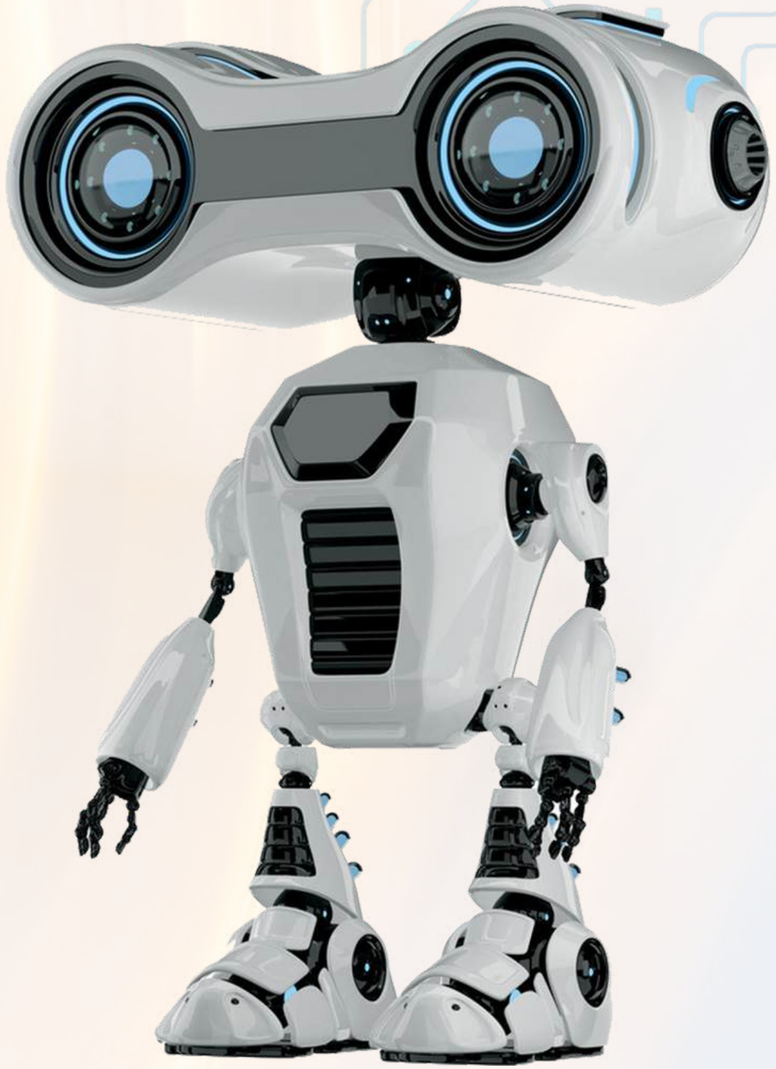
Smart Task Manager System

HOW GENETIC & FIREFLY & GREEDY ALGORITHM USED IN THE SYSTEM

Deya Faisal Wadee Smirat
Leen Mohammed

APOUT OUR PROJECT

This project focuses on developing a Smart Task Manager system using optimization algorithms to enhance task distribution among employees based on skill matching and workload balance. The system incorporates a Genetic Algorithm (GA), Firefly Algorithm, and Greedy Algorithm to ensure efficient task allocation. These approaches allow for intelligent exploration of possible assignments without manually evaluating all combinations, which can be computationally expensive. Each algorithm contributes uniquely: GA handles global optimization, Firefly focuses on convergence through attractiveness, and the Greedy approach ensures quick, locally optimal decisions



Problem Statement!

n many organizations, assigning tasks to employees manually often leads to inefficiencies, such as unbalanced workloads or assigning tasks to individuals who lack the required skills. Manually exploring all possible task-employee combinations becomes impractical as the number of tasks and employees increases, leading to suboptimal performance and reduced productivity.

Project Objectives & Algorithms Used

- **Improving Task Distribution Among Employees**
- **Considering Skills and Workload Balance**
- **Comparing the Performance of Multiple Algorithms**
- **To assign tasks to employees in an optimal way by:**
- **Respecting employee time limits.**
- **Maximizing overall task distribution quality.**

Genetic Algorithm (GA)

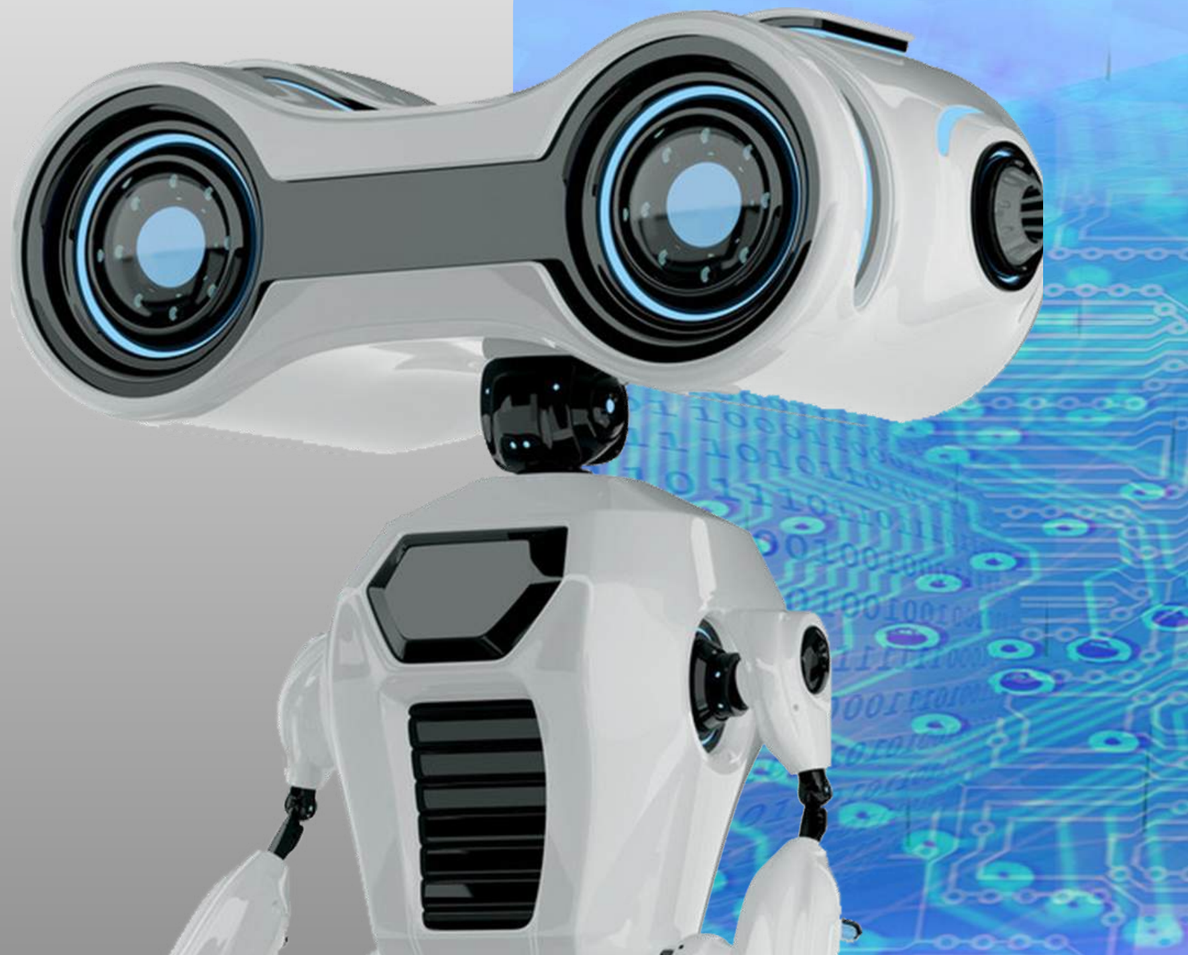
Based on the principle of natural selection, it gradually evolves better solutions over generations.

Firefly Algorithm (FA)

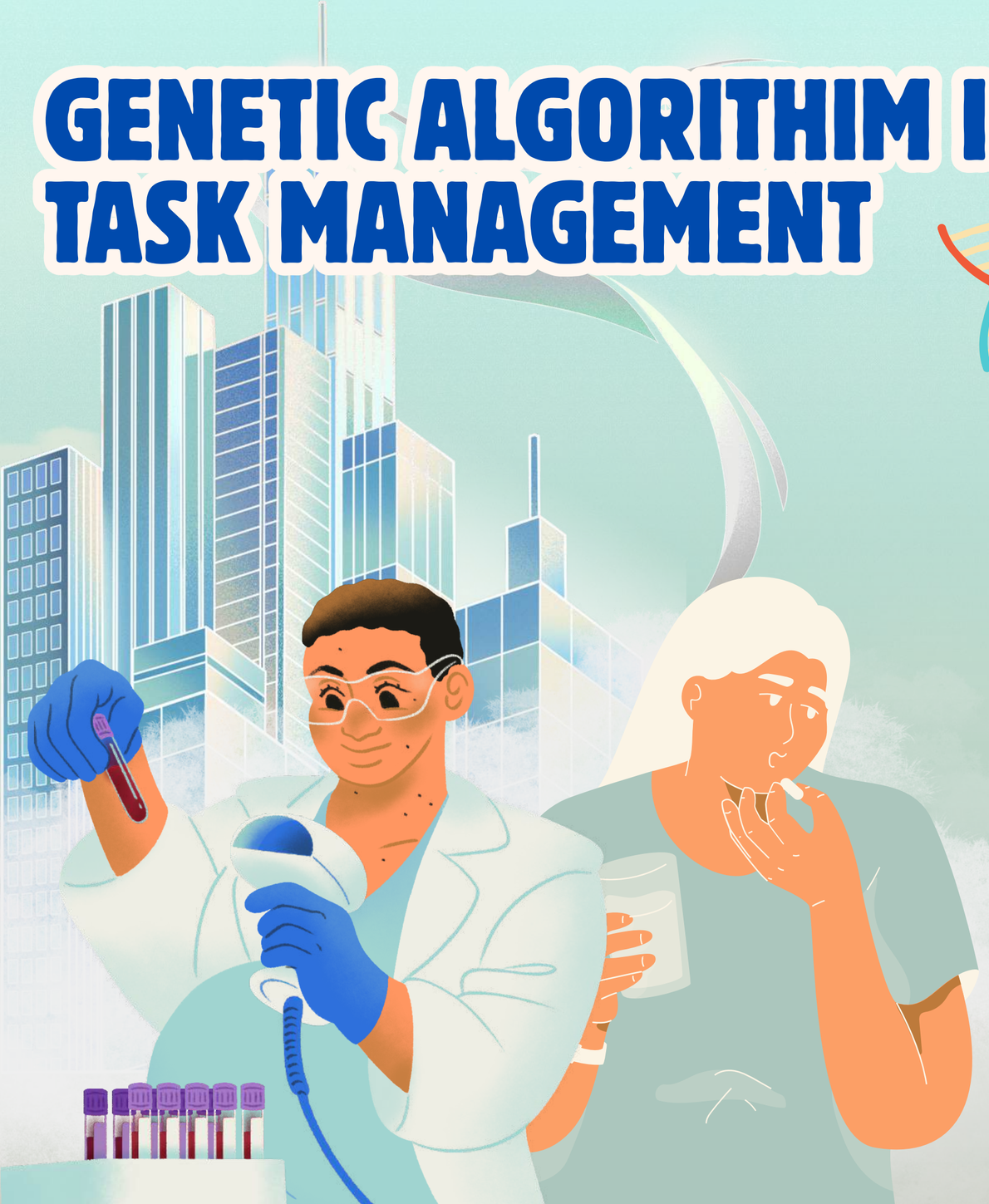
Inspired by the natural behavior of fireflies, it uses attraction mechanisms to move towards globally optimal solutions.

Greedy Algorithm

Follows a straightforward approach by selecting the best immediate option at each step to quickly reach a final solution.



GENETIC ALGORITHIM IN TASK MANAGEMENT



GOAL:

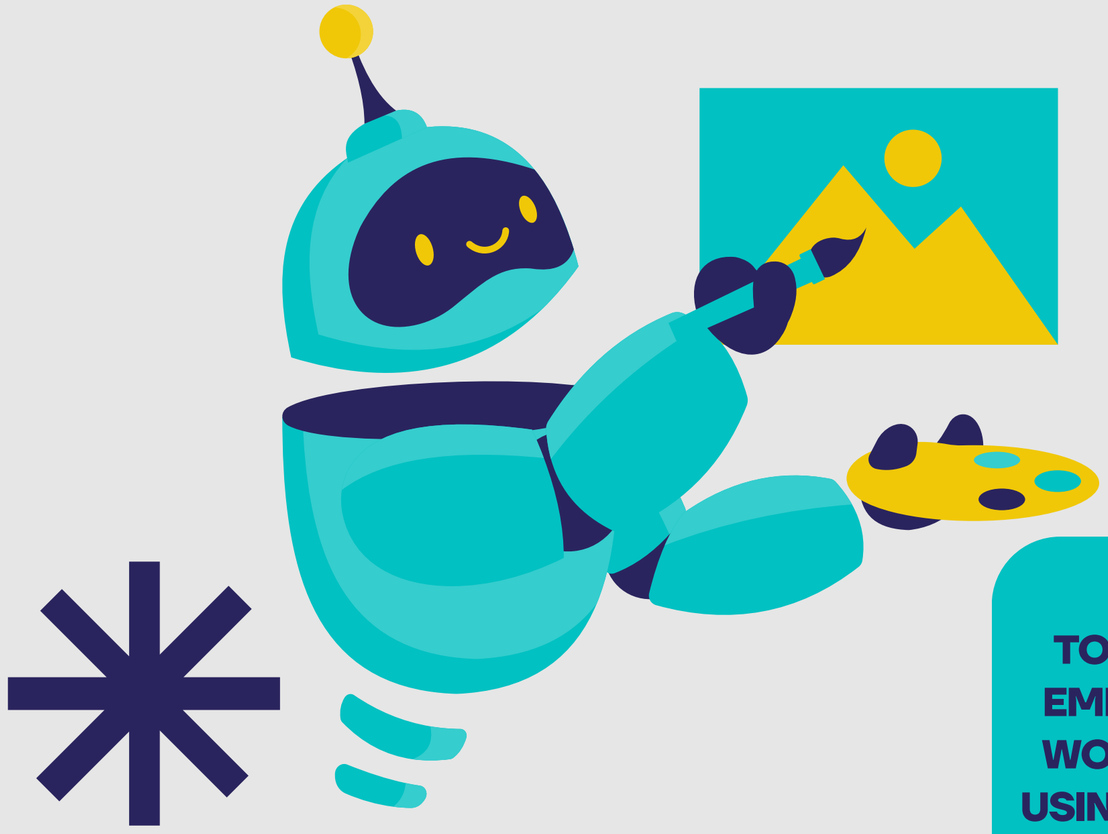
- ASSIGN TASKS TO EMPLOYEES EFFICIENTLY.
- MATCH EMPLOYEE SKILLS AND AVAILABILITY.
- REDUCE FALLBACK OR UNQUALIFIED ASSIGNMENTS.
- BALANCE WORKLOAD AMONG THE TEAM

HOW IT WORKS:

INSPIRED BY NATURAL EVOLUTION, IT IMPROVES SOLUTIONS OVER GENERATIONS:

1. GENERATE RANDOM ASSIGNMENTS (INITIAL POPULATION).
2. EVALUATE EACH ASSIGNMENT BASED ON:
   - SKILL MATCH
   - AVAILABLE WORKING HOURS
3. SELECT BEST SOLUTIONS, COMBINE THEM (CROSSOVER).
4. RANDOM MUTATIONS TO MAINTAIN DIVERSITY.
5. REPEAT FOR SEVERAL GENERATIONS TO GET THE BEST RESULT.

FIREFLY ALGORITIM



HOW IT WORKS :

GENERATE RANDOM SOLUTIONS (TASK-TO-EMPLOYEE ASSIGNMENTS).

1. **EACH FIREFLY (SOLUTION) HAS A BRIGHTNESS = FITNESS SCORE:**
   - **+10 POINTS FOR PERFECT MATCH (SKILL + TIME).**
   - **+5 POINTS FOR PARTIAL MATCH (SKILL ONLY).**
   -
2. **FIREFLIES MOVE TOWARD BRIGHTER (BETTER) SOLUTIONS USING:**
   - **ATTRACTION BASED ON SIMILARITY (DISTANCE).**
   -
   - **RANDOM EXPLORATION (MUTATION).**
3. **THIS REPEATS FOR MULTIPLE GENERATIONS.**

GOAL :

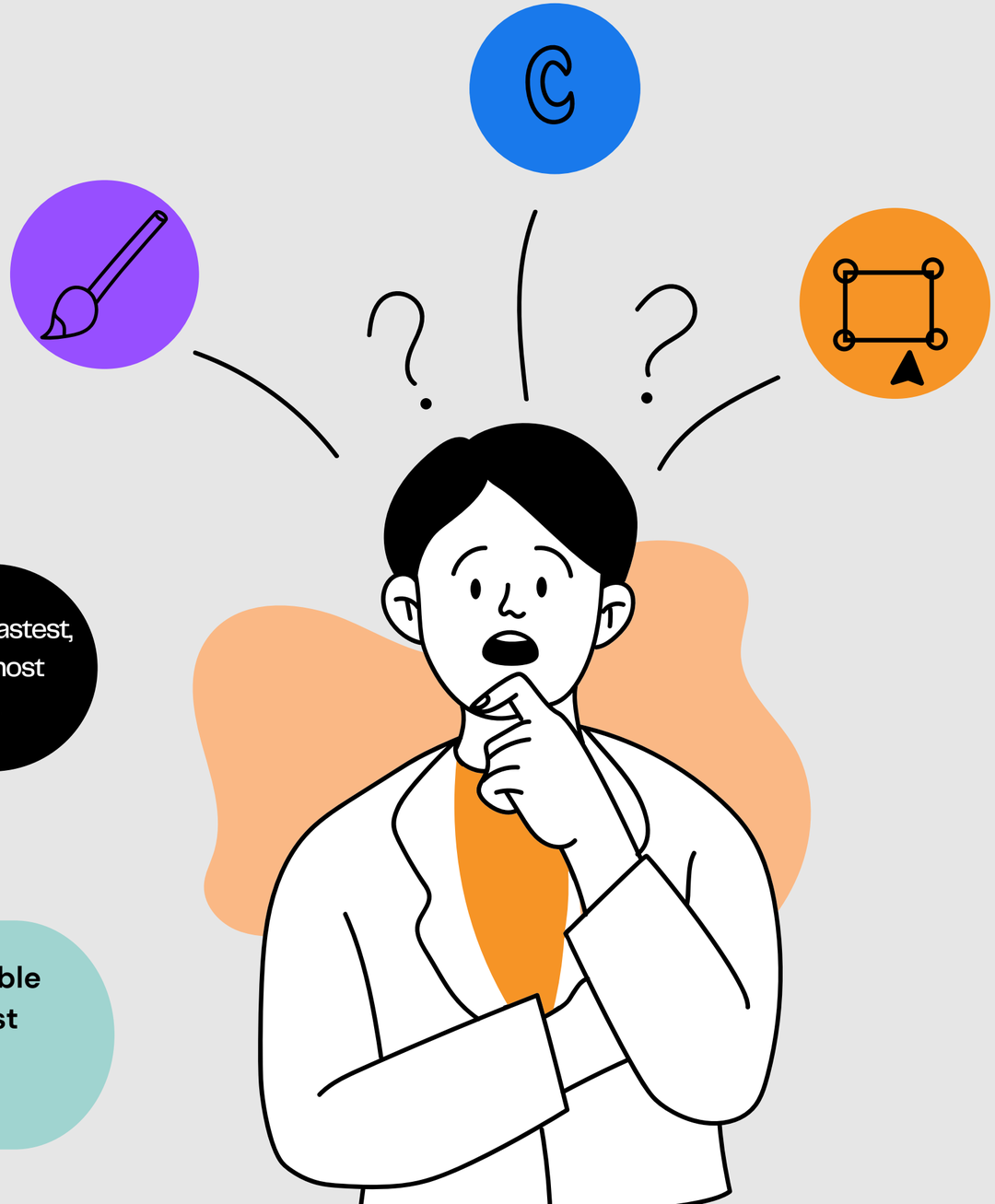
TO FIND THE BEST WAY TO ASSIGN TASKS TO EMPLOYEES BY OPTIMIZING SKILL MATCHING, WORKLOAD BALANCE, AND TIME AVAILABILITY USING A SMART, NATURE-INSPIRED ALGORITHM.

GREEDY ALGORITHIM

How It Works:

1. **For each task:**
   - **Find employees who have the required skill and enough time.**
   - **From those, pick the one with least current workload.**
2. **If no skilled employee is available, assign to any employee with available time (fallback).**
3. **Repeat until all tasks are assigned**

Greedy is the fastest, but not the most accurate.



Goal:

Quickly assign tasks to available and suitable employees using the simplest and fastest method. and Doesn't guarantee global optimization like Genetic or Firefly.

Explanation of Results

🧬 Genetic Algorithm:

- Pros: Finds high-quality distributions by evolving solutions over generations.
- Cons: Takes more time; depends on parameters (generations, population size).
- Best For: Situations needing optimal assignment with many constraints.

🔥 Firefly Algorithm:

- Pros: Good balance between quality and time; uses brightness (fitness) and attraction logic.
- Cons: Slightly slower than greedy, slightly less accurate than genetic.
- Best For: Medium-sized tasks where trade-off between time and accuracy matters.

⚡ Greedy Algorithm:

- Pros: Very fast; great for real-time systems.
- Cons: Does not guarantee optimal distribution; may assign tasks unfairly if options are limited.
- Best For: Small or urgent task assignments where speed is critical.

Comparison of Task Assignment Algorithms

Criterion	Genetic Algorithim	Firefly Algorithim	Greedy Algorithim	
Assignment Quality	Excellent optimal skill & time matching	Good – balanced skill/time match	Basic – assigns quickly without full evaluation	
Execution Time	Moderate (0.3–0.4 sec)	Slightly slower (≈ 0.4 sec)	Very fast (~0.05 sec)	
Skill Matching	High accuracy in matching tasks to skills	Good matching, but not perfect	Limited – picks first available	
Fallback Usage	Few fallback assignments	Moderate number of fallback	High fallback coun	
Approach Type	Evolutionary (crossover + mutation)	Attraction-based (light intensity simulation)	Greedy local decision	
Best For	Complex scenarios with multiple constraints	Balanced tasks and performance	Small datasets or time-critical cases	
Customizability	Highly configurable	Moderately configurable	Minimal customization	
Exploration Depth	Strong global search	Moderate exploration of solution space	Very limited	
Code Complexity	High – needs more logic and setup	Medium – manageable complexity	Low – easy to implement and understand	
Scalability	Good with parameter tuning	Decent for mid-size problems	Best with small or simple inputs	

THE END - CONCLUSION

In this project, we explored and compared three optimization algorithms — Genetic, Firefly, and Greedy — for intelligent task assignment.

- ***Each algorithm demonstrated unique strengths:***
  - Genetic Algorithm provided the most accurate results.
  - Firefly Algorithm offered a good balance between speed and quality.
  - Greedy Algorithm was the fastest, suitable for simple tasks.

- **By visualizing results and performance, we gained insight into how AI can improve decision-making in workforce management.**

- **This work highlights the power of smart algorithms in solving real-world problems efficiently and flexibly.**