization algorithms in a smooth and convex landscape.
- **Michalewicz (Multimodal):** Contains multiple local minima, making it **challenging for algorithms**. **Tests the ability to avoid local optima** and find the global minimum.
- **Zakharov (Unimodal):** A bowl-shaped function. It evaluates how efficiently an algorithm can handle linear and quadratic terms.
- **Branin (Multimodal):** A multimodal function with **three global minima**. Used for testing algorithms' exploration and exploitation balance.
- **Schwefel (Multimodal):** Features **many local minima**. Tests an algorithm's ability to **navigate through large search spaces with deceptive local minima**.
- **Beale (Unimodal):** A smooth function with a **single global minimum**. It evaluates an algorithm's performance in smooth, nonlinear spaces.



• Optimization

❖ Optimization Test Functions

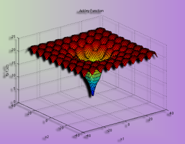
- **Rastrigin (Multimodal):** Contains **many regularly spaced local minima**. It is widely used to test optimization algorithms in **high-dimensional spaces**.
- **De Jong (Sphere) (Unimodal):** A simple quadratic function. It is a basic benchmark for testing an algorithm's **convergence speed.**
- **Powell (Unimodal):** A **complex function with flat regions**. Tests an algorithm's performance in scenarios **where gradients are small**.
- **Egg Holder (Multimodal):** A **highly non-convex**, multimodal function. Used to test **algorithms' exploration capabilities.**
- **Easom (Multimodal):** Has a **single sharp global minimum and is flat elsewhere**. It **challenges algorithms to find the minimum** in scattered information regions.
- **Pyramid (Multimodal):** Contains **many evenly spaced peaks**. Used to evaluate an **algorithm's global search capabilities.**
- **Dixon (Unimodal):** Tests the convergence behavior of optimization methods with quadratic features.
- **Bukin 6 (Multimodal):** Features **a narrow valley with many ridges**. Tests **algorithms' precision** and navigation in **constrained spaces**.
- **Cross-in-Tray (Multimodal): Highly complex** with **multiple minima**. Tests the **exploration** of rugged landscapes.

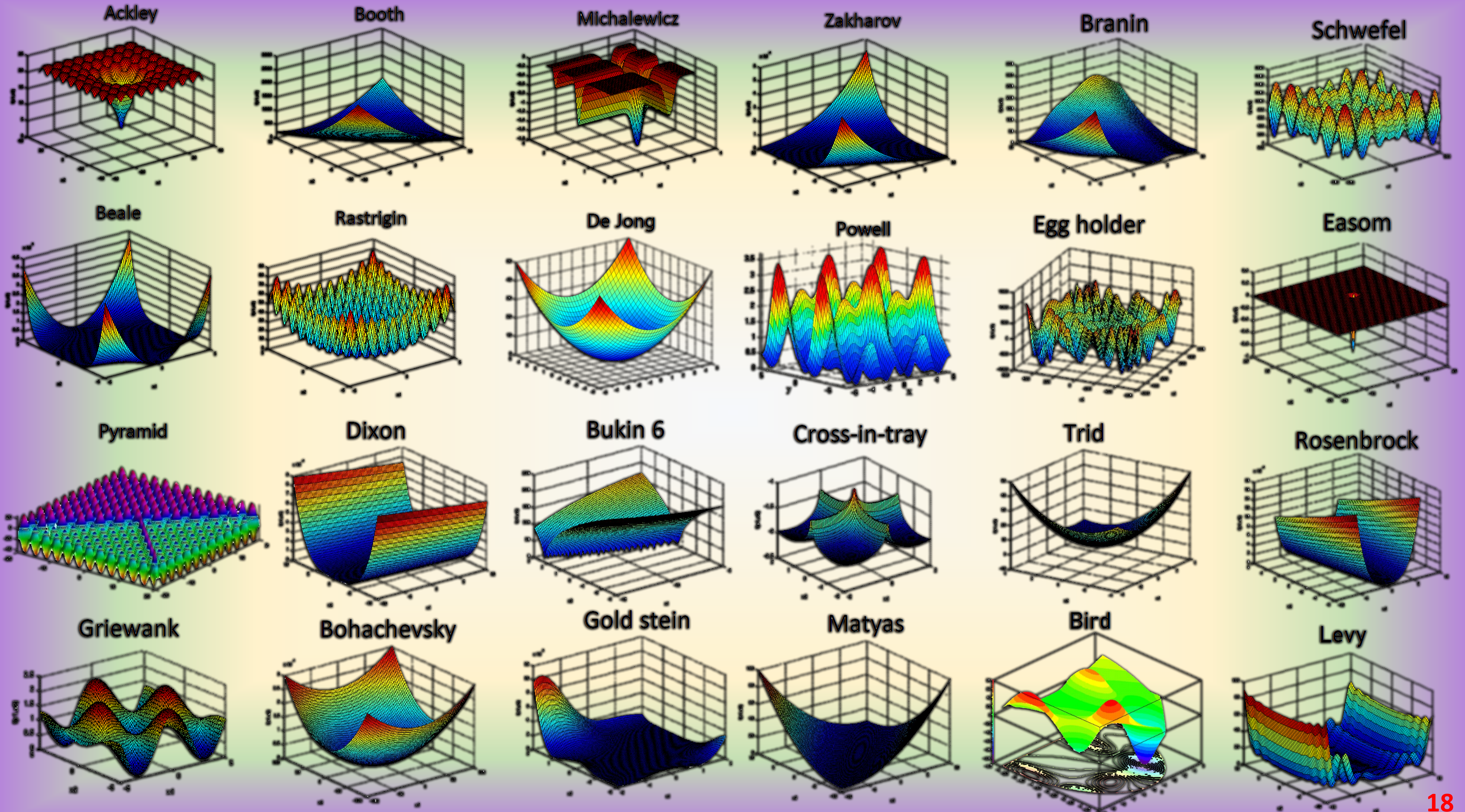


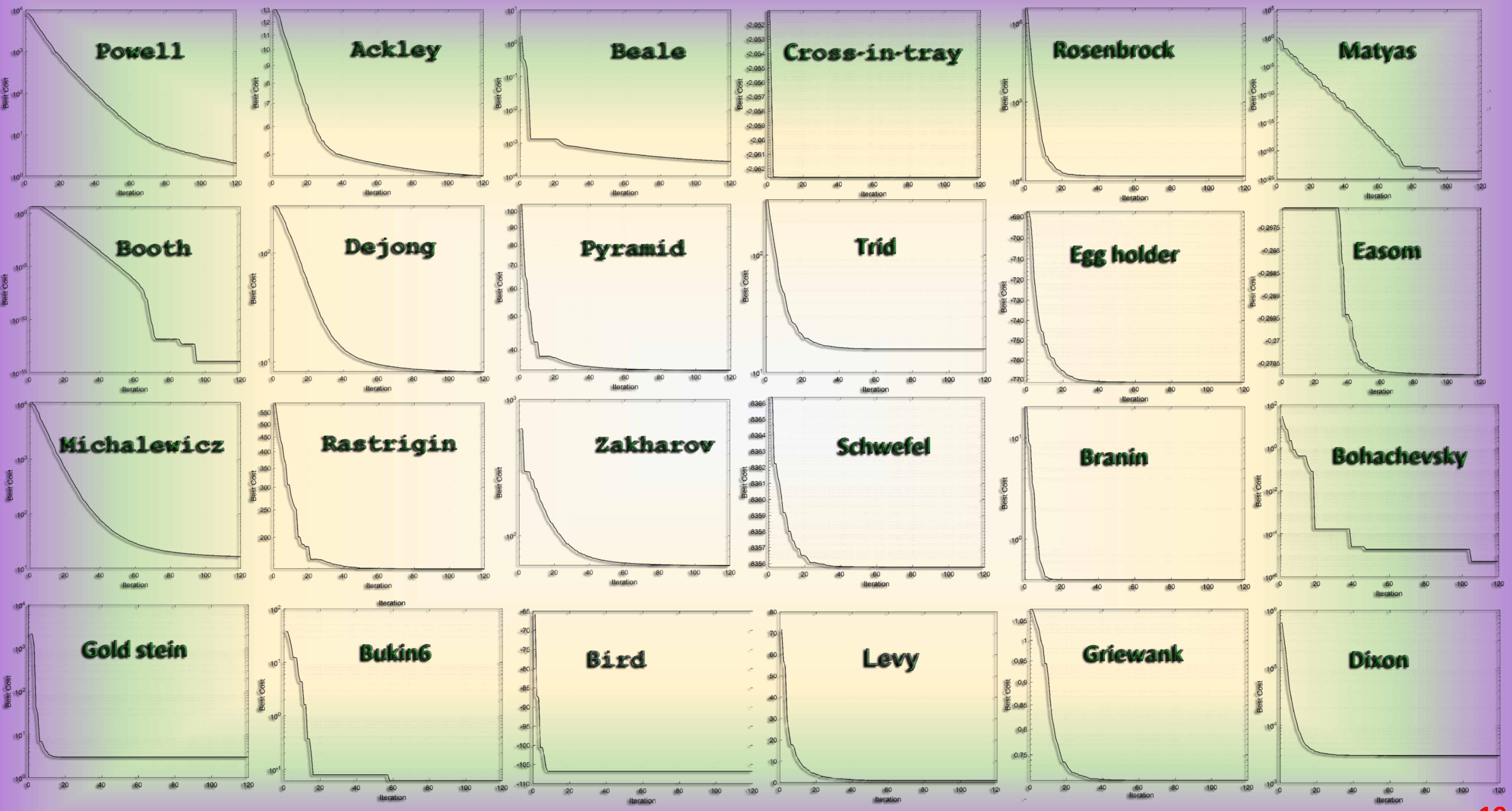
- # Optimization

❖ Optimization Test Functions

- **Trid (Unimodal):** A simple, bowl-shaped function. Useful for algorithms testing smooth landscapes.
- **Rosenbrock (Unimodal):** A well-known function with a narrow valley leading to the global minimum. It tests an algorithm's **convergence efficiency**.
- **Griewank (Multimodal):** Features many local minima and a single global minimum. It evaluates algorithms' **exploration in non-convex spaces**.
- **Bohachevsky (Unimodal):** A simpler variant of the Rastrigin function. Useful for testing convergence on smooth surfaces.
- **Goldstein-Price (Multimodal):** A function with **multiple local minima**. Used to test algorithms' robustness in **complex landscapes.**
- **Matyas (Unimodal):** A symmetric, bowl-shaped function. Used to test optimization methods in **smooth, low-dimensional landscapes.**
- **Bird (Multimodal):** Highly complex with multiple minima. Used to test algorithms’ ability to **explore rugged terrains**.
- **Levy (Multimodal):** Tests algorithms' global search capability with its **deceptive local minima**.




Ackley
Booth
Michalewicz
Zakharov
Branin
Schwefel
Beale
Rastrigin
De Jong
Powell
Egg holder
Easom
Pyramid
Dixon
Bukin 6
Cross-in-tray
Trid
Rosenbrock
Griewank
Bohachevsky
Gold stein
Matyas
Bird
Levy


Powell
Ackley
Beale
Cross-in-tray
Rosenbrock
Matyas
Booth
Dejong
Pyramid
Trid
Egg holder
Easom
Michalewicz
Rastrigin
Zakharov
Schwefel
Branin
Bohachevsky
Gold stein
Bukin6
Bird
Levy
Griewank
Dixon
Best Cost
Iteration

• Optimization

❖ Optimization Problems

- **Clustering**
  - **Problem**: Group data points into clusters where points in the same cluster are similar and points in different clusters are dissimilar.
  - **Default Methods**: Algorithms like k-means iteratively assign points to clusters based on simple distance metrics, but **they can get stuck in suboptimal solutions (local minima).**
  - **Optimization's Improvement**: Optimization algorithms **explore a larger solution space and avoid local optima by globally minimizing the clustering objective**, resulting in more robust clustering for complex or high-dimensional data.
- **Regression**
  - **Problem**: Predict a target variable by modeling the relationship between inputs and outputs.
  - **Default Methods**: Nonlinear regression uses techniques like Least Squares or Gradient Descent, but **they can get stuck in local minima or struggle with complex relationships**.
  - **Optimization Improvement**: Optimization algorithms **perform global searches, handle complex objective functions,** and provide more robust solutions for noisy or highly nonlinear data.

Optimization

Optimization Problems

- **Minimum Spanning Tree (MST)**
  - **Problem**: Find the subset of edges in a connected graph that connects all nodes with the minimum total weight without forming cycles.
  - **Default Methods**: Algorithms like Kruskal's and Prim's use greedy approaches to construct the MST but **may not efficiently handle dynamic changes or constraints in large, complex graphs.**
  - **Optimization's Improvement**: Global optimization approaches **can handle additional constraints** (e.g., specific edges must be included/excluded) and **adapt to dynamic or multi-objective scenarios** that default methods cannot.
- **Hub Location Allocation (HLA)**
  - **Problem**: Identify optimal hub locations and allocate demand points to these hubs to minimize transportation costs or maximize efficiency in a network.
  - **Default Methods**: Heuristic or greedy algorithms allocate hubs based on proximity or fixed cost models but may **struggle with complex demand patterns or additional constraints**.
  - **Optimization Improvement**: Global optimization approaches **can balance multiple objectives** (e.g., cost and time) and **incorporate complex constraints like hub capacity, dynamic demands, or multi-tier allocations**.

- # Optimization

❖ Optimization Problems

- **Traveling Salesman Problem (TSP)**
  - **Problem**: Determine the shortest possible route that visits each city exactly once and returns to the starting point.
  - **Default Methods**: Greedy approaches like Nearest Neighbor quickly find solutions but often **result in suboptimal routes, especially for larger problems.**
  - **Optimization Improvement**: Optimization methods **explore the global solution space**, **balancing exploration and exploitation to find near-optimal or globally optimal routes for complex, large-scale instances or multi-objective scenarios.**
- **Vehicle Routing Problem (VRP)**
  - **Problem**: Plan optimal routes for a fleet of vehicles to deliver goods to multiple locations while minimizing costs and meeting constraints like capacity and time windows.
  - **Default Methods**: Heuristic approaches, like Clarke-Wright Savings or nearest neighbor, provide quick solutions but may **fail to handle complex constraints effectively in large-scale or dynamic scenarios.**
  - **Optimization Improvement**: Global optimization methods can **incorporate multiple constraints (e.g., vehicle capacity, delivery windows) and objectives (e.g., cost, distance),** ensuring more efficient and scalable solutions for real-world applications.

- # Optimization
  - ## Optimization Problems
    - **Economic Dispatching (ED)**
      - **Problem**: Determine the optimal power output of multiple generators to meet electricity demand while minimizing total generation costs.
      - **Default Methods**: Traditional methods like Lambda-Iteration or Gradient Descent solve linear cost models but **struggle with nonlinear cost functions or constraints like generator limits and transmission losses**.
      - **Optimization Improvement**: Advanced approaches **can handle complex nonlinearities, multiple objectives (e.g., cost and emissions), and real-world constraints**, leading to more precise and efficient dispatch solutions.
    - **Image Segmentation**
      - **Problem**: Partition an image into distinct regions (e.g., separating objects from the background) to extract meaningful information for tasks like object detection or medical imaging.
      - **Default Methods**: Techniques like thresholding or edge detection rely on basic intensity differences, which often **fail in noisy or complex images with overlapping textures**.
      - **Optimization Improvement**: Optimization-based methods globally minimize cost functions that **balance region similarity and boundary smoothness**, leading to more accurate and robust segmentation for challenging datasets.

• Optimization

❖ Optimization Problems

- **Feature Selection**
  - **Problem**: Identify the most relevant subset of features from a dataset to improve model performance while reducing complexity and overfitting.
  - **Default Methods**: Techniques like correlation-based filtering or forward selection evaluate features individually or sequentially but may **miss interactions between features or subsets**.
  - **Optimization Improvement**: Global optimization approaches **explore the full search space to identify feature combinations that maximize model performance while adhering to constraints, such as limiting feature count or minimizing redundancy.**
- **Bin Packing**
  - **Problem**: Assign items of varying sizes to a fixed number of bins in a way that minimizes the number of bins used or maximizes space utilization.
  - **Default Methods**: Greedy algorithms like First-Fit or Best-Fit assign items sequentially, often **leading to suboptimal packing configurations.**
  - **Optimization Improvement**: **Global optimization approaches consider the overall packing arrangement, balancing space utilization and constraints like item priority or weight limits, leading to more efficient and feasible solutions.**

• Optimization

❖ Optimization Problems

- Quadratic Assignment Problem (QAP)
  - **Problem**: Assign a set of facilities to a set of locations to minimize the total cost, where costs depend on the flow between facilities and the distance between locations.
  - **Default Methods**: Heuristics or greedy algorithms provide quick solutions but often **fail to find near-optimal assignments in large or highly interconnected problems**.
  - **Optimization Improvement**: Global optimization approaches **efficiently explore the solution space, handling complex interactions between facilities and locations** to achieve more accurate and cost-effective assignments.
- Parallel Machine Scheduling (PMS)
  - **Problem**: Assign jobs to multiple parallel machines to minimize the total completion time, makespan, or other objectives while considering constraints like machine capacity or job priorities.
  - **Default Methods**: Heuristic algorithms, such as Shortest Processing Time (SPT) or Earliest Due Date (EDD), provide quick but often **suboptimal schedules**.
  - **Optimization Improvement**: Global optimization approaches **explore the solution space more effectively**, **balancing job allocation and machine utilization under complex constraints** to achieve near-optimal schedules.

Optimization

Optimization Problems

Resource Allocation

- **Problem**: Distribute limited resources among competing activities to maximize efficiency or achieve specific objectives while meeting constraints.
- **Default Methods**: Simple rule-based approaches or greedy algorithms allocate resources based on predefined priorities but often **fail to account for complex dependencies or trade-offs**.
- **Optimization Improvement**: Advanced approaches **explore the allocation space globally**, **balancing competing objectives and constraints (e.g., costs, time, or capacity)** for more efficient and effective resource utilization.

- # Optimization

❖ Optimization Problems

- **Evolved Antenna**
  - **Problem**: Design an antenna with specific performance characteristics (e.g., frequency range, signal strength) while adhering to physical and operational constraints.
  - **Default Methods**: Conventional design **relies on predefined templates and manual adjustments**, often requiring significant trial and error to meet complex requirements.
  - **Optimization Improvement**: Optimization approaches automate the design process, **exploring a wide range of configurations** to evolve novel and efficient antenna structures that **outperform manually designed counterparts**.
- **Space-Time Bending**
  - **Problem**: Design and optimize spacetime configurations for specific purposes, such as simulating gravitational effects, modeling relativistic phenomena, or exploring advanced physics scenarios.
  - **Default Methods**: Traditional approaches rely on fixed theoretical models or numerical approximations, which can be **computationally intensive and may not adapt well to complex constraints**.
  - **Optimization Improvement**: Advanced optimization methods **explore a broader range of spacetime geometries, dynamically balancing constraints like energy, mass distribution, and curvature to achieve more accurate and efficient models for specific applications.**

• Optimization

❖ Optimization Problems

- **Protein Structure Prediction (PSP)**
  - **Problem**: Predict the 3D structure of a protein from its amino acid sequence to understand its function and interactions.
  - **Default Methods**: Rule-based methods or sequence alignment approaches rely on known templates or heuristics, which often **fail for novel or complex proteins without similar known structures**.
  - **Optimization Improvement**: Optimization approaches explore the **vast conformational space, balancing energy minimization and structural constraints to predict more accurate and stable 3D structures, even for novel proteins**.
- **Exoplanetary Adaptation**
  - **Problem**: Adapt human genetics for survival on exoplanets with extreme environmental conditions.
  - **Default Methods**: Earth-centric models fail to address unique exoplanetary challenges like high radiation or low oxygen
  - **Optimization Improvement**: Explore genetic profiles in complex environments to predict adaptive traits, maximizing survival and functionality

Optimization

Optimization Problems

CNN Optimized (Weights and biases)

- **Problem**: Optimize the weights and biases of Convolutional Neural Networks (CNNs) to enhance accuracy and generalization across tasks.
- **Default Methods**: Gradient-based optimizers (e.g., SGD, Adam) often converge to suboptimal solutions, especially for complex or high-dimensional problems.
- **Optimization Improvement**: Advanced optimization algorithms explore the weight space more effectively, overcoming local minima and ensuring better performance and robustness.

Variational Auto Encoders (VAE) Optimized (Latent Space)

- **Problem**: Optimize the latent space of VAEs to improve the quality of reconstruction and generative performance.
- **Default Methods**: Standard training often results in poorly disentangled latent spaces, limiting representation accuracy and generation diversity.
- **Optimization Improvement**: Advanced optimization techniques enhance latent space structure, ensuring better disentanglement, reconstruction fidelity, and meaningful data generation.