Dynamics-Aware Fast Multi-Drone Exploration of Unknown Environments

Juan Zacarías Bardají
DIIS-i3A
Universidad de Zaragoza
Zaragoza, Spain
757279@unizar.es

David Morilla-Cabello
DIIS-i3A
Universidad de Zaragoza
Zaragoza, Spain
davidmc@unizar.es

Eduardo Montijano
DIIS-i3A
Universidad de Zaragoza
Zaragoza, Spain
emonti@unizar.es

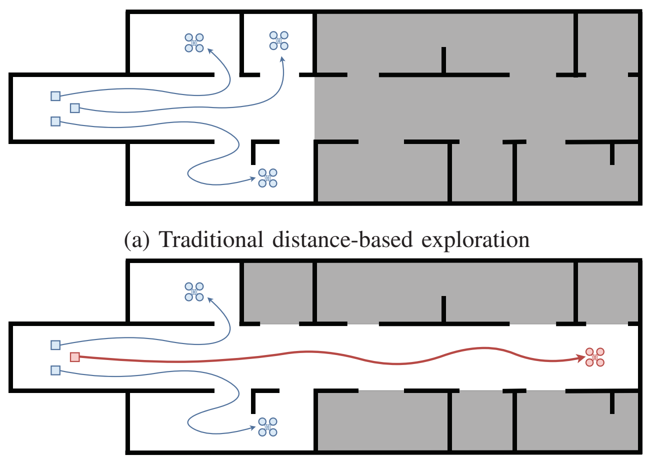
***Abstract*—Tasks such as surveillance, search and rescue operations, or disaster mapping require exploring unknown environments in the shortest possible time. Furthermore, when these tasks are performed by multiple robots, team coordination is crucial to achieve the desired efficiency. In this spirit, this paper proposes a method that introduces notions about each robot's dynamics into the planning process. We formulate novel costs into the Vehicle Routing Problem (VRP) to maintain each drone's inertia, required for efficiency. The costs of exploring regions are calculated based on the individual state of each robot, such as their speed and orientation, to favour distinct behaviours. The proposed solution is implemented on RACER, a state-of-the-art decentralized exploration system. The original implementation of RACER is modified to integrate our method, including a different optimization library to solve the VRP. To validate our approach, we conduct exhaustive experiments in four maps, with two and four agents and varying initial conditions, comparing our proposal with the state-of-the-art method. The results illustrate the advantages of incorporating knowledge about the team's dynamics into the planning process, demonstrating improvements in the efficiency and effectiveness of exploration in unknown environments.**

***Index Terms*—drone, exploration, coordination, dynamics**

I. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, play a pivotal role in a multitude of applications. Their great mobility enables vantage points for tasks such as surveillance, search and rescue operations, or exploration of unknown spaces [1]–[3]. In time-critical tasks, such as search and rescue, it is of vital importance that all operations are performed in an efficient and coordinated manner. Drones are mainly constrained by their reduced flight time. Therefore, it is important to ensure efficient planning, such as reducing still hovering, to maximize utilisation of the battery.

Achieving efficient exploration of previously unknown environments using UAVs requires overcoming several challenges, including path planning, coordination among multiple agents, and robust navigation. A number of state-of-the-art exploration approaches exist, such as [3]–[5]. However, these approaches do not consider the dynamics of each agent when planning. The efficiency of drones highly depends on inertia changes during the flight, benefiting from swift flights instead of slow hovering and abrupt turns [6].



(a) Traditional distance-based exploration

(b) Our dynamics-aware exploration

Fig. 1: Behaviors of traditional (1a) and our (1b) exploration methods. Bright and dark regions represent explored and unexplored areas, respectively. In 1a, the areas to explore are assigned based on the traversed distance by drones. Our method aims to leverage drone behaviours based on current drones' dynamics. For example, if the red drone moves with fast dynamics, it prioritizes exploring the whole corridor to maintain its high velocity, while the blue drones explore the remaining rooms with slower motions as depicted in 1b.

This paper proposes the development of a collaborative exploration approach that leverages drone-specific dynamics for coordination, ultimately improving mission performance. Based on the principle that maintaining drone dynamics is beneficial, we argue that considering dynamics in exploration can improve exploration efficiency. For example, when exploring unknown areas, some agents might maintain their dynamic behaviour in exchange for missing certain regions. Other agents then target the remaining regions while planning in already explored space. This is achieved by incorporating notions about each agent's dynamics by means of different costs in a Vehicle Routing Problem (VRP). The VRP solves the dynamics-aware task assignment for exploring the environment, enabling collaborative behaviour in the team. To implement our algorithm, we extended RACER [7], a state-of-the-art exploration method. Then, we validate our approach by extensive evaluation of the exploration performance.

II. Related work

Robotic exploration has been a topic of extensive research aimed at enhancing the ability of robots to efficiently gather information about a previously unknown environment.

The use of drones for exploration is a common practice due to their mobility and speed when traversing space. Most exploration strategies are based on the concept of frontiers [6], [8], defined as the boundaries between explored and unexplored areas [9]. In order to improve the efficiency of the planning, Rapidly-exploring Random Trees [10] are a common approach [11]–[13]. Methods that aim to enhance efficiency in planning encompass a range of techniques. Some consider surface frontiers to sample viewpoints only where high information gain is expected [11]. Others propose a specific exploration strategy [14] or focus on the safety of both the robot and the environment, with the objective of enabling the most efficient exploration possible [15], [16]. The importance of dynamics in drone exploration has been considered before in low-level planning [6]. However, its application to high-level planning and multi-robot coordination remains unexplored. All the previous approaches consider single drone exploration. The use of multiple drones is beneficial in terms of performance, but it also introduces the problem of coordinating the robot team.

The problem of multi-drone exploration requires solving the coordination between the agents. This can be solved by assigning regions to explore based on distances to the drones as the environment is discovered [7]. By not accounting for dynamics, the agents might inefficiently change their inertia. The assignment might be more intelligent with different operating modes that balance exploration and exploitation [5]. However, this method does not explicitly account for cooperation between different drone's dynamics. Other methods try to first obtain a global estimate of the map to generate efficient trajectories and also perform global task assignment [4]. However, this method is centralized and does not adapt to each agent's dynamics.

III. Method

A. Overview

The aim of our method is to efficiently explore an unknown environment by taking advantage of the distinct dynamics exhibited by the agents within a robot team.

In order to achieve this, exploration tasks are allocated adaptively, taking into account the current dynamics of each agent and adjusting the assignment accordingly. For instance, robots with current high velocities can be assigned to cover a greater distance faster in a straighter path, reducing changes in speed and direction, and resulting in a high level of coverage but possibly missing regions. Conversely, robots with current low velocities can be assigned to the areas left behind by the other drones so that they can explore them. An illustrative example can be found in Figure 1. By incorporating information about the drone dynamics of the team into the planning process, our approach aims to favour behaviours for each agent to enhance exploration efficiency.

B. Problem definition

Consider the unknown bounded environment to explore modelled as a voxel map, where each voxel is denoted by $m \in \mathcal{M}$. A team of drones, $k \in \mathcal{K}$, moves with depth sensors to observe the environment. The final goal of the exploration is to obtain the occupancy value of all the unexplored m_u. For that, agents must coordinate to explore different regions of the map. The environment is partitioned in a grid with cells $h \in \mathcal{H}$, containing a set of unexplored voxels, $\mathcal{M}_h$. Denote $\mathcal{H}^k$ as the set of grid cells that agent k has to explore. The goal of the coordination module is to assign $h \in \mathcal{H}^k$ to maximize the efficiency of the mission.

Assuming a decentralized setup with range-limited communication between agents, a common approach is to solve the assignment problem via pair-wise interaction, simplifying the task allocation into sub-problems such that

$$\min_{\mathcal{H}^1, \mathcal{H}^2} C^1(\mathcal{H}^1) + C^2(\mathcal{H}^2), \tag{1}$$

where C denotes the cost of each agent for exploring each set of locations.

C. Planning for coordination

The problem of minimizing the cost of agents traversing a set of locations is known as the Vehicle Routing Problem (VRP). We use this formulation to solve (1). The VRP is defined over an undirected graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where nodes are locations to visit, corresponding to h and edges are connections between them. Notably, the VRP defines a common initial location known as the *depot* where all agents must depart from and return. Let $\mathcal{X} = \{x_{ij}^k\}$, for $i, j \in \mathcal{V}$, be the set of decision variables that indicate whether an agent k traverses the route from i to j or not. The edges are also weighted by c_{ij}, which indicates the cost to go from location i to location j, e.g., the distance between the two locations or the time to reach one from the other. The VRP solves the following optimization problem

$$\min_{\mathcal{X}} \sum_i \sum_j c_{ij} x_{ij}^k, \quad s.t. \tag{2a}$$

$$\sum_k \sum_i x_{ij}^k = 1 \qquad \forall j \in V \setminus \{0\}, \tag{2b}$$

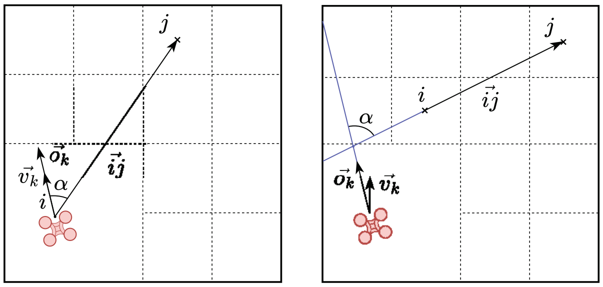
$$\sum_k \sum_j x_{ij}^k = 1 \qquad \forall i \in V \setminus \{0\}, \tag{2c}$$

$$\sum_k \sum_i x_{i0}^k = \sum_k \sum_j x_{0j}^k = |K|, \tag{2d}$$

$$\sum_{i,j \in S} x_{i,j}^k \leq |S| - 1, \quad \forall S \subset V \setminus \{0\}, S \neq \emptyset, \tag{2e}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i, j \in V, \tag{2f}$$

where (2b), and (2c) ensure that each location must be visited only once, (2d) ensures that all vehicles start and depart



(a) Distance cost (b) Orientation cost

Fig. 2: Representation of the terms that are used during cost calculation. 2a illustrates the vector and terms employed in the calculation of the cost of the agent moving from its position to j, while 2b does so with the cost of moving from i to j.

from the depot, (2e) removes possible sub-tours where the depot is not included and (2f) imposes the binary condition on the decision variables.

Two modifications of the traditional VRP are required to adapt it for drone exploration [7]. First, we relax (2d) to avoid a common depot as explained in Section III-E. Second, even if the sum of the costs is optimized, it does not ensure a balanced workload assignment to agents that can work in parallel. Therefore, we add a capacity constraint to limit each agent's workload. We assign a *demand* per node h in the VRP that corresponds to the amount of voxels to explore, $|\mathcal{M}_h|$, and add the constraint

$$\sum_{h\in\mathcal{H}^i} |\mathcal{M}_h| \leq \alpha_k \sum_{h\in\mathcal{H}} |\mathcal{M}_h|, \quad i = 1, 2, \tag{3}$$

where α_k is the percentage of the work that can be assigned to agent k. This change turns the problem into a Capacitated Vehicle Routing Problem (CVRP).

D. Dynamics cost

The conventional VRP formulation primarily relies on distance values to determine costs. The cost associated with distance represents the total distance that each vehicle (or agent) must traverse in order to complete its assigned route. This cost can be approximated based on the Euclidean distance without considering obstacles or using pathfinding algorithms such as A* to compute the true distance.

Our approach hypothesises that different agents may incur different travel costs for the same distance due to variations in their dynamics. Consequently, the cost c_{ij}^k for agent k travelling from location i to j is not only dependent on the distance cost but also on the current dynamics of agent k. High-velocity agents should be assigned to tasks requiring less directional change, maximizing their speed advantage. Conversely, tasks requiring significant directional changes should be handled by slower agents. Therefore, we consider the following dynamics-aware cost:

$$c_{ij}^k = \frac{d_{ij}}{\|\vec{v_k}\|} + \frac{C_d\left(\vec{o_k}, \vec{ij}\right)}{2} \cdot \|\vec{v_k}\| \,. \tag{4}$$

The first term in (4) encodes the velocity cost, combining the distance between cells i and j, d_{ij}, and the agent's velocity, $\vec{v_k}$. This term decreases as the velocity of the agent increases, favouring fast drones to cover long distances. The second term is a turning cost where C_d refers to the cosine distance between the drone orientation vector, $\vec{o_k}$, and the travel vector $\vec{ij}$ (see Figure 2) defined as

$$C_d(\vec{o_k}, \vec{ij}) = 1 - \frac{\vec{o_k} \cdot \vec{ij}}{\|\vec{o_k}\| \cdot \left\|\vec{ij}\right\|}. \tag{5}$$

This value is multiplied by the velocity to penalize that fast-moving drones change their direction which would result in an abrupt inertia change.

E. Implementation

To implement the proposed dynamics cost matrix we must encode all relevant information into a matrix format suitable for the optimization library employed, Or-Tools [17], [18]. The entire code structure of RACER was modified so that it works with the Or-Tools library, rather than LKH [19], the original library used by RACER, which does not allow the definition of our problem.

Figure 3 illustrates the structure of the asymmetric cost matrix implemented within the RACER framework. We explain the rationale behind each block and how to include our proposed costs. Moreover, it is important to note, that differently from RACER, our approach computes one cost matrix per drone.

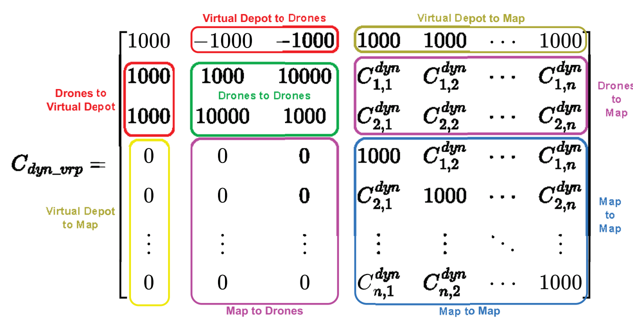
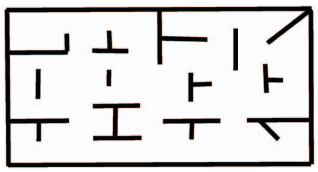
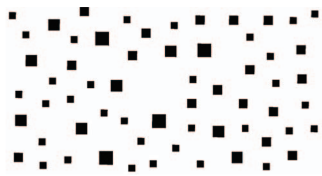


Fig. 3: Dynamics Cost Matrix implemented in RACER.

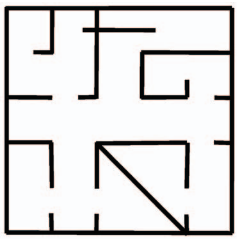
- **Drones $\leftrightarrow$ Virtual Depot**: In order to relax (2d), the depot is transformed into a Virtual Depot. A very small cost (-1000) from the depot to the drones ensures they start from their current locations. Similarly, a very high cost (1000) from the drones to the depot prevents going back.
- **Map $\leftrightarrow$ Virtual Depot**: Following from the previous, we assign a high cost from the Virtual Depot to every map cell (1000) and zero cost from every map cell to the Virtual Depot. This effectively turns the VRP into an Asymmetric VRP, where the vehicles are not required to return to the depot. The final paths are obtained by removing the edges connecting from and to the virtual depot.



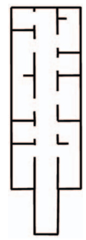
(a) Map 1: Office



(b) Map 2: Pillar

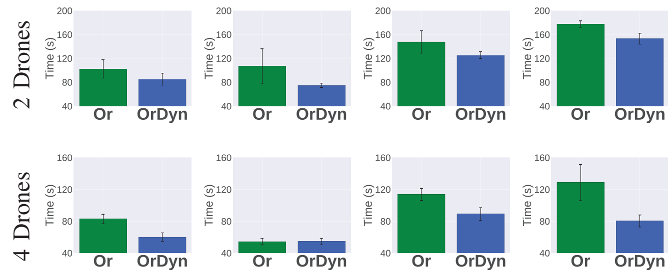


(c) Map 3: Rooms1



(d) Map 4: Rooms2

Fig. 4: Environments employed in the experiments. Map 4a and 4b belong to the experimental setup of RACER. Map 4c and 4d present two different, larger scenarios.



(a) Office (b) Pillar (c) Rooms1 (d) Rooms2

Fig. 5: Exploration times comparing **Or** and **OrDyn** for two (top) and four (down) drones in four environments with **Single Corner** configuration. The plots represent the mean and standard deviation for 5 missions. **OrDyn** outperforms **Or** in most environments and performs on par in the rest.

- **Drones ↔ Drones**: To ensure motion and avoid collisions in the beginning, these costs are set to artificially high numbers.
- **Map ↔ Map**: This is were the dynamics-aware costs are implemented. For cells within five metres, d_{ij} is computed using the A* algorithm, and the Euclidean distance otherwise. The drone velocities and orientations are those available when the matrix is created.
- **Drones ↔ Map**: Similarly, the costs to reach map cells are computed as described above using (4). Also, the cost to reach the drones is set to zero to allow the AVRP as mentioned before.

IV. Experiments & Results

The experiments aim to evaluate the efficiency and potential benefits of the implemented method, focusing on exploration performance.

A. Setup

A variety of environments were included, with different structure and size. The same simulation setup used in RACER [7] was employed, which utilizes a multi-rotor simulator and RViz for visualization. The mission was executed in four different environments. *Office* and *Pillar*, in Figures 4a and 4b, are part of the experimental setup of RACER. These environments were employed to assess the efficacy of the method in comparison to the original RACER algorithm. Additionally, *Rooms1* and *Rooms2* are two larger scenarios, as illustrated in Figures 4c and 4d, respectively.

B. Baselines & Metrics

We want to validate the potential advantages of considering the dynamics of the drones in the planning process. Therefore we compare our method, **OrDyn** (Dynamics Or-Tools), integrating the multiple cost matrices pipeline and the dynamics cost with **Or** (Or-Tools), which just accounts for distance costs and employs a single cost matrix. Per exploration mission, we measure the total exploration time [s] and the map completeness [%] over time. The mission finishes after the map completeness reaches 100%, the mission time surpasses 180s or the map completeness remains unchanged for 50 or more updates. The planning time for all the methods is negligible compared to other pipeline parts and, due to the page limit, the results were not included. Each baseline is run five times in each scene and the mean and variances for the different metrics are presented to account for stochasticity in the planning process.

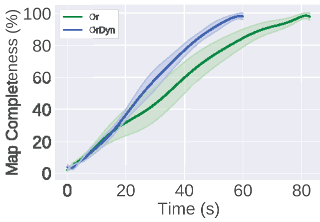
C. Exp1: Exploration Efficiency

In this experiment, we compare the exploration performance of our system with teams of two and four drones. Drones start the exploration from one corner of the map, named **Single Corner** configuration. This is considered the most realistic configuration for exploring an unknown environment, with drones initiating their journey from one side of the map.

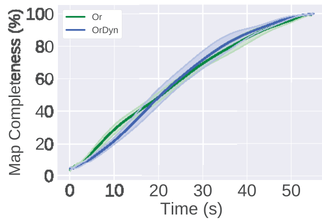
The resulting exploration times presented in Figure 5 show that our method, **OrDyn**, outperforms the non-dynamics version, **Or**, in all the environments by incorporating information about the dynamics of the team into the planning process. In environments like *Office* and *Rooms2*, where the layout allows drones to maintain their dynamics without continuous turning, a performance improvement was observed. Furthermore, in *Pillar*, with an open-like layout, and *Rooms2*, with a more complex layout, an improvement in performance can still be observed.

The experimental results indicate that an increase in the number of agents leads to enhanced coordination and overall performance. The method benefits from enhanced team dynamics with four drones, resulting in more efficient exploration.

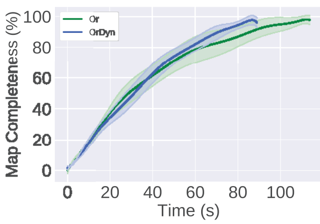
Figure 6 illustrates a comparison of map completeness in the four environments, showing that **OrDyn** outperforms **Or**. It might be observed that our method fall behind in environments like *Pillar* or *Rooms1*. This may be due to some agents initially being slower in covering missing regions, but they subsequently catch up, resulting in faster overall exploration.



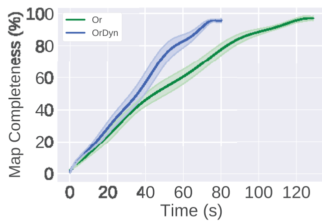
(a) Office



(b) Pillar

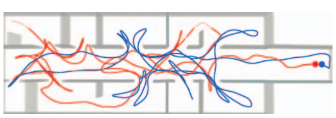


(c) Rooms1



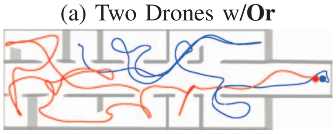
(d) Rooms2

Fig. 6: Map completeness over time for **Or** and **OrDyn** in all environments with four drones and **Single Corner** configuration. The mean and standard deviation for 5 missions is represented.

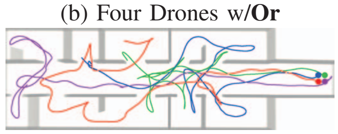


(a) Two Drones w/**Or**

(b) Four Drones w/**Or**



(c) Two Drones w/**OrDyn**



(d) Four Drones w/**OrDyn**

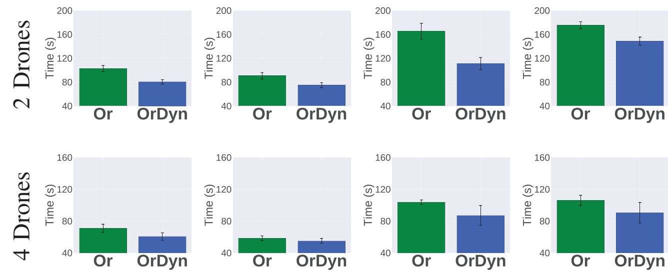
Fig. 7: Final trajectories of **Or** and **OrDyn** for two and four drones with a **Single Corner** configuration in the *Rooms2* map. Notice how the red drone in the case of two drones using **OrDyn**, and the red and purple drones in the case of four drones using **OrDyn** take advantage of velocity inertia and exhibit further and.

Final trajectories in *Rooms2* can be seen in Figure 7. The trajectories generated by **Or** exhibit a high degree of directional variability, whereas those produced by **OrDyn** show a notable reduction. This is observed in Figure 7c for the red drone and in Figure 7d for the red and purple drones.

D. Exp2: Alternative Starting Configurations

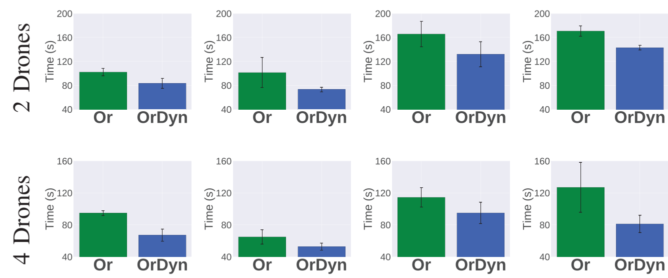
In this experiment, alternative initial configurations are tested to analyse the impact on performance. Firstly, in **Center**, all drones start in the center of the map. In **Corners**, each drone starts in a different corner of the map. In executions with two drones, the drones start from opposite corners.

Exploration times regarding the **Center** configuration are presented in Figure 8. The results align with the previous experiments as our method, **OrDyn**, continues to outperform **Or**, indicating its robustness across different initial conditions



(a) Office (b) Pillar (c) Rooms1 (d) Rooms2

Fig. 8: Exploration times comparing **Or** and **OrDyn** for two (top) and four (down) drones in four environments with *Center* configuration. The plots represent the mean and standard deviation for five executions. The same conclusions as in the **Single Corner** configuration hold.



(a) Office (b) Pillar (c) Rooms1 (d) Rooms2

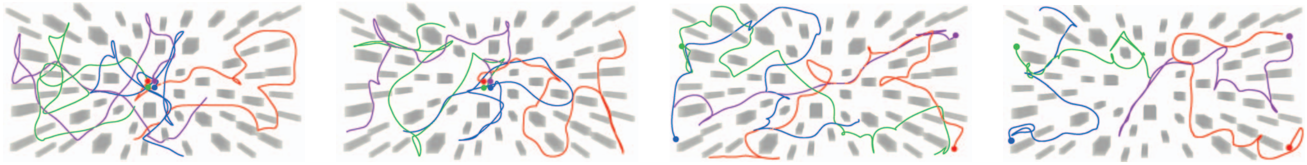
Fig. 9: Exploration times comparing **Or** and **OrDyn** for two (top) and four (down) drones in four environments with *Center* configuration. The plots represent the mean and standard deviation for five executions.

where the drones can collaborate closely. As an alternative setup, where drones are not initially close in space, we test the **Corners** case. Results, presented in Figure 9, show that although drones are initiating the exploration independently from separate positions, **OrDyn** still outperforms **Or**.

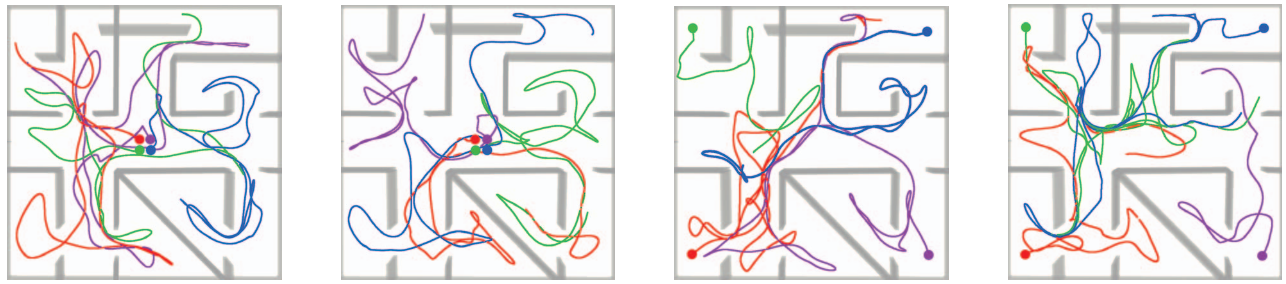
Finally, Figure 10 illustrates the trajectories for teams of four drones in the *Rooms1* and *Pillar* maps, both in the **Center** and **Corners** configurations, respectively. Regarding the **Center** configuration, in *Rooms1*, despite the map layout being more complex, requiring abrupt turns and manoeuvres, our method, **OrDyn**, achieves a faster exploration than **Or** as a result of the enhancement in collaboration between drones. In the case of **Corners**, when drones start from separate positions, collaboration is constrained. Nevertheless, **OrDyn** continues to outperform **Or** in all settings.

V. Conclusion

In order to improve efficiency in the exploration of unknown environments with drones, this paper has presented a method that allows to incorporate knowledge about the dynamics of a team in the planning process. This is done by proposing a novel dynamics cost into the VRP to incorporate information of their dynamics into the planning process. The method was integrated by extending and modifying components from a



(a) Pillar w/**Or** (b) Pillar w/**OrDyn** (c) Pillar w/**Or** (d) Pillar w/**OrDyn**



(e) Rooms1 w/**Or** (f) Rooms1 w/**OrDyn** (g) Rooms1 w/**Or** (h) Rooms1 w/**OrDyn**

Fig. 10: Final trajectories of **Or** and **OrDyn** for four drones with the **Center** and **Corners** configuration in the *Pillar* and *Rooms1* maps. In the *Rooms1* environment, we observe that the layout requires abrupt turns continuously in order to be fully explored. But **OrDyn** still outperforms **Or**.

state-of-the-art exploration system, RACER. Extensive experiments have been performed in simulated environments to validate the proposed method in different scenes, number of drones in the team and different initial configurations. The experimental results show improvements in exploration efficiency. Furthermore, the results suggest that the improvements clearly appear when the agents cooperate rather than improving single agent performance, emphasising the significance of agent collaboration and the impact of initial configurations. From the positive results, future work will further study variants of the dynamics cost. Additionally, exploring other features apart from the dynamics such as different capabilities in terms of traversability safety (e.g., aerial and ground robots), or sensor configurations in a similar setup is promising to enhance agent collaboration and improve the overall mission performance.

Acknowledgments

This work was supported by DGA project T45_23R, by MCIN/AEI/ERDF/European Union NextGenerationEU/PRTR project PID2021-125514NB-I00; ONR grant N62909-24-1-2081 and grant FPU20-06563.

References

[1] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.

[2] Y. Du, "Multi-uav search and rescue with enhanced a* algorithm path planning in 3d environment," *International Journal of Aerospace Engineering*, vol. 2023, no. 1, 2023.

[3] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," 2020.

[4] D. Morilla-Cabello, L. Bartolomei, L. Teixeira, E. Montijano, and M. Chli, "Sweep-your-map: Efficient coverage planning for aerial teams in large-scale environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 810–10 817, 2022.

[5] L. Bartolomei, L. Teixeira, and M. Chli, "Fast multi-uav decentralized exploration of forests," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5576–5583, 2023.

[6] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 2135–2142.

[7] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, 2023.

[8] H. Zhang, S. Wang, Y. Liu, P. Ji, R. Yu, and T. Chao, "Efp: Efficient frontier-based autonomous uav exploration strategy for unknown environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2941–2948, 2024.

[9] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 1997, pp. 146–151.

[10] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," *The annual research report*, 1998.

[11] Y. Kompis, L. Bartolomei, R. Mascaro, L. Teixeira, and M. Chli, "Informed sampling exploration path planner for 3d reconstruction of large scenes," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7893–7900, 2021.

[12] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 1500–1507, Apr. 2020.

[13] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[14] Y. Yang, J. Zhang, W. Qian, H. Geng, and Y. Xie, "Autonomous exploration for mobile robot in three dimensional multi-layer space," *Int. Conf. on Intelligent Robotics and Applications Smart Robotics for Sustainable Society*, 07 2023.

[15] D. S. Sundarsingh, J. Bhagiya, J. Chatrola, and P. Jagtap, "Autonomous exploration using ground robots with safety guarantees," in *2023 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[16] Y. Tao, Y. Wu, B. Li, F. Cladera, A. Zhou, D. Thakur, and V. Kumar, "Seer: Safe efficient exploration for aerial robots using learning to predict information gain," in *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2023.

[17] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/

[18] V. Furnon and L. Perron, "Or-tools routing library," Google. [Online]. Available: https://developers.google.com/optimization/routing/

[19] K. Helsgaun, "An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems," *Roskilde: Roskilde University*, 2017.