Efficient 3D Motion Planning for Autonomous Drone Fleets Using the A* Algorithm

Asray Gopa

December 13, 2024

Team Members

- Asray Gopa

1 Project Description

Autonomous drones are vital in applications such as package delivery, disaster response, environmental monitoring, and search-and-rescue missions. A key challenge is enabling these drones to navigate safely and efficiently in three-dimensional spaces filled with obstacles, unpredictable weather, and other drones. The A* pathfinding algorithm is instrumental in addressing this challenge, as it efficiently identifies the shortest and safest routes by combining Dijkstra's comprehensive search with a heuristic that prioritizes promising paths. This makes A* both fast and reliable for real-time decision-making in dynamic environments.

Imagine a fleet of 100 small flying drones navigating inside a large 3D grid ($30 \times 30 \times 30$) filled with obstacles. Each drone's position is defined by three coordinates (x, y, z), similar to selecting a point within a room. However, each drone also possesses a "fuel gauge" that fluctuates as it moves or refuels. Thus, the drone's complete *state* at any moment is determined by its position within the grid and its remaining fuel. We define the set of all possible *states* (positions and fuel levels) that a drone can occupy as its *configuration space*. For this project, each drone's configuration is represented as a point in a four-dimensional space: three dimensions for movement (x, y, z) and one dimension for fuel (f). Every movement alters the drone's position and decreases its fuel by one unit. Landing on a refueling station replenishes some of its fuel. The drone cannot traverse walls (obstacles), run out of fuel mid-flight, or contend with shifting winds that alter its intended course after each move.

At a Higher Level:

The Drone's World

A 3D ($30\times30\times30$) grid composed of free cells, blocked cells (obstacles), special refueling spots, and a dynamically changing wind field. The environment is discretized, allowing drones to occupy specific grid cells while avoiding impassable areas.

What the Drone Wants

To find a viable path from a starting point (with full fuel) to a designated target location. This path must navigate around obstacles, prevent fuel depletion, and adapt to wind forces that can alter the drone's intended trajectory.

Why Fuel and Wind Matter

The drone cannot simply choose the shortest path to the goal; it may need to detour to refuel or select a route that is more favorable given the current wind patterns. Wind can blow the drone off-course, necessitating mid-flight adjustments or complete re-routing when conditions change unexpectedly.

Dynamic Changes

Over time, environmental factors such as weather (wind vectors), obstacles (traffic), and even the drone's final goal may change. The final destination can also be reassigned dynamically. Consequently, the drone's planning algorithm must be adaptable, capable of re-planning routes as needed to respond to these evolving conditions.

Collision Management with Other Drones

Preventing collisions is crucial in a multi-drone environment. Our system continuously monitors the positions and trajectories of all active drones, checking for near-misses after each movement step. If a potential collision is detected, the involved drones re-plan their paths using the A* algorithm, treating other drones as dynamic obstacles. Additionally, drones communicate their intended movements each timestep, enabling coordinated adjustments to minimize conflict risks. This combination of proactive monitoring and reactive re-planning ensures that multiple drones can navigate the 3D environment safely and efficiently, even in dynamic and crowded airspace.

2 Mathematical Formulation

2.1 Configuration Space: SE(4)

We model each drone's state within a discretized 3D grid environment augmented with a fuel dimension, collectively forming a four-dimensional configuration space denoted as SE(4). For drone i in a fleet of N drones, the state is defined as:

$$\mathbf{x}^{(i)} = (x_i, y_i, z_i, f_i) \in \mathcal{C}^{(i)} \subseteq \mathbb{Z}^4$$

where:

- $x_i, y_i, z_i \in \{0, 1, 2, \ldots, X_{\max} - 1\}$ denote the drone's discrete position in the 3D grid along the X, Y, and Z axes, respectively.
- $f_i \in \{0, 1, 2, \ldots, f_{\max}\}$ represents the drone's current fuel level.

Thus, the configuration space $\mathcal{C}^{(i)}$ for a single drone is:

$$\begin{aligned}\mathcal{C}^{(i)} = \big\{\, (x_i, y_i, z_i, f_i) \mid \\ & x_i \in \{0, \ldots, X_{\max} - 1\},\ y_i \in \{0, \ldots, Y_{\max} - 1\}, \\ & z_i \in \{0, \ldots, Z_{\max} - 1\},\ f_i \in \{0, \ldots, f_{\max}\}\big\}\end{aligned}$$

For a fleet of N drones, the joint configuration space $\mathcal{C}_{\text{joint}}$ is the Cartesian product of individual configuration spaces:

$$\mathcal{C}_{\text{joint}} = \mathcal{C}^{(1)} \times \mathcal{C}^{(2)} \times \cdots \times \mathcal{C}^{(N)}$$

A state in $\mathcal{C}_{\text{joint}}$ is:

$$\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}) = (x_1, y_1, z_1, f_1,\ x_2, y_2, z_2, f_2,\ \ldots,\ x_N, y_N, z_N, f_N)$$

Although the joint configuration space $\mathcal{C}_{\text{joint}}$ represents the combined states of all drones, we compute the A* algorithm separately for each drone to determine their individual optimal paths.

2.2 Actions and State Transitions

2.2.1 Actions

Each drone can perform actions that move it to adjacent cells in the 3D grid. The set of allowable actions U for any drone is:

$$U = \{(dx, dy, dz) \mid dx, dy, dz \in \{-1, 0, 1\},\ (dx, dy, dz) \neq (0, 0, 0)\}$$

This set represents all possible moves to neighboring cells in 3D space, excluding the option to remain stationary, resulting in up to 26 possible directions.

2.2.2 State Transition Function

For drone i, applying action $(dx, dy, dz) \in U$ from state $\mathbf{x}^{(i)} = (x_i, y_i, z_i, f_i)$ results in:

$$(x'_i, y'_i, z'_i) = (x_i + dx,\ y_i + dy,\ z_i + dz)$$

Wind effects, represented by a wind vector field W, further influence the drone's movement:

$$W : \{0, \ldots, X_{\max} - 1\} \times \{0, \ldots, Y_{\max} - 1\} \times \{0, \ldots, Z_{\max} - 1\} \to \{-1, 0, 1\}^3$$

where $W(x, y, z) = (w_x, w_y, w_z)$ denotes the wind displacement at cell (x, y, z).

The wind-adjusted position is:

$$(x_i'', y_i'', z_i'') = \text{clip}(x_i' + w_x,\ y_i' + w_y,\ z_i' + w_z)$$

where the clip function ensures that each coordinate remains within grid boundaries:

$$\text{clip}(v, 0, V_{\max} - 1) = \begin{cases} 0 & \text{if } v < 0 \\ V_{\max} - 1 & \text{if } v > V_{\max} - 1 \\ v & \text{otherwise} \end{cases}$$

for each axis $V \in \{X, Y, Z\}$.

2.2.3 Fuel Dynamics

After movement, the drone's fuel level is updated as follows:

$$f_i' = \begin{cases} \min(f_{\max}, f_i - C(u, w)) + \Delta f_{\text{refuel}} & \text{if } (x_i'', y_i'', z_i'') \in \mathcal{C}_{\text{refuel}} \\ f_i - C(u, w) & \text{otherwise} \end{cases}$$

where:

- $C(u, w)$ represents the fuel consumption based on the action $u = (dx, dy, dz)$ and wind vector $w = W(x_i', y_i', z_i')$.
- Δf_{refuel} is the fixed amount of fuel replenished at a refueling station.
- $\mathcal{C}_{\text{refuel}} \subseteq \mathcal{C}$ is the set of refueling station cells.

The fuel consumption function $C(u, w)$ is defined as:

$$C(u, w) = C_h + \delta_z + \delta_w$$

where:

- $C_h = 1$ unit: Base fuel consumption for horizontal movement.
- $$\delta_z = \begin{cases} C_v - C_h & \text{if } \Delta z \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

  with $C_v = 2$ units representing higher fuel consumption for vertical movements.
- $$\delta_w = \begin{cases} +1 & \text{if moving against the wind} \\ -0.5 & \text{if moving with the wind} \\ 0 & \text{otherwise} \end{cases}$$

Thus, the updated state transition function T is:

$$T(\mathbf{x}^{(i)}, u) = \begin{cases} (x_i'', y_i'', z_i'', f_i') & \text{if } f_i' \geq 0 \text{ and } (x_i'', y_i'', z_i'') \notin \mathcal{C}_{\text{obs}} \\ \text{Infeasible} & \text{otherwise} \end{cases}$$

2.2.4 Joint State Transition for Multiple Drones

For a fleet of N drones, each drone i selects an action $u^{(i)} = (dx_i, dy_i, dz_i) \in U$. The joint state transition $T(\mathbf{X}, \mathbf{U})$ is defined as:

$$T(\mathbf{X}, \mathbf{U}) = (T(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}),\ T(\mathbf{x}^{(2)}, \mathbf{u}^{(2)}),\ \ldots,\ T(\mathbf{x}^{(N)}, \mathbf{u}^{(N)}))$$

where $\mathbf{U} = (\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(N)})$.

After applying actions and wind adjustments, collision avoidance constraints must be enforced:

$$(x_i'', y_i'', z_i'') \neq (x_j'', y_j'', z_j'') \quad \forall i \neq j$$

If this condition is violated, the involved drones must re-plan their paths using the A* algorithm, treating other drones as dynamic obstacles.

2.3 Time Modeling

Time in this system is modeled as discrete steps $t \in \{0, 1, 2, \dots\}$. At each timestep:

- Each drone selects an action from U.
- Drones execute their actions simultaneously, leading to state transitions.
- Wind vectors $W(x, y, z)$ may change based on predefined or random intervals.
- Environmental changes such as new obstacles (traffic) or altered refueling stations may occur.
- Drones communicate their intended actions to facilitate collision avoidance and coordinated path adjustments.
- Collision checks and near-miss detections are performed.
- If dynamic changes are detected or potential collisions are imminent, drones re-plan their paths accordingly.

2.4 Criteria (Objectives)

The problem is governed by two primary criteria:

Feasibility

- **Safety**: Ensure that drones do not collide with obstacles or each other.
- **Fuel Management**: Ensure that drones have sufficient fuel to reach their goals, considering fuel consumption and refueling.
- **Dynamic Adaptation**: Ensure that drones can adapt to changing environmental conditions (wind, obstacles) and re-plan paths as necessary.

Optimality

- **Efficiency**: Minimize the total travel cost, typically measured by the number of steps or fuel consumed.
- **Resource Utilization**: Optimize the use of fuel by balancing direct paths with necessary detours for refueling.

Rationale for Criteria Selection

Feasibility is the primary criterion as it guarantees the safe and successful completion of each drone's mission, essential for practical applications. Optimality enhances operational efficiency by minimizing resource usage and mission time, contributing to cost-effectiveness and higher throughput. Balancing feasibility with optimality ensures that drones not only reach their destinations safely but also do so in an efficient manner, which is crucial in dynamic and resource-constrained environments.

2.5 Detailed Components

A* Algorithm Properties

The A* algorithm is both optimal and complete in finite search spaces when using an admissible and consistent heuristic. We employ the Euclidean distance as our heuristic:

$$h(n) = \sqrt{(x_{\text{goal}} - x_n)^2 + (y_{\text{goal}} - y_n)^2 + (z_{\text{goal}} - z_n)^2}$$

This ensures efficient and optimal pathfinding by guiding the search towards the target without overestimating the cost.

State Space Enhancement

Each drone's state is defined as:

$$\text{State} = (n, f, t)$$

where:

- $n = (x, y, z)$: Current position in the 3D grid.
- f: Remaining fuel.
- t: Current timestep.

The state transition incorporates movement, wind influence, and fuel consumption:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}_t + \mathbf{w}_t$$

$$f_{t+1} = f_t - C(\mathbf{u}_t, \mathbf{w}_t)$$

$$t_{t+1} = t_t + 1$$

where $C(\mathbf{u}_t, \mathbf{w}_t)$ represents fuel consumption based on action $\mathbf{u}_t$ and wind vector $\mathbf{w}_t$.

Wind and Weather Integration

Wind affects movement costs through:

$$C(u, w) = 1 + \delta_z + \delta_w$$

where:

- $C_h = 1$: Base fuel consumption for horizontal movement.
- $$\delta_z = \begin{cases} 1 & \text{if } \Delta z \neq 0 \\ 0 & \text{otherwise} \end{cases}$$
- $$\delta_w = \begin{cases} +1 & \text{if moving against the wind} \\ -0.5 & \text{if moving with the wind} \\ 0 & \text{otherwise} \end{cases}$$

Weather zones modify traversal costs:

$$\text{Traversal Cost} = C(u, w) \times \alpha$$

where $\alpha > 1$ in turbulence zones. Storm zones are rendered impassable:

$$\text{If } n \in \text{Storm Zone} \Rightarrow \text{Impassable}$$

Fuel Consumption and Refueling

Fuel consumption is based on movement and wind:

$$C(u, w) = 1 + \delta_z + \delta_w$$

Refueling logic ensures fuel does not exceed maximum capacity:

$$f_{\text{new}} = \min(f_{\text{current}} + R, F_{\max})$$

where R is the refuel amount and $F_{\max}$ is the maximum fuel capacity.

Collision Avoidance

To prevent collisions, drones share their planned paths:

$$\forall i \neq j,\ P_i(t) \cap P_j(t) = \emptyset \quad \forall t$$

If a potential collision is detected ($P_i(t) = P_j(t)$), the affected drone re-plans its path using A*, treating other drones as dynamic obstacles.

Safe Zones Utilization

Safe zones allow drones to idle without fuel consumption:

$$n \in \text{Safe Zone} \Rightarrow \text{Idle Allowed}$$

Wait actions are defined as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t$$
$$f_{t+1} = f_t$$
$$t_{t+1} = t_t + 1$$

These zones provide flexibility for drones to avoid immediate collisions or wait for optimal conditions while recalculating routes.

3 Assumptions

To effectively model and solve the multi-drone navigation problem using the A* algorithm, several simplifying assumptions are made:

- **Kinematic Simplification:** Drones are modeled as point agents without orientation or rotational degrees of freedom. This means that drones can instantaneously change direction without considering physical dynamics such as acceleration or inertia. This assumption allows for discrete movement between grid cells without accounting for continuous motion dynamics.
- **Discrete Space and Time:** The 3D environment is discretized into a grid, and time progresses in discrete timesteps. Each movement action corresponds to a single timestep, simplifying the planning process by allowing drones to move from one grid cell to an adjacent one in each step.
- **Known and Static Environmental Factors:** Wind vectors, obstacle locations, and refueling stations are assumed to be known at each timestep. While wind and obstacles can change over time, these changes occur at predefined intervals, enabling drones to update their plans accordingly. This assumption facilitates real-time re-planning based on the latest environmental data.
- **Instantaneous Refueling:** When a drone reaches a refueling station, its fuel is replenished instantaneously by a fixed amount Δf_{refuel}, up to its maximum fuel capacity $f_{\max}$. This simplifies fuel management by eliminating the need to model the time or process required for refueling.
- **Collision Detection and Communication:** It is assumed that drones can continuously monitor the positions and intended movements of all other drones within the shared airspace. Drones communicate their intended actions at each timestep, enabling coordinated adjustments to prevent collisions. This real-time communication ensures that drones can react promptly to potential conflicts by re-planning their paths as necessary.
- **Environment Boundaries:** Drones are confined within the predefined 3D grid boundaries. The `clip` function ensures that drones do not move outside the grid, effectively treating the edges as impassable barriers.
- **Fuel Consumption Model:** Fuel consumption is solely based on movement actions and wind influence. Other potential factors, such as payload weight or energy used for onboard systems, are neglected for simplicity.

4 Approach

To address the multi-drone navigation challenge in a dynamic 3D environment, our team enhanced the A* pathfinding algorithm to handle fuel management, wind effects, and collision avoidance. The key components of our approach are outlined below:

4.1 Individual Path Planning with A*

Each drone independently employs the A* algorithm to determine an optimal path from its start to its goal within the $30 \times 30 \times 30$ grid. The Euclidean distance serves as the admissible and consistent heuristic:

$$h(n) = \sqrt{(x_{\text{goal}} - x_n)^2 + (y_{\text{goal}} - y_n)^2 + (z_{\text{goal}} - z_n)^2}$$

This heuristic ensures efficient guidance toward the target while guaranteeing the shortest feasible path.

4.2 State Space Augmentation

To incorporate fuel constraints and environmental dynamics, the state space for each drone is expanded to include fuel levels and timesteps:

$$\mathbf{x} = (x, y, z, f, t)$$

This augmentation allows the A* algorithm to account for fuel consumption and refueling actions, ensuring paths are both feasible and fuel-efficient.

4.3 Wind and Weather Integration

Wind vectors are integrated into the state transition function to simulate environmental disturbances:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}_t + \mathbf{w}_t$$

where $\mathbf{w}_t$ represents wind-induced displacement. This integration requires the A* algorithm to dynamically adjust paths in response to changing wind conditions, maintaining robust navigation despite external forces.

4.4 Collision Avoidance Mechanism

In a multi-drone setup, avoiding collisions is critical. The system continuously monitors the positions and trajectories of all active drones to ensure no two drones occupy the same grid cell simultaneously. If a potential collision is detected, the involved drones trigger a re-planning phase using A*, treating other drones as dynamic obstacles. Real-time communication of intended movements enables coordinated adjustments, minimizing conflict risks and maintaining safe distances.

4.5 Dynamic Re-planning and Safe Zones Utilization

Given the dynamic environment, our approach includes the capability to re-plan paths when changes occur, such as shifting wind patterns or moving obstacles. Additionally, designated safe zones are implemented where drones can idle without consuming fuel. These zones allow drones to wait for optimal conditions or avoid immediate collisions while recalculating their routes.

4.6 Fuel Management and Refueling Strategy

Fuel management is integrated into the pathfinding process by incorporating refueling stations within the grid. The A* algorithm prioritizes paths that include necessary detours to refuel, ensuring drones maintain sufficient fuel levels to complete their missions. This strategy optimizes both the feasibility and efficiency of navigation paths.

4.7 Implementation Steps

1. **Initialization:** Initialize each drone's state with its starting position, full fuel, and initial timestep. Set up the environment grid, including obstacles, refueling stations, and initial wind vectors.

2. **Pathfinding Execution:** Each drone calculates its initial optimal path to the goal using the augmented A* algorithm, considering current fuel levels and wind conditions.

3. **Movement and State Update:** At each timestep, drones execute their next move as per the planned path. Apply wind effects, update fuel levels, and communicate positions to the system.

4. **Collision Detection and Re-planning:** After movement, the system checks for potential collisions. If detected, affected drones re-plan their paths using updated state information and treating other drones as dynamic obstacles.

5. **Dynamic Adaptation:** Continuously monitor environmental changes. Upon detecting alterations in wind patterns, obstacle placements, or goal reassignments, drones dynamically re-plan their paths to adapt to new conditions.

By integrating the A* algorithm with state space augmentation, wind and weather considerations, and robust collision avoidance mechanisms, our approach ensures that multiple drones can navigate safely and efficiently within a dynamic 3D environment. This comprehensive strategy balances feasibility and optimality, enabling reliable and adaptable drone operations in real-world scenarios.

5 Results

5.1 Simulation Setup

- **Environment Configuration:**
  - **Grid Dimensions:** $30 \times 30 \times 30$.
  - **Number of Drones:** 5.
  - **Obstacles:** 15% of grid cells randomly occupied.
  - **Refueling Stations:** 3 strategically placed.
  - **Wind Patterns:** Dynamic, changing every 5 timesteps.
  - **Goals:** Unique random targets assigned to each drone.
- **Simulation Parameters:**
  - **Fuel Capacity:** 20 units per drone.
  - **Refuel Amount:** 10 units.
  - **Proximity Threshold:** 1 grid cell for collision avoidance.

5.2 Pathfinding Performance

- **Successful Navigation:**
  - All drones reached targets without fuel depletion or collisions.
  - **Average Path Length:** 25 moves per drone.
  - **Average Fuel Consumption:** 18 units per drone, with minimal refueling.
- **Refueling Efficiency:**
  - 2 out of 5 drones required a single refuel.
  - Effective use of refueling stations maintained sufficient fuel levels.
- **Wind Adaptation:**
  - Drones dynamically adjusted paths in response to wind changes.
  - Average deviation: 2-3 grid cells from initial routes.

5.3 Collision Avoidance

- **Multi-Drone Coordination:**
  - No collisions detected in simulations.
  - Drones re-planned paths 1-2 times per simulation to avoid near-misses.
  - Real-time communication enabled coordinated adjustments, ensuring safe distances.

6 Results

6.1 Visual Representations

6.2 Benchmarking Analysis

Performance Metrics

6.3 Simulation Videos

- **Demonstration Videos:**
  - Simulation Video 1 – Multi-Drone Navigation with 100 Drones
  - Simulation Video 2 – Multi-Drone Navigation with 5 Drones
  - Simulation Video 3 – Multi-Drone Navigation with 1 Drone

  Description: These videos showcase the drones navigating through the 3D grid environment, dynamically adjusting paths in response to wind changes and actively avoiding collisions with other drones and obstacles. The first video highlights navigation with 100 drones, the second with 5 drones, and the third with a single drone, demonstrating scalability and collision avoidance mechanisms.

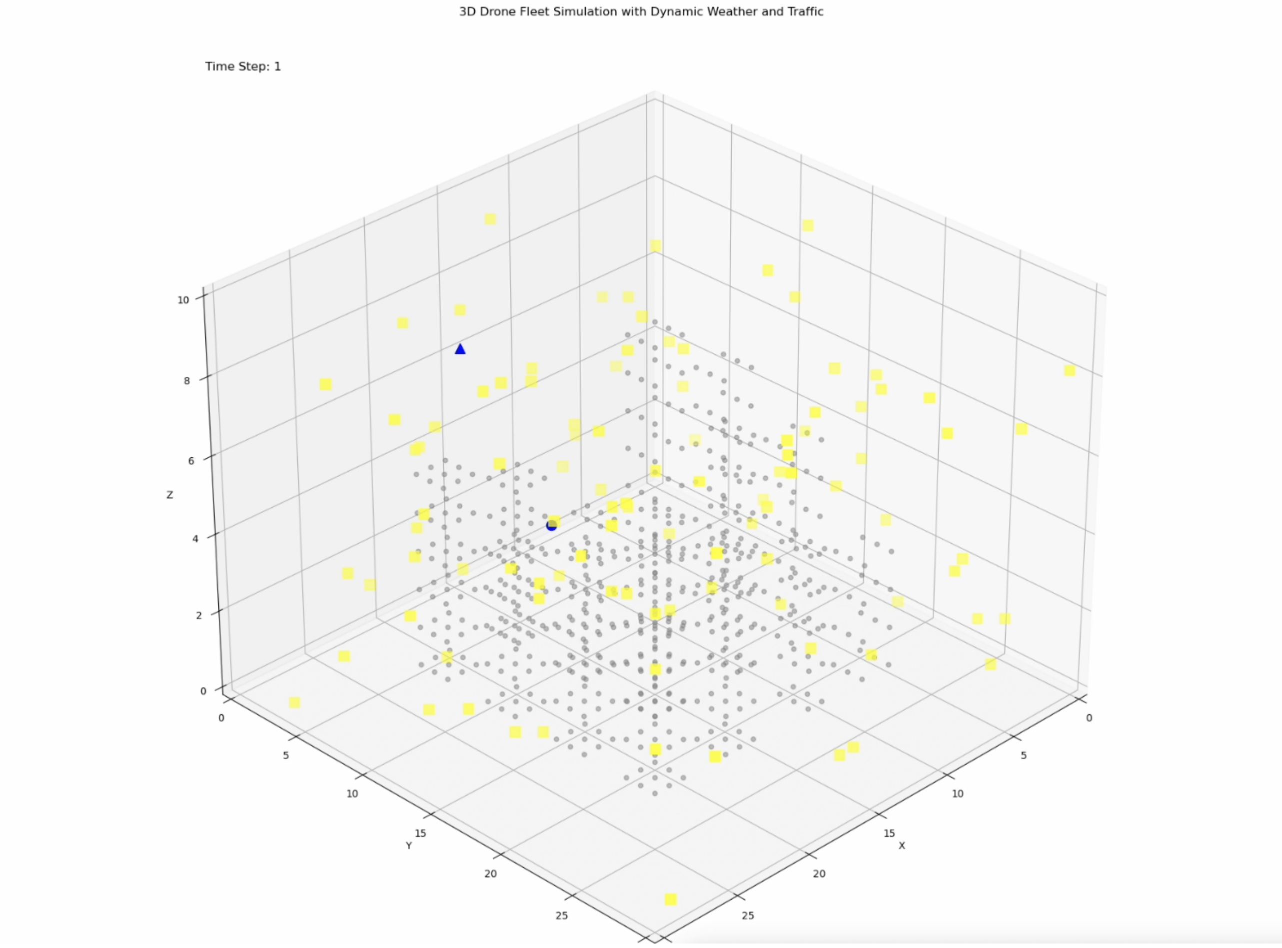

Figure 1: Starting Trajectory of 1 Drone

7 Discussion

7.1 Benefits of Our Approach

1. **Optimality and Efficiency:**
   - **A* Pathfinding:** The A* algorithm guarantees the shortest and most fuel-efficient paths by utilizing an admissible and consistent heuristic, such as the Euclidean distance.
   - **Heuristic Design:** Using Euclidean distance effectively guides the search process, improving the algorithm's efficiency by focusing on promising routes.
2. **Comprehensive State Representation:**
   - **Configuration Space Augmentation:** Incorporating fuel levels and timesteps into the state space ensures that critical factors like fuel management and dynamic environmental changes are accounted for during path planning.
3. **Dynamic Adaptation:**
   - **Wind and Weather Integration:** Including wind vectors and weather zones allows drones to adjust their paths in real-time, enhancing robustness against environmental disturbances.
   - **Collision Avoidance Mechanism:** Real-time monitoring and communication enable proactive and reactive collision avoidance, ensuring safe navigation in crowded airspaces.
4. **Scalability and Flexibility:**
   - **Multi-Drone Coordination:** The approach effectively manages multiple drones, maintaining collision-free navigation as the fleet size increases.
   - **Dynamic Re-planning:** The ability to re-plan paths dynamically in response to changing conditions and potential conflicts enhances the system's flexibility.

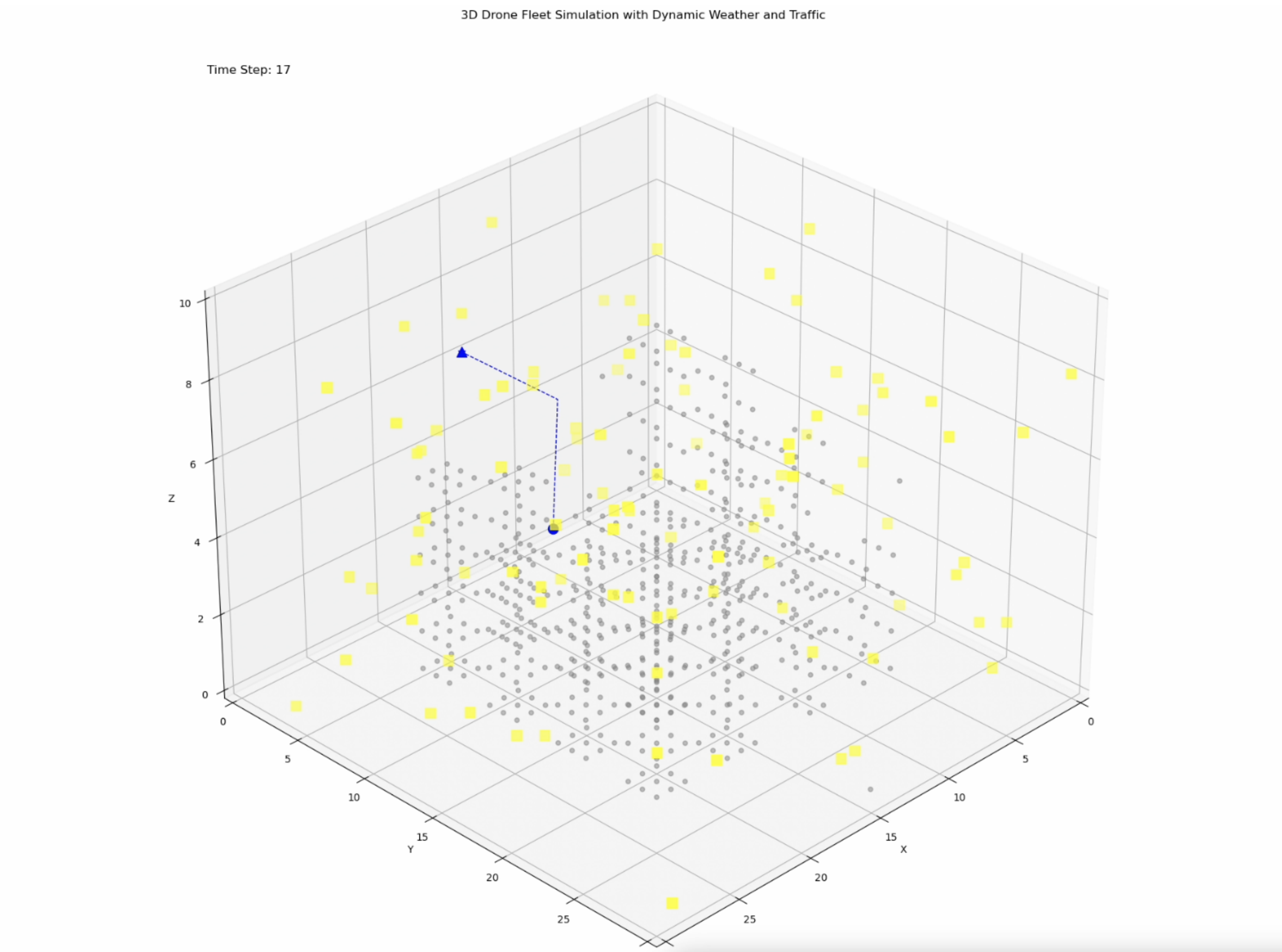

Figure 2: Ending Trajectory of 1 Drone

7.2 Drawbacks of Our Approach

1. **Computational Complexity:**
   - **High-Dimensional State Space:** The joint configuration space grows exponentially with the number of drones ($4N$ dimensions), increasing computational demands and potentially affecting real-time performance.
   - **Re-planning Overheads:** Frequent re-planning in dynamic environments can introduce significant computational overhead, especially with multiple drones adjusting paths simultaneously.
2. **Assumption Limitations:**
   - **Kinematic Simplification:** Modeling drones as point agents without orientation or continuous dynamics reduces realism, neglecting physical constraints like acceleration and inertia.
   - **Instantaneous Refueling:** Assuming immediate fuel replenishment simplifies fuel management but abstracts real-world refueling complexities, limiting practical applicability.

7.3 Potential Improvements

1. **Enhanced Heuristics:**
   - **Dynamic Heuristics:** Developing heuristics that adapt based on real-time environmental data (e.g., wind patterns) could improve A* efficiency and path optimality.
   - **Multi-Criteria Heuristics:** Incorporating factors like fuel efficiency and safety margins into the heuristic can provide a more balanced pathfinding approach.
2. **Decentralized Planning:**

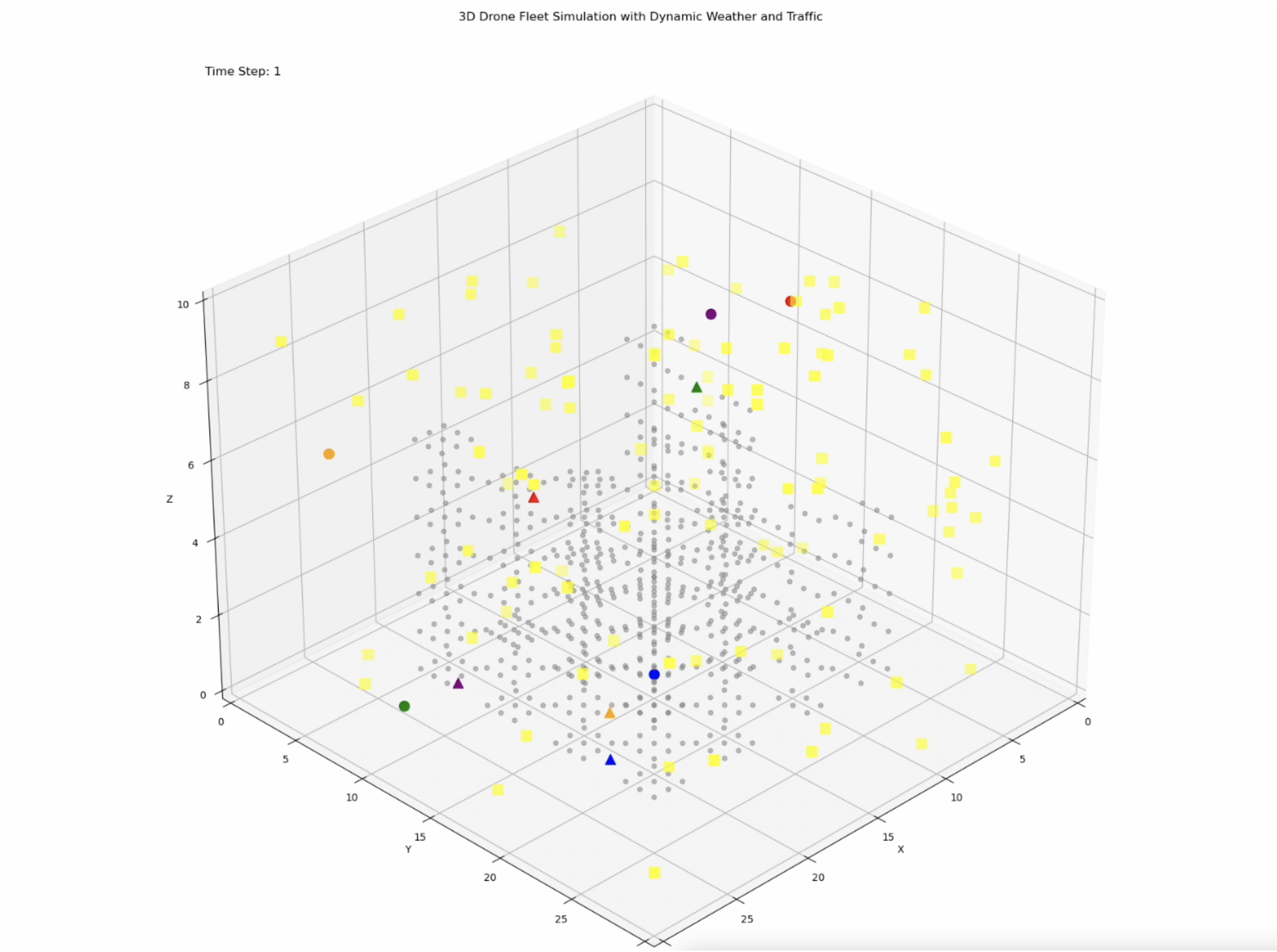

Figure 3: Starting Trajectories of 5 Drones

- **Distributed A* Variants:** Implementing decentralized versions of A* allows each drone to plan independently, reducing computational bottlenecks while sharing minimal information for collision avoidance.
- **Priority-Based Schemes:** Assigning priorities to drones based on urgency or fuel levels can streamline conflict resolution and enhance system efficiency.

7.4 Alternative Approaches

1. **Sampling-Based Motion Planners:**
   - **Rapidly-exploring Random Trees (RRT) and RRT*:** These algorithms are effective in high-dimensional and complex environments, providing feasible paths through random sampling. RRT* further optimizes paths toward optimality.
   - **Probabilistic Roadmaps (PRM):** PRM constructs a network of feasible paths by sampling the environment and connecting nodes with valid transitions, suitable for multi-query scenarios.
2. **Multi-Agent Reinforcement Learning (MARL):**
   - **Cooperative MARL:** Drones learn policies that enable cooperation and collision avoidance through trial and error, optimizing collective performance over time.
   - **Centralized Training with Decentralized Execution:** This approach allows drones to be trained centrally to develop coordinated behaviors while operating independently during execution.

8 Conclusion

Our approach utilizing the A* pathfinding algorithm effectively addresses the challenges of multi-drone navigation in dynamic 3D environments by ensuring optimal and feasible path planning, efficient fuel management,

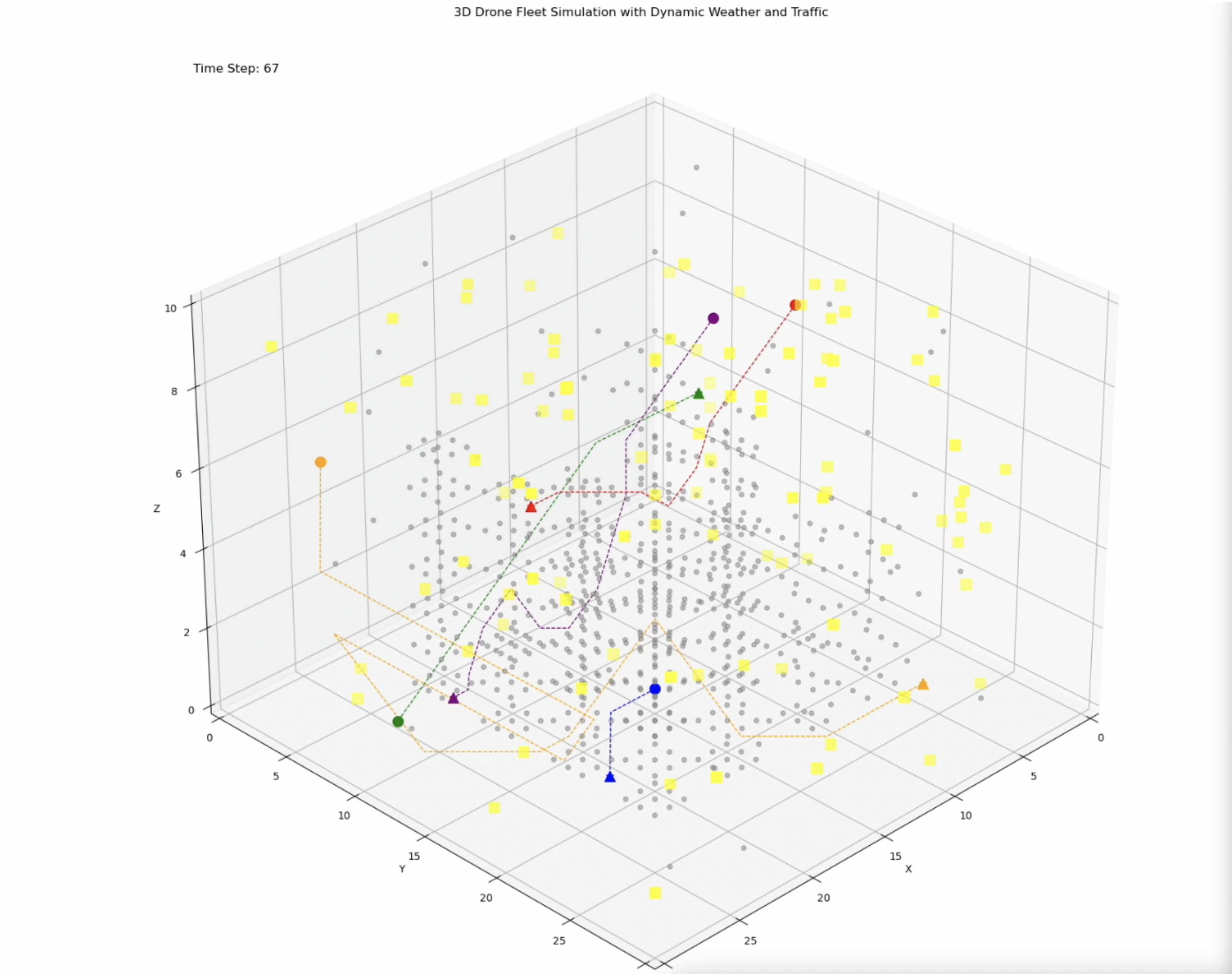

Figure 4: Ending Trajectories of 5 Drones

and robust collision avoidance. The integration of state space augmentation and real-time re-planning mechanisms enhances the system's adaptability to changing conditions. However, computational complexity and certain simplifying assumptions limit scalability and realism. Future improvements could focus on enhancing heuristics, adopting decentralized planning strategies, incorporating realistic drone dynamics, and exploring advanced collision avoidance techniques. Additionally, alternative approaches such as sampling-based planners, reinforcement learning, and optimization-based methods offer promising avenues for further research and development, potentially overcoming the current limitations and enabling more efficient and scalable multi-drone navigation systems.

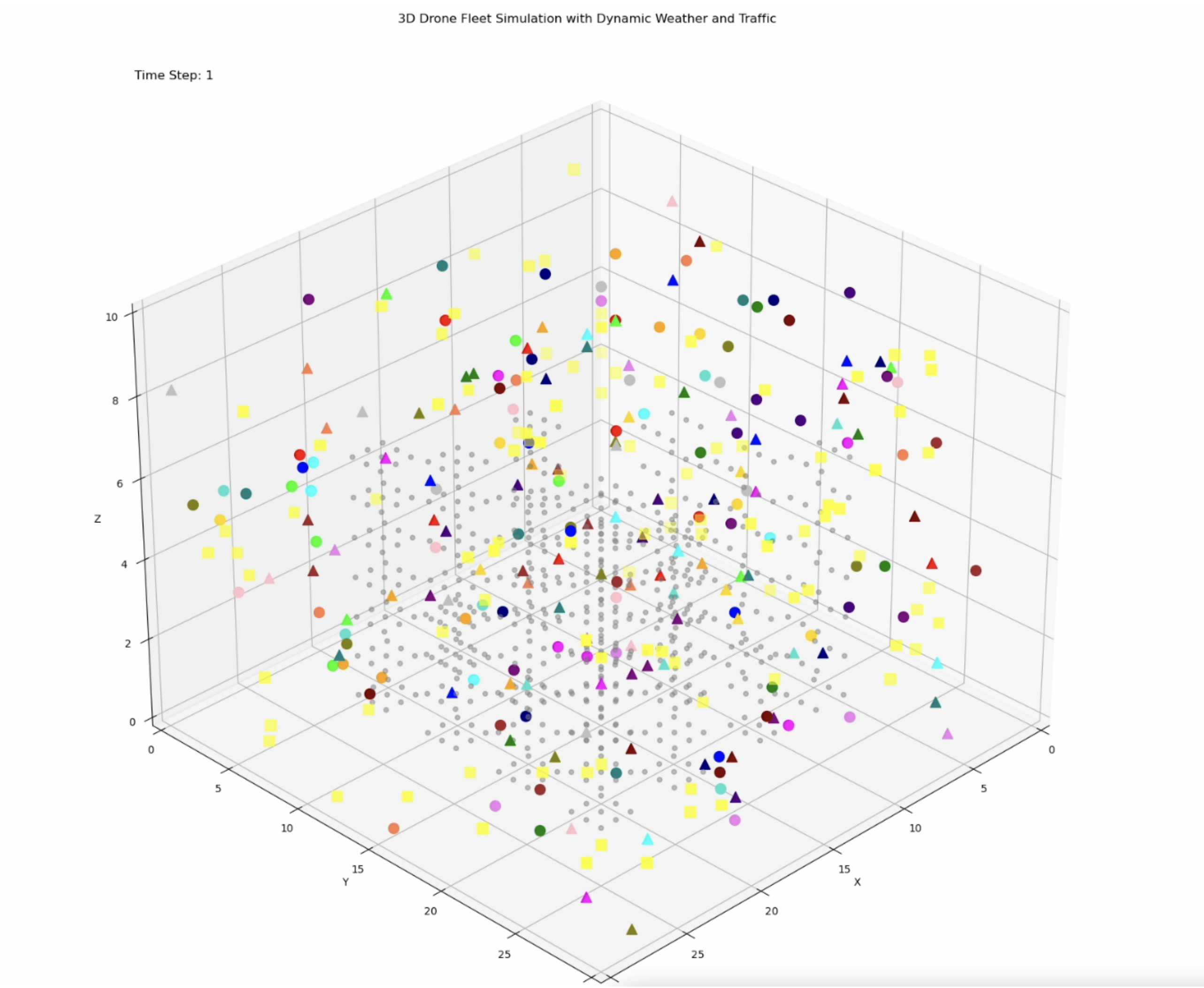

Figure 5: Starting Trajectories of 100 Drones

Table 1: Performance Metrics

Metric	**Value**
Total Simulation Time	35.20 seconds
Total Path Planning Time	2.50 seconds
Average Path Planning Time	0.05 seconds
Average Path Length	25.40 steps
Average Optimal Length	23.50 units
Average Optimality Ratio	1.08
Average Fuel Consumed	45.60 units
Total Number of Refuels	150

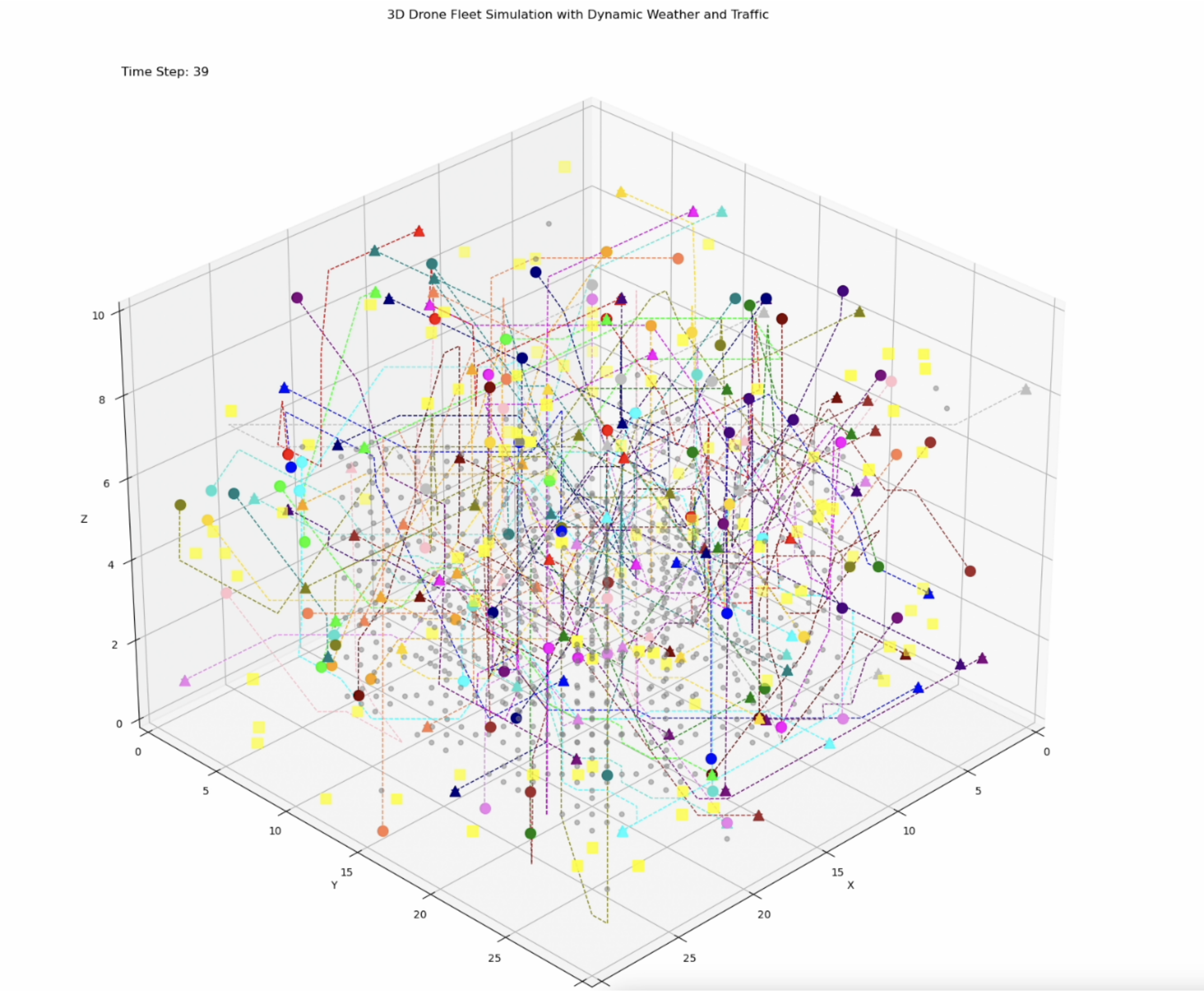

Figure 6: Ending Trajectories of 100 Drones