Algorithms for Data Science

Optimization: Heuristic and Metaheuristic Algorithms

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Heuristic and Metaheuristic Algorithms

Heuristic Algorithms

Strategies designed to find approximate solutions efficiently for complex optimization problems, focusing on problem-specific rules to simplify the search process.

Metaheuristic Algorithms

Higher-level frameworks guiding heuristics to explore the solution space that are capable of escaping local optima and explore global solutions.

- Scalability
- Flexibility
- No Guarantee of Optimality

Simulated Annealing

Based on the annealing process in metallurgy, where a material is heated and cooled slowly to minimize energy states.

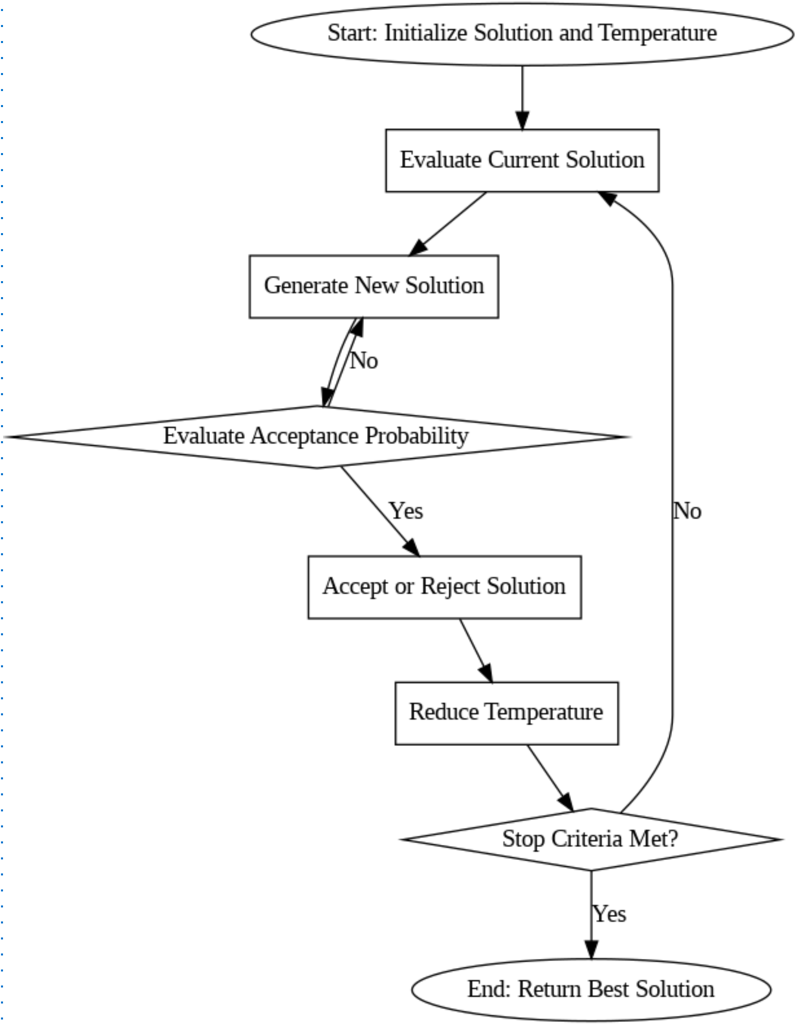
How it Works

- Start with an initial solution and a high "temperature"
- At each step, randomly modify the solution (exploration).
- Accept worse solutions with a probability: $P=e^{-\Delta E/T}$
  - ΔE: Change in objective function value
  - T: Current Temperature
- Gradually lower the temperature to refine the solution.

Algorithm

1. Initialize with a random solution and a high temperature.
2. Generate a neighbor solution.
3. Evaluate the change in objective function value (ΔE).
4. Accept the neighbor based on P.
5. Repeat until the system "freezes."

Simulated Annealing (cont.)

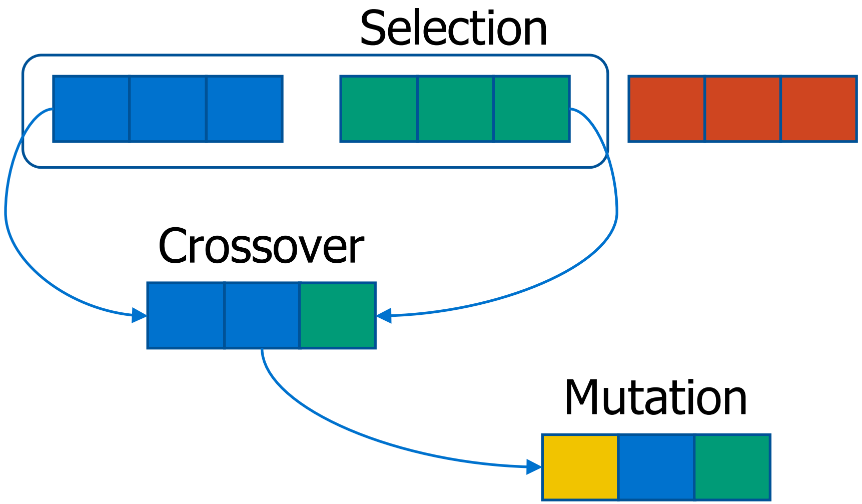


Genetic Algorithms

Based on the process of natural selection and evolution, mimicking how populations evolve over generations to adapt to their environment.

How it Works

- Start with an initial population of solutions.
- Use evolutionary operations:
  - Selection: Choose the fittest solutions based on their fitness score.
  - Crossover: Combine pairs of solutions (parents) to create new solutions.
  - Mutation: Introduce random changes to maintain diversity.
- Iterate over multiple generations to evolve better solutions.

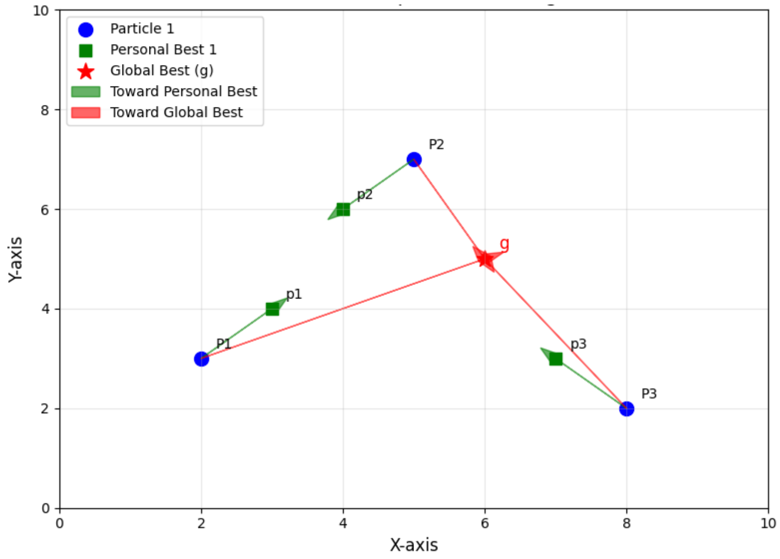


Particle Swarm Optimization (PSO)

Modeled after the collective behavior of swarms (such as birds or fish), each particle represents a potential solutions that "move" through the solution space by sharing information with their neighbors.

How it Works

- **Particles:** Each particle represents a candidate solution.
- **Velocity Update:** Particles adjust their velocity based on:
    - Their own best-known position (p_i)
    - The global best position (g)
    - Random exploration factors
- **Position Update:** Particles update their position based on the velocity.

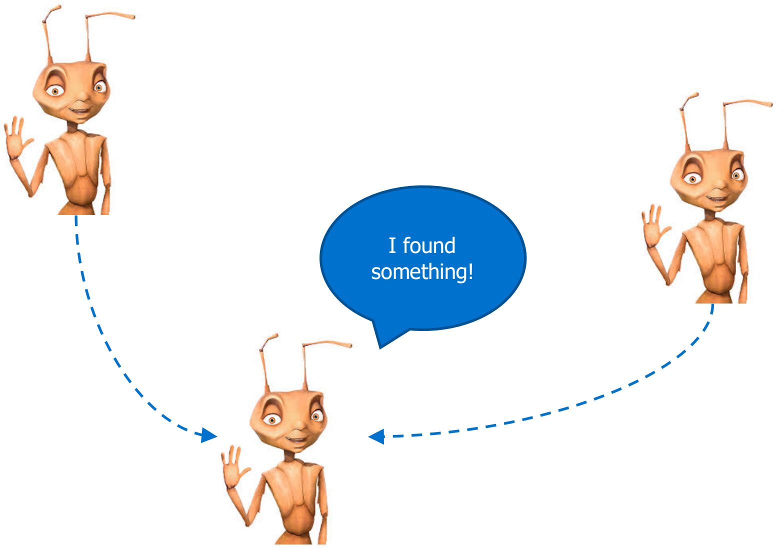


Ant Colony Optimization (ACO)

Modeled after the foraging behavior of ants, where ants deposit pheromones on paths as they search for food, influencing other ants to follow and reinforce successful routes.

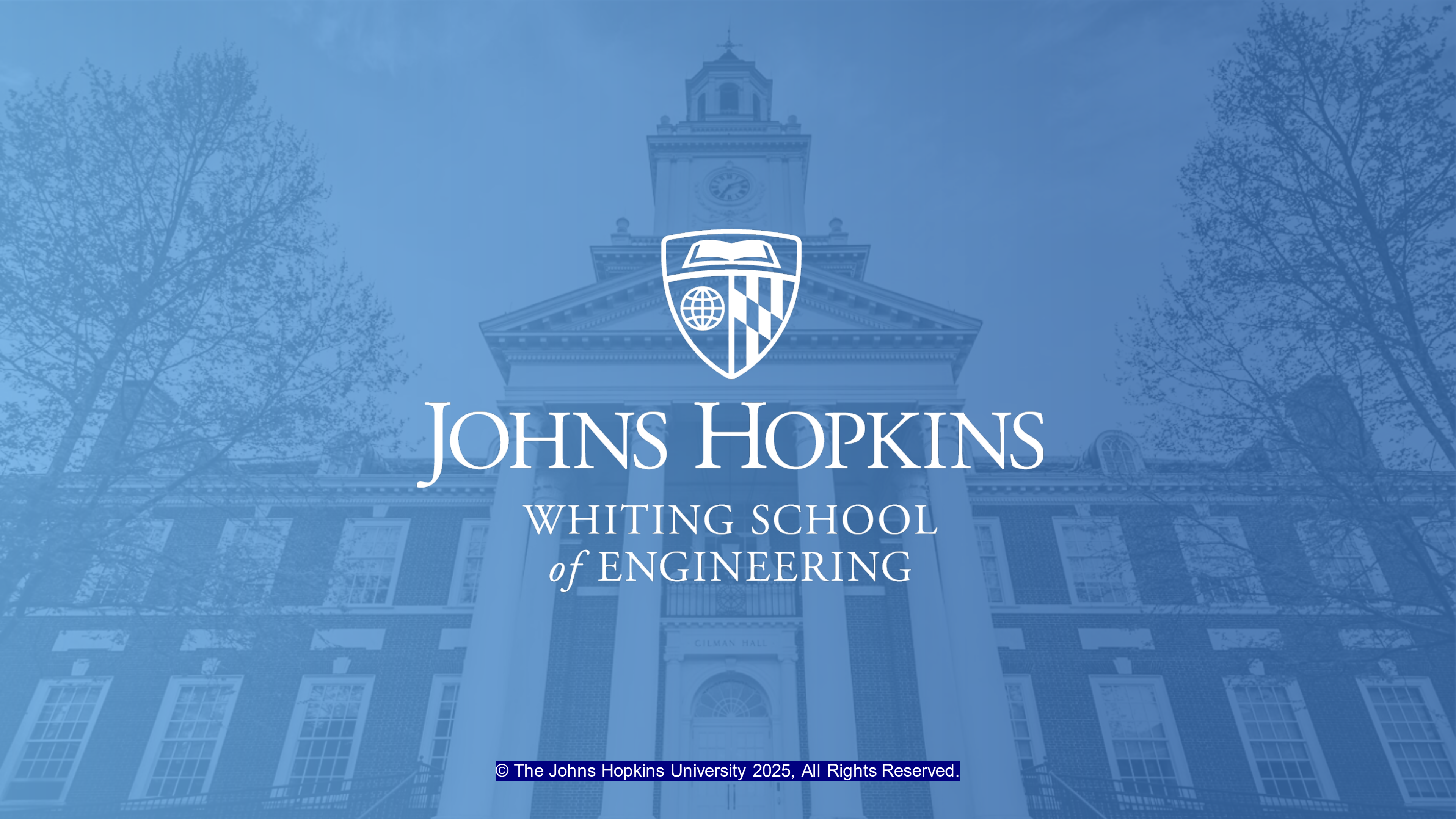
How it Works

- **Pheromone Trails:** Paths are marked with pheromones, with higher pheromone levels indicating better paths.
- **Exploration/Exploitation:** Ants probabilistically choose paths based on pheromone levels and heuristic desirability.
- **Reinforcement:** Successful paths accumulate more pheromones, leading to convergence on the best solutions.



Wrapping Up Optimization

1. **What is Optimization?**
   1. Optimization involves finding the best solution to a problem within a set of constraints.
   2. Types include linear, quadratic, integer, dynamic, and heuristic/metaheuristic.
2. **Exact Optimization Techniques:**
   1. Linear Programming
   2. Quadratic Programming
   3. Dynamic Programming
   4. Integer Programming
3. **Heuristic/Metaheuristic Approaches:**
   1. Useful for complex problems where exact solutions are computationally infeasible.
   2. Simulated Annealing, Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization.


JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING
GILMAN HALL