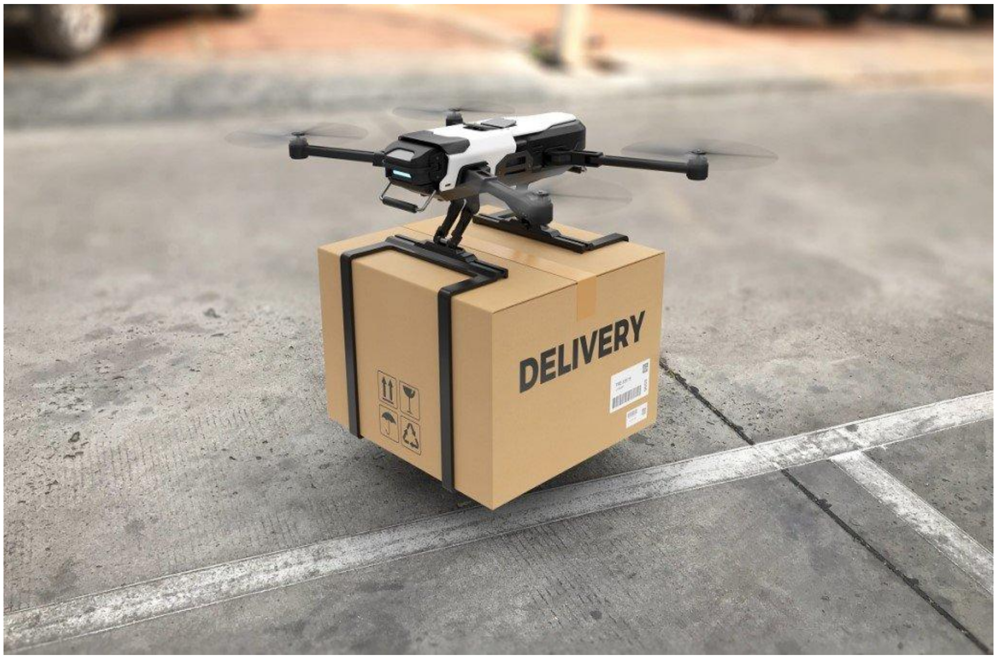
Delivery Drone Route Proposal



TEAM MEMBERS :

- شيماء عماد محمد محمود
- أميرة حسين احمد شخبه
- حمزة محمود فرج سليمان
- عبدالرحمن اليمني معوض
- اسماعيل وصال المصلحى عامر
- مريم جمعه حسين ضباش
- محمد محمود عبد الحميد الشويمى
- محمد عبد الغني الدسوقي

Project Proposal

Drone Delivery Route Optimization Using AI Search Algorithms

1. Project Title

Drone Delivery Route Optimization Using Multiple AI Search Algorithms

2. Problem Definition

The goal of this project is to solve the **Drone Delivery Route Optimization Problem**, where an Unmanned Aerial Vehicle (UAV) must find an efficient route to reach specific delivery points.

- **Representation:** The delivery environment is modeled as a **graph**, where each location is a **Node**, and each valid route between nodes is an **Edge** that may have a weight (distance, energy, or time).
- **Main Challenge:** Selecting and comparing multiple Artificial Intelligence (AI) search algorithms to determine which one can generate a valid and optimal route efficiently, considering factors such as reachability, cost, memory usage, and execution time.

3. Rationale (Why This Problem?)

This problem was selected due to its practical relevance and academic value:

- **Modern Application:** Drone delivery is increasingly used in logistics and smart cities.
- **Optimization Impact:** Enhanced route optimization reduces delivery time, energy consumption, and improves system efficiency.
- **Academic Suitability:** The problem perfectly matches the concepts taught in the AI course, especially search algorithms.

4. Algorithms to Be Implemented

Below is the list of algorithms the team will implement and compare.

Uninformed Search

- **Depth First Search (DFS)**
- **Breadth First Search (BFS)**
- **Uniform Cost Search (UCS)**

Informed Search

- **A* Search**
- **Greedy Best-First Search**

Evolutionary Algorithm

- **Genetic Algorithm (GA)**

---

5. Detailed Algorithm Roles and Implementation

#	Algorithm	Team Member	Purpose	Key Notes	Implementation
1	Depth First Search (DFS)	**Shimaa**	Check reachability and explore paths deeply.	Does not guarantee optimality. Useful for connectivity.	Python (Stack-based).
2	Breadth First Search (BFS)	**Hamza**	Find shortest path in terms of number of steps.	Guarantees shortest path in unweighted graphs.	Python (Queue-based).
3	Uniform Cost Search (UCS)	**Mariam**	Find the least-cost path in weighted graphs.	Guarantees optimal solution.	Python (Priority Queue).
4	A* Search	**Ismail**	Find an optimal and efficient path using heuristics.	Combines cost + heuristic. Optimal if heuristic is admissible.	Python (Priority Queue).
5	Greedy Best-First Search	**Amira**	Choose the next node closest to the goal.	Very fast but not optimal. Can get stuck in local minima.	Python or C++.

#	Algorithm	Team Member	Purpose	Key Notes	Implementation
6	Genetic Algorithm (GA)	**Abdelrahman**	Produce near-optimal routes via population evolution.	Useful for large or complex search spaces.	Python (Evolutionary approach).
7	Data Collection & File Organization	**Mohamed ElShemimy**	Collect all algorithm outputs and organize them into one structured file.	Ensures proper formatting and documentation of results.	Word
8	Final Comparison & Evaluation	**Mohamed Abdelghany**	Compare all algorithms using metrics provided by the team and create the final comparison tables.	Produces final performance summary.	Word

6. Comparison Measures

All algorithms will be evaluated and compared based on the following metrics:

- **Execution Time:** How fast each algorithm finds a route.
- **Memory Usage:** Total memory consumed during execution.
- **Success Rate:** Whether the algorithm can reach the goal or not.
- **Solution Optimality:** How close the solution is to the true optimal route.
- **Scalability:** Performance when the graph size increases.

7. Team Roles and Responsibilities

Team Member	Primary Role	Secondary Role	Tools Used
Shimaa	DFS Implementation	Testing & Documentation	Python
Hamza	BFS Implementation	Testing & Documentation	Python
Ismail	A* Search Implementation	Testing & Documentation	Python
Mariam	UCS Implementation	Testing & Documentation	Python
Amira	Greedy Search Implementation	Testing & Documentation	Python / C++
Abdelrahman	Genetic Algorithm Implementation	Testing & Documentation	Python
Mohamed ElShemimy	Data Collection & File Organization	Documentation	Word
Mohamed Abdelghany	Final Comparison & Evaluation	Documentation	Word
Team Leader(Shimaa)	GitHub Repo Management, Merging Code, Proposal Submission	—	GitHub