Clustered Temporal Path Planning (CTPP) for Drone Swarms in Real-time Updated Digital Twins

Berk Cetinsaya[1], Reese Gallagher[1], Carsten Neumann[1], Dirk Reiners[1], and Carolina Cruz-Neira[1]

University of Central Florida, Orlando FL 32816, USA,
{berk.cetinsaya,reese.gallagher,carsten.neumann,dirk,carolina}@ucf.edu

Abstract. This paper presents the Clustered Temporal Path Planning (CTPP) algorithm, designed to optimize drone swarm operations in dynamically updated digital twin environments. CTPP integrates the DBSCAN clustering technique with trajectory planning, leveraging time data to identify and cluster regions likely to become outdated. The algorithm calculates the necessary waypoints within each cluster and uses a Traveling Salesman Problem (TSP) solver to determine the optimal path, guiding drones efficiently while minimizing unnecessary travel. This approach enhances system efficiency and reduces 3D update latency. We evaluated CTPP through three experimental sets, varying area sizes and drone counts, including scenarios with fixed drone numbers and increasing area sizes to test scalability. The results show that CTPP significantly outperforms existing algorithms, achieving improvements of 66.47% and 18.41%, respectively. This leads to a substantial improvement in reducing the 3D update latency of digital twin environments.

Keywords: Coverage path planning, Clustering, Autonomous control, Unmanned aerial vehicles

1 Introduction

With the growing deployment of drone swarms in a variety of applications, including environmental monitoring [1, 2], disaster response [3, 4], and surveillance [5, 6], the demand for efficient and real-time path planning has become increasingly important. As drone swarms operate in complex and dynamic environments, their ability to update and adapt their trajectories in real-time is essential for maximizing mission success and minimizing operational delays. In recent years, digital twin technology has emerged as a powerful tool for real-time monitoring and control [7], providing a virtual replica of the physical environment in which drones operate. However, managing the vast amounts of data and ensuring timely updates to the digital twin while optimizing the drones' trajectories remains a significant challenge.

Traditional coverage path planning (CPP) algorithms have been widely utilized in drone operations to ensure that all areas of interest are effectively covered [8, 9]. These algorithms typically focus on optimizing a single pass over the operational area by dividing it into manageable segments and generating paths that ensure full coverage while minimizing redundancy and travel time. On the other hand, persistent monitoring and multi-pass coverage strategies have been introduced for dynamic and real-time environments where traditional CPP methods may not be suitable [16]. These techniques focus on revisiting certain locations multiple times rather than covering an area only once. This is particularly useful for applications such as surveillance, environmental monitoring, and search and rescue, where continuous updates from specific areas are necessary. The main challenge is balancing the frequency of revisits to critical areas with the efficiency of path planning to optimize time and resources. In rapidly changing environments, such as those represented by digital twins requiring constant updates, traditional CPP methods, and persistent monitoring strategies are not fully equipped to manage the timing and frequency of revisits effectively. While traditional CPP provides complete coverage, it does not address the need for timely information updates in dynamic settings. Persistent monitoring, which focuses on revisiting areas repeatedly, also faces challenges in balancing revisit frequency with efficient resource use. Both approaches fall short in environments where continuous updates are essential to track evolving conditions.

To address these challenges, this paper proposes the Clustered Temporal Path Planning (CTPP) algorithm that focuses on repeated passes over the environment, prioritizing the minimization of information age. CTPP integrates the Density-based Spatial Clustering of Applications with Noise (DB-SCAN) algorithm with advanced trajectory planning to enhance the performance of drone swarms in environments that are continuously updated in real time. The core idea of CTPP is to leverage stored time data to identify regions likely to become outdated subsequently. By identifying these "future oldest areas" and clustering them, the algorithm generates waypoints within each cluster based on the drones' coverage area. These waypoints are designed to ensure comprehensive coverage of outdated regions. These waypoints are then used in conjunction with a Traveling Salesman Problem (TSP) solver to determine the optimal path for the drone swarm. The integration of clustering and TSP solving in CTPP provides several advantages over traditional path planning algorithms. CTPP ensures that all areas are revisited in an efficient manner, keeping the digital twin up-to-date by systematically scheduling passes based on the state of the environment. By focusing on areas likely to become outdated, CTPP prioritizes drone resources effectively, minimizing unnecessary travel and reducing overall mission time while keeping the digital twin updated in real time.

The main contributions of the proposed algorithm are:

- CTPP enhances the process of identifying the next areas that require updates by grouping these regions into clusters using the DB-SCAN algorithm. This clustering approach ensures that the drone swarm can efficiently target multiple outdated areas within close proximity, reducing overall travel time.
- By calculating the number of waypoints based on the drones' coverage area in the identified clusters, CTPP generates optimized waypoints to be used in a TSP solver to determine the most efficient path.

- The CTPP algorithm significantly reduces latency in 3D updates by focusing on minimizing the return route after updating the oldest identified area. This leads to more timely updates across the entire digital twin environment, improving the overall performance of drone swarm operations.

The remainder of this paper is organized as follows: Section 2 provides a review of related work in the field of drone swarm path planning and digital twin technology. Section 3 details our newly proposed algorithm CTPP. Section 4 presents the experimental setup and results, highlighting the advantages of CTPP in various scenarios. Finally, Section 5 concludes the paper with a discussion of future work and potential applications of the CTPP algorithm.

2 Related Works

Traditional Coverage Path Planning (CPP) algorithms are designed to ensure that an entire area of interest is comprehensively covered by a drone or drone swarm. Early methods, such as grid-based [12] and sweeping algorithms [13], focused on dividing the operational area into smaller segments or cells, and systematically covering them to ensure complete area coverage. The work done by Choset et al. [11] on generalized CPP techniques has provided a foundation for many applications, particularly in structured environments where the area can be easily divided into manageable sections. However, these methods often fall short in dynamic environments, where certain areas may require more frequent updates due to changes over time, leading to inefficiencies when applied to real-time scenarios. Persistent monitoring and multi-pass coverage approaches have been increasingly recognized as essential for dynamic, real-time environments where traditional single-pass CPP methods are insufficient. Feng and Katupitiya [17] proposed a method that balances persistent and complete coverage by using a Minimum Spanning Tree (MST)-based recursive algorithm to estimate the upper bound of UAVs needed for multi-pass coverage with frequency constraints. Despite its advantages in efficiently covering dynamic environments, this approach is limited by its reliance on predefined time thresholds and static goal prioritization, which may not fully adapt to rapidly changing environmental conditions. Moreover, Kusnur et al. [18] present a planning framework for persistent coverage using multiple UAVs, addressing the challenge of maintaining desired coverage levels in environments where information decays over time. The method employs a continuous cycle of goal assignment and globally deconflicting, kinodynamic path planning. While it provides robust persistent coverage, the decoupled goal assignment and planning process can be suboptimal, and the framework struggles with planning delays in high-dimensional state spaces.

Cetinsaya et al. [14] presented novel autonomous drone control methods designed to update the digital twin (DT) of a region of interest in real time using one or multiple drones. In their approach, each region within the DT logs the timestamp of its most recent update. Utilizing this timestamp information, their algorithms autonomously direct one or more drones to the areas of the DT that are most outdated. The authors proposed various algorithms tailored to different control complexities and objectives. The *sequential control* algorithm, for instance, identifies the drone closest to the DT region with the oldest update and directs it to that location to perform the necessary update. This method ensures that only one drone is in motion at any given time, simplifying the control algorithm and minimizing drone movement. However, it may also lead to delays in updating certain segments of the DT. In addition, the authors [15] also introduced a control method, CALF-DLFO, that incorporates two methods: adaptive formation control and dynamic leader-following optimization. In the *adaptive formation control* approach, they employed a pyramid-like formation algorithm to organize the drones. This setup ensures that drones with larger coverage areas are positioned centrally within the swarm, while those with smaller coverage are placed towards the edges. After arranging the drones, the *dynamic leader-following optimization* technique is used to manage the swarm. The lead drone is directed to the next target using the previously mentioned sequential control algorithm, while the follower drones adjust their positions within the formation according to the leader's movements. However, this method may become inefficient when managing larger swarms due to its limitations.

In summary, while persistent coverage methods enhance traditional CPP algorithms by integrating time-sensitive revisits, challenges persist in optimizing path planning for real-time adjustments and adapting to unpredictable environmental changes. Significant improvements have been made in both CPP and time-based optimization; however, merging these approaches into a unified algorithm remains an evolving area of research. The Clustered Temporal Path Planning (CTPP) algorithm addresses this gap by combining the strengths of DB-SCAN clustering and time-based path planning to optimize drone swarm operations in real-time, dynamic environments.

3 Clustered Temporal Path Planning (CTPP)

The Clustered Temporal Path Planning (CTPP) algorithm is designed to enhance the efficiency of drone swarm missions by optimizing the path based on real-time data updates. Unlike previous methods where drones navigate directly to the oldest point, CTPP evaluates potential intermediate points that will soon become outdated, optimizing the overall coverage and minimizing the update latency. This section details the problem formulation and steps in implementing the CTPP algorithm.

3.1 Problem Statement

The core problem addressed by the CTPP algorithm is the optimization of the path taken by a drone to update regions in a dynamic environment. In such environments, different regions become outdated over time, and it is important to efficiently determine a path that minimizes the overall update latency.

Given a set of regions $R = \{r_1, r_2, \dots, r_n\}$, where each region $r_i = (L_i, T_i)$ is characterized by its location L_i and a timestamp T_i, indicating when it was last updated, the problem involves determining the most efficient path that a drone should follow to update these regions.

Formally, let:

- $Age(r_i, t) = t - T_i$ represents the time passed since the last update of the region r_i at time t.
- L_d represents the drone's initial position.
- L_o represents the location of the **current oldest region**, i.e., the region that has the largest age.
- L_s represents the location of the **subsequent oldest region**, i.e., the next region in line to become outdated, has the second largest age.

To illustrate the problem, consider two paths the drone can take:
1. $L_d \to L_o \to L_s$ (abbreviated $\overrightarrow{dos}$) or 2. $L_d \to L_s \to L_o$ (abbreviated $\overrightarrow{dso}$)
These paths are shown below in Fig. 1.

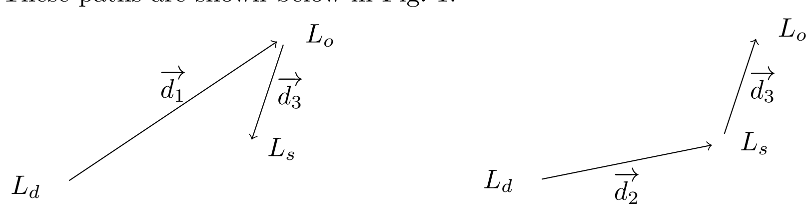

Fig. 1: From left to right, dos (Oldest Time First) and dso (Subsequent Oldest Time Heuristic).

A drone following the Oldest Time First (OTF) strategy, where it always heads toward the current oldest region, will take the path $\overrightarrow{dos}$. In contrast, a drone using the Subsequent Oldest Time Heuristic (SOTH), where a nearby subsequent oldest region may be visited first, will follow the path $\overrightarrow{dso}$. The total average age of the regions at any time t is given by:

$$\text{Age}_{\text{ave}}(t) = \frac{1}{n} \sum_{i=1}^{n} Age(r_i, t) = \frac{1}{n} \sum_{i=1}^{n} (t - T_i)$$

where $Age(r_i, t) = t - T_i$ represents the age of region r_i at time t. If the drone visits a region, its age resets to zero. Note that the drone travels at a constant speed, so the distances d_1, d_2, and d_3 correspond directly to the time spent traveling between regions, as distance = speed $\times$ time. Therefore, we will use the magnitudes of $\overrightarrow{d_1}$, $\overrightarrow{d_2}$, and $\overrightarrow{d_3}$ to represent both the distances and the time taken by the drone to travel between the regions.

Oldest Time First (OTF) For the OTF path, the drone follows $L_d \to L_o \to L_s$. After traveling to the oldest region L_o, the drone spends time equivalent to the distance d_1, resetting $\text{Age}(L_o, t)$ to 0. During this time, the ages of regions

L_s and L_n continue to increase by d_1. After traveling from L_o to L_s, the total time spent is $d_1 + d_3$, and the age of L_s is reset to 0. Thus, the total average age of the regions after following the OTF path is:

$$\mathrm{Age}_{\mathrm{OTF}}(d_1 + d_3) = \frac{(t - T_d) + (t - T_o) + (t - T_s)}{3} = \frac{3(d_1 + d_3) + T_d + T_o + T_s}{3}$$

Subsequent Oldest Time Heuristic (SOTH) For the SOTH path, the drone follows $L_d \to L_s \to L_o$. After traveling to L_s, the drone spends time equivalent to the distance d_2, resetting $\mathrm{Age}(L_s, t)$ to 0. During this time, the ages of regions L_d and L_o continue to increase by d_2. After traveling from L_s to L_o, the total time spent is $d_2 + d_3$, and the age of L_o is reset to 0. Thus, the total average age of the regions after following the SOTH path is:

$$\mathrm{Age}_{\mathrm{SOTH}}(d_2 + d_3) = \frac{(t - T_d) + (t - T_s) + (t - T_o)}{3} = \frac{3(d_2 + d_3) + T_d + T_s + T_o}{3}$$

Comparison To compare the two strategies, if $d_2 < d_1$, then:

$$\mathrm{Age}_{SOTH}(d_2 + d_3) < \mathrm{Age}_{OTF}(d_1 + d_3)$$

This shows that if the distance to the subsequent oldest region L_s is shorter than the distance to the oldest region L_o, the SOTH path will result in a lower average age.

3.2 Clustering The Subsequent Oldest Points

After identifying the oldest region, the algorithm looks for the subsequent oldest region. Then, the algorithm uses the DB-SCAN clustering algorithm to group the subsequent oldest regions based on their spatial proximity. This clustering reduces the number of intermediate regions the drone needs to consider by grouping nearby regions together. For each cluster, the algorithm calculates the axis-aligned bounding box (AABB) that tightly encloses the cluster, and the area of this bounding box, C_a, is used to determine the necessary number of waypoints for coverage. Let the coverage area of the drone D_a. The number of waypoints W needed to cover a cluster C_a is determined by:

$$W = \left\lceil \frac{C_a}{D_a} \right\rceil$$

This formula ensures that the entire cluster is covered by the drone's field of view, optimizing the path planning process. After calculating the waypoints, the algorithm further refines the process by removing any waypoints that do not cover any region within the cluster. This refinement ensures that the drone only visits waypoints necessary for covering points in the cluster, improving efficiency and reducing unnecessary movement.

3.3 Traveling Salesman Problem Solver

Finally, the waypoints generated from the clusters are used as inputs to a Traveling Salesman Problem (TSP) solver, which determines the most efficient route that visits all waypoints and the oldest point. The heuristic TSP solver iteratively checks each cluster, starting with the one closest to the drone's current position, and determines if visiting the cluster reduces the total path distance:

1. Find the closest cluster to the current position.
2. Solve the TSP for this cluster to determine the optimal path through the waypoints.
3. Calculate the distance from the current position to the closest waypoint and add the TSP distance between waypoints within this cluster.
4. Compare this total distance to the distance from the start position to the oldest point:
   (a) If the TSP total distance is greater than the distance to the oldest point, exclude this cluster.
   (b) If the TSP total distance is less than or equal to the distance to the oldest point, include this cluster. Then, identify the last traveled waypoint in this cluster as the current position.
5. Repeat the process from step 1 until all clusters have been evaluated.

3.4 Dynamic Leader Changes and Boundary Checks

To optimize further, we introduce a dynamic leader change mechanism for CALF-DLFO with the CTPP. In the original CALF-DLFO algorithm, the leader drone was selected at the beginning of the simulation based on design considerations, and this leader remained unchanged throughout the mission. However, with CTPP, the leader drone is now selected dynamically during the mission based on proximity to the oldest point. Specifically, the edge drone closest to the oldest point at any given moment becomes the new leader, ensuring that the swarm's path is always optimized for reaching the oldest areas. Additionally, the original CALF-DLFO did not incorporate boundary checks within the environment, leading to inefficiencies, especially with an increased number of drones. Drones positioned outside the boundaries would not contribute effectively to updating the environment, rendering them inefficient. To address this, we integrated boundary checks into CTPP. Whenever a follower drone's formation position is outside the boundary, the algorithm adjusts the formation direction, ensuring all drones remain within the environment's boundaries. This modification ensures that every drone contributes to reducing the overall update latency.

4 Experiments and Results

4.1 Experimental Setup

In this study, we conducted a series of simulations to compare the performance of CTPP with state-of-the-art time-based control methods, sequential control, and

CALF-DLFO. The primary objective was to determine whether the CTPP offers significant improvements over the previous algorithms across various scenarios. The simulations were implemented using the Unity game engine [10], providing a robust and flexible environment for testing and visualizing the algorithms (see Fig. 2). Three distinct experiments were designed to evaluate the algorithms with varying area sizes and numbers of drones. The area sizes ranged from 3000 m^2 to 10,000 m^2, increasing in steps of 1000 m^2. The number of drones was adjusted based on the area size. For instance, the 3000 m^2 area had two scenarios with 2 and 3 drones, the 4000 m^2 area had three scenarios with 2, 3, and 4 drones, and this pattern continued up to the 10,000 m^2 area. To further assess the scalability of the new algorithm, an additional set of scenarios was tested where the number of drones was fixed at 10, while the area size was incrementally increased from 15,000 m^2 to 50,000 m^2 in steps of 5000 m^2.

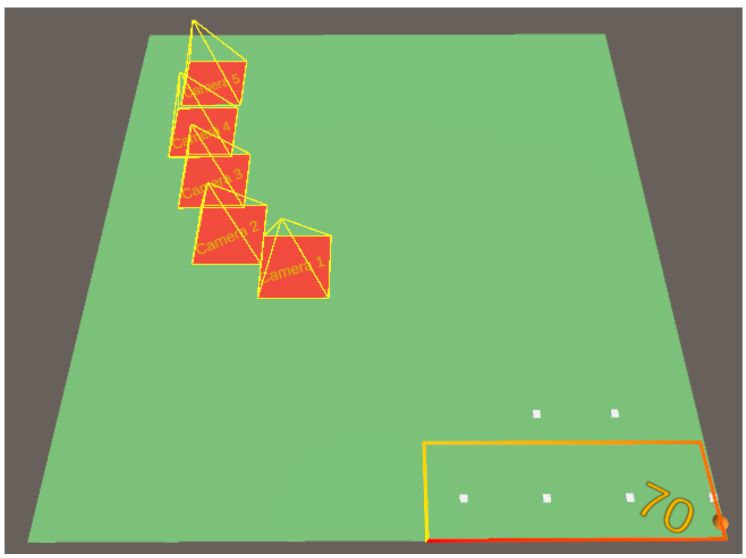

Fig. 2: A sample simulation run involves 5 drones covering an area of 8000 m^2. Orange squares represent the drones' coverage areas, while the orange sphere labeled "70" marks the oldest region with an age of 70 seconds. The rectangle represents the cluster's bounding box, and the white cubes indicate the waypoints generated based on the drones' coverage areas.

Each simulation consists of two phases: the initial phase and the CTPP phase. In the initial phase, since none of the regions have been observed yet, waypoints are randomly generated to cover the entire area. Once every region has been visited and updated at least once, the simulation transitions to the CTPP phase. During the CTPP phase, the CTPP algorithm is employed. Due to the randomized nature of the initial phase, three separate experiments were conducted with different random seeds to ensure the robustness and consistency of the results. In the scenarios considered, while the area size and the number of drones vary, all drones maintain the same field of view (FOV) and ground sampling distance (GSD). This consistent setup ensures that the only variable impacting the results is the algorithm itself, allowing for a precise evaluation of its effectiveness without interference from differences in drone specifications.

4.2 Metrics

Sequential control, CALF-DLFO, and CTPP were evaluated in each scenario, with performance metrics recorded for direct comparison. The paired data resulting from these experiments allowed for statistical analysis to assess the differences between the algorithms. The differences between the paired data in each scenario were first tested for normality using the Shapiro-Wilk test to determine the appropriate statistical tests for subsequent analysis. Based on the normality results, either a paired t-test or a Wilcoxon test was conducted to evaluate whether the CTPP's performance differed significantly from existing methods, providing t-statistics and p-values for each experiment. To assess the magnitude of these differences, Cohen's d was calculated, quantifying the practical significance of the performance variation between algorithms. Additionally, bootstrapping was employed to ensure the robustness of the statistical results, generating confidence intervals for the t-statistics, p-values, and Cohen's d, offering a more reliable estimate of variability and uncertainty. Finally, a meta-analysis was performed across all experiments, using Fisher's method to combine p-values and a weighted average of Cohen's d to derive an overall effect size, providing a comprehensive evaluation of CTPP's performance.

4.3 Comparison with Other Methods

To contextualize the performance of the CTPP, it was compared against the sequential control method and CALF-DLFO across three independent experiments. The results consistently showed a statistically significant improvement in favor of the CTPP.

The statistical analysis revealed a significant difference favoring the CTPP algorithm over the sequential algorithm as shown in Table 1. The paired t-test produced a highly significant t-value of 17, with a corresponding p-value of 3.93×10^{-21}, well below the conventional significance threshold. The effect size, as measured by Cohen's d, was large at 1.6, indicating a substantial advantage of the CTPP algorithm over sequential. On average, the CTPP algorithm outperformed the sequential method by 66.47%. Bootstrap analysis further confirmed these findings, providing confidence intervals for the t-statistic, p-value, and effect size that consistently supported the superiority of CTPP. For example, the bootstrap confidence interval for Cohen's d ranged from 1.347 to 2.138, reinforcing the strong performance gain observed.

In the comparison between CTPP and CALF-DLFO, the Wilcoxon test also showed a significant improvement, with a t-value of 24 and a p-value of

Table 1: Comparison of Sequential vs. CTPP and CALF-DLFO vs. CTPP Test Results

Metric	Sequential vs. CTPP (Paired t-test)	CALF-DLFO vs. CTPP (Wilcoxon)
T-Value	1.7×10^1	2.4×10^1
P-Value	3.93×10^{-21}	8.66×10^{-11}
Cohen's d		
Effect Size	1.6	0.52
Standard Error	0.25	0.22
Bootstrap		
Mean T-Statistic	17.936	25.716
95% CI for T-Statistic	[14.660, 22.248]	[0.000, 86.000]
Mean P-Value	3.6×10^{-19}	4.299×10^{-8}
95% CI for P-Value	$[3.22 \times 10^{-25}, 2.47 \times 10^{-18}]$	$[1.14 \times 10^{-13}, 1.59 \times 10^{-7}]$
Mean Cohen's d	1.676	0.533
95% CI for Cohen's d	[1.347, 2.138]	[0.367, 0.732]

8.66×10^{-11}. The effect size, represented by Cohen's d, was moderate at 0.52, suggesting that CTPP outperforms CALF-DLFO, though the effect is less pronounced than in the sequential vs. CTPP comparison. On average, CTPP demonstrated an improvement of 18.41% over CALF-DLFO.

Finally, a meta-analysis combining these results yielded an overall effect size of Cohen's d $= 1.006$, with a 95% confidence interval of [0.823, 1.189]. The combined p-value from Fisher's test was 2.422×10^{-29}, conclusively demonstrating CTPP's enhanced performance across different conditions.

Additionally, the performance comparison across larger area sizes, ranging from 15,000 m^2 to 50,000 m^2, with 10 drones, highlights the superiority of CTPP. CTPP consistently achieved the lowest update times, reducing the times by 48.6% to 61.5% compared to the sequential algorithm, and outperforming CALF-DLFO by 6.1% to 29.0%. These results demonstrate that CTPP provides better performance across all tested area sizes.

These results collectively demonstrate that the CTPP algorithm offers significant and practically meaningful improvements over both sequential and CALF-DLFO algorithms, as confirmed through statistical analysis.

4.4 Limitations and Future Work

Despite the promising results demonstrated by the CTPP algorithm, several limitations must be acknowledged. First, the experiments were conducted in a simulated environment using Unity, which, while controlled, does not fully replicate real-world complexities such as sensor noise, communication delays, or hardware failures. This will be explored in future work through validation in real-world scenarios. Second, as the number of drones and area size increase, the algorithm's computational demands grow, particularly in the clustering and TSP-solving processes. While effective in the tested scenarios, future research could explore optimizations to enhance scalability for larger deployments. Finally, the environments tested focused primarily on varying area sizes, without incorporating additional real-world complexities such as obstacles or altitude variations. While this provides a controlled basis for evaluating the algorithm's core performance, future work could explore its effectiveness under more dynamic environmental conditions. Addressing these limitations in future work will be important to enhance CTPP's practical applicability in diverse and complex real-world settings.

5 Conclusion

In this paper, we introduced the Clustered Temporal Path Planning (CTPP) algorithm to improve drone swarm efficiency in real-time, dynamic environments. CTPP combines temporal data analysis, DB-SCAN clustering, and trajectory optimization via a Traveling Salesman Problem (TSP) solver, enabling drones to prioritize time-sensitive regions and minimize unnecessary travel. This approach addresses the limitations of traditional coverage path planning (CPP) and persistent coverage methods, which struggle with dynamic updates. Experiments across various scenarios showed that CTPP significantly improves system efficiency and reduces 3D model update latency compared to existing methods. CTPP's dynamic leader changes and boundary checks ensure all drones contribute effectively, even in constrained environments. These results highlight CTPP's potential for optimizing drone swarm operations in applications such as surveillance and environmental monitoring.
Future work could extend CTPP to handle more complex environments, heterogeneous drone swarms, and real-time adaptation, with further optimization of clustering and TSP-solving for large-scale deployments. Overall, CTPP is a significant step forward in temporal-based path planning for efficient, adaptive, and scalable drone swarm operations.

References

1. Masi, G. & Ferrante, E. Quality-dependent adaptation in a swarm of drones for environmental monitoring. *2020 Advances In Science And Engineering Technology International Conferences (ASET)*. pp. 1-6 (2020)
2. Schilling, F., Schiano, F. & Floreano, D. Vision-Based Drone Flocking in Outdoor Environments. *IEEE Robotics And Automation Letters*. **6** pp. 2954-2961 (2021)
3. Gladence, L., Anu, V., Anderson, A., Stanley, I., J, J. & Revathy, S. Swarm Intelligence in Disaster Recovery. *2021 5th International Conference On Intelligent Computing And Control Systems (ICICCS)*. pp. 1-8 (2021)
4. Kurt, A., Saputro, N., Akkaya, K. & Uluagac, A. Distributed Connectivity Maintenance in Swarm of Drones During Post-Disaster Transportation Applications. *IEEE Transactions On Intelligent Transportation Systems*. **22** pp. 6061-6073 (2021)

5. Venturini, F., Mason, F., Pase, F., Chiariotti, F., Testolin, A., Zanella, A. & Zorzi, M. Distributed Reinforcement Learning for Flexible and Efficient UAV Swarm Control. *IEEE Transactions On Cognitive Communications And Networking.* **7** pp. 955-969 (2021)
6. Bandarupalli, A., Swarup, D., Weston, N. & Chaterji, S. Persistent Airborne Surveillance using Semi-Autonomous Drone swarms. *Proceedings Of The 7th Workshop On Micro Aerial Vehicle Networks, Systems, And Applications.* (2021)
7. Mihai, S., Yaqoob, M., Hung, D., Davis, W., Towakel, P., Raza, M., Karamanoğlu, M., Barn, B., Shetve, D., Prasad, R., Venkataraman, H., Trestian, R. & Nguyen, H. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Communications Surveys & Tutorials.* **24** pp. 2255-2291 (2022)
8. Fevgas, G., Lagkas, T., Argyriou, V. & Sarigiannidis, P. Coverage Path Planning Methods Focusing on Energy Efficient and Cooperative Strategies for Unmanned Aerial Vehicles. *Sensors (Basel, Switzerland).* **22** (2022)
9. Hu, W., Yu, Y., Liu, S., She, C., Guo, L., Vucetic, B. & Li, Y. Multi-UAV Coverage Path Planning: A Distributed Online Cooperation Method. *IEEE Transactions On Vehicular Technology.* **72** pp. 11727-11740 (2023)
10. *Real-Time 3D Development Platform & Editor.* Unity. https://unity.com/products/unity-engine. Accessed: 2024-09-02.
11. Choset, H. Coverage for robotics – A survey of recent results. *Annals Of Mathematics And Artificial Intelligence.* **31**, 113-126 (2001,10), https://doi.org/10.1023/A:1016639210559
12. Moravec, H. & Elfes, A. High resolution maps from wide angle sonar. *1985 IEEE International Conference On Robotics And Automation Proceedings.* **2** pp. 116-121 (1985,3), https://ieeexplore.ieee.org/abstract/document/1087316
13. Luo, C. & Yang, S. A real-time cooperative sweeping strategy for multiple cleaning robots. *Proceedings Of The IEEE Internatinal Symposium On Intelligent Control.* pp. 660-665 (2002,10), https://ieeexplore.ieee.org/abstract/document/1157841, ISSN: 2158-9860
14. Cetinsaya, B., Neumann, C., Reiners, D. & Cruz-Neira, C. Real-time Updated Digital Twins for Drone Swarm Command and Control. *Interservice/Industry Training, Simulation, And Education Conference (I/ITSEC).* (2023)
15. Cetinsaya, B., Neumann, C., Bruder, G., Reiners, D. & Cruz-Neira, C. Camera-based Adaptive Line Formation and Dynamic Leader-Following Optimization (CALF-DLFO) for Drone Swarms in Real-time Updated Digital Twins. *2024 10th International Conference On Control, Decision And Information Technologies (CoDIT).* pp. 2769-2774 (2024,7), https://ieeexplore.ieee.org/document/10708152, ISSN: 2576-3555
16. Jing, T., Wang, W., Wang, T., Li, X. & Zhou, X. Dynamic Control Scheme of Multiswarm Persistent Surveillance in a Changing Environment. *Computational Intelligence And Neuroscience.* **2019**, 6025657 (2019), https://onlinelibrary.wiley.com/doi/abs/10.1155/2019/6025657, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2019/6025657
17. Feng, L. & Katupitiya, J. UAV-based persistent full area coverage with dynamic priorities. *Robotics And Autonomous Systems.* **157** pp. 104244 (2022,11), https://www.sciencedirect.com/science/article/pii/S0921889022001415
18. Kusnur, T., Mukherjee, S., Saxena, D., Fukami, T., Koyama, T., Salzman, O. & Likhachev, M. A Planning Framework for Persistent, Multi-UAV Coverage with Global Deconfliction. *Field And Service Robotics.* pp. 459-474 (2021)