Cooperative Decision-Making for Drone Swarms

Luiz Giacomossi Jr.[1], **Marcos R. O. A. Maximo**[1], **José F. B. Brancalion**[2]

[1]Autonomous Computational Systems Lab – Aeronautics Institute of Technology (ITA)
São José dos Campos – SP – Brazil

[2]Technological Development Department – EMBRAER S.A.
São José dos Campos – SP – Brazil

luiz.giacomossi@ga.ita.br, mmaximo@ita.br, jose.brancalion@embraer.com.br

Abstract. *Unmanned Aerial Vehicles (UAVs) are being researched for their potential in applications like search and rescue, and defense missions. The goal is to enhance the intelligence, communication, and strategic organization. Decision-making techniques enable intelligent UAV decisions, freeing human commanders to focus on higher-level decisions. This research focuses on defense and search and rescue scenarios, and combines AI-based decision-making with UAVs. The study analyze the Loyal Wingman concept in a defense scenario. Also, we propose a solution for a drone swarm to cooperatively search for people in a rescue scenario. Our results demonstrate the effectiveness of distributed decision-making methods in solving problems in both scenarios.*

Student level: *M.Sc. Date of conclusion (defense): 07/28/2023. To be considered in CTDR.*

1. Introduction

The study of decision-making in drone swarms is crucial for optimizing performance, autonomy, and safety in various industries. Understanding swarm intelligence, human-swarm interaction, and swarm behavior in complex scenarios can drive advanced swarm technologies, transforming industries and tackling challenges. This research is essential for civil and military applications, such as search and rescue missions and the deployment of loyal wingman unmanned aerial vehicle (UAV), an emerging combat drone in the aeronautical industry.

This study investigates the utilization of combat-capable Loyal Wingman (LW) drones in cooperative efforts to engage and disable aerial explosive threats. The focus is on a Manned-Unmanned Team (MUM-T) defense scenario, where LW UAVs protect a leader UAV and critical infrastructure. The research considers fully actuated UAV models [Santos and Bezerra 2022] and proposes a problem breakdown for high-level decision-making tasks, enabling effective coordination and collaboration within the MUM-T. The project employs finite-state machines (FSMs) and behavior trees (BTs) as AI techniques to design autonomous UAV behavior. The Kuhn-Munkres task allocation algorithm is utilized for task distribution, promoting cohesive teamwork. The study also extends the Cooperative Engagement Capability (CEC) to include drones and investigates distributed autonomous decision-making in a heterogeneous drone swarm within the CEC concept, aiming to enhance cooperative engagement strategies. Integrating autonomous drone systems with CEC is expected to enhance mission efficiency and success.

Moreover, this work also utilizes drone swarms for search and rescue (SAR) applications. The integration of drone technology in SAR missions represents a groundbreaking advancement in emergency response capabilities, as it harnesses the collective power and efficiency of UAV systems. By employing drone swarms, SAR operations can revolutionize the way we respond to emergencies and save lives. By integrating advanced swarm intelligence algorithms to drones, this research aims to optimize SAR missions, and ultimately save lives. In this work we describe the scenario where a swarm of drones searches for a person in distress in a forest. To tackle this problem, we utilize an approach combining robotics and AI-based decision-making. This work aims for a future where drone swarms act as indispensable allies in crisis management, rapidly locating and aiding individuals in distress, and providing invaluable support to first responders. Note that the two scenarios in this work are integral components of a unified study that explores the application of decentralized decision-making within drone swarms in simulated environmens.

This paper is organized as follows. Section 2 presents the backgroung. Section 3 presents the loyal wingman application, its experiments and results. Also, Section 4 presents the search and rescue application, its experiments and results. Finally, Section 5 concludes the paper.

2. Background

Behavior in robotics refers to the actions or responses of a robot in a given situation. Behaviors can be pre-programmed or learned through machine learning [Colledanchise and Ogren 2018, Mahadevan and Connell 1992]. Robots exhibit a range of behaviors, from simple actions to complex tasks based on sensory input [Choset et al. 2005]. Behavior-based control architectures are commonly used, employing reactive behaviors that interact to produce the desired overall behavior [Arkin et al. 1998, Colledanchise and Ogren 2018]. UAVs, or drones, utilize behavior-based control architectures and sensors such as GPS and accelerometers to perform tasks like surveillance and inspection [Garcia-Aunon et al. 2019, Ricardo Jr and dos Santos 2023]. They adapt their behavior based on sensor input to adjust flight paths and avoid obstacles [Ricardo Jr and dos Santos 2023]. This adaptability enables UAVs to perform various tasks in different environments, providing flexibility and surpassing human capabilities [Ricardo Jr and dos Santos 2023].

2.1. Finite State Machines

FSMs are the most common mathematical model of computation where the system can be in only one of a finite number of states at any given time [Buckland 2004]. *i.e* an FSM guarantees the permanence in a certain state, unless a transition is triggered. The developer is responsible for defining the behaviors (states) and the conditions that trigger transitions between behaviors. The wide use of FSMs is due to their intuitive structure and ease of implementation. However, FSMs have scalability disadvantages with the addition of behaviors and transitions, so code maintenance is laborious [Colledanchise and Ogren 2018]. Reusability is also an issue, making it unpractical for reusing behaviors in other projects.

2.2. Behavior Trees

The behavior tree (BT) approach is used to encode behaviors that are modular and reactive [Colledanchise and Ogren 2018]. Since most of the problems found in the FSM are easily handled by BTs, the method has surpassed the FSMs as the industry standard in AI games [Ogren 2012]. A BT framework [Colledanchise and Ogren 2018] is composed by nodes, which can be composite or leaf. Composite nodes control the BT logic, while leaf nodes execute the behaviors or check conditions. When executed, each node returns a execution status: `Success`, `Failure`, or `Running`.

The types of nodes are dictated by the framework. The frequent types of composite nodes are: sequence, selector, parallel, and decorator. After its execution, each node return a status of its execution, Table 1 shows the return status logic of each node type. Sequence nodes sequentially

Tabela 1. Node types of a BT.

Node type	Success	Failure	Running
Selector	If one child succeeds	If all children fail	If one child returns running
Sequence	If all children succeeds	If one child fails	If one child returns running
Parallel	If N children succeeds	If M-N children fail	If all children return running
Action	Upon completion	When impossible to complete	During completion
Condition	If true	If false	Never
Inverter	If `Failure`	If `Success`	-

executes all their children in order, as long as they are successful. A selector is used when any child can perform the task, it selects the first child that is successful. A parallel node executes its children in parallel (at the same run time). A decorator (`Inverter`) node changes the execution status of its child. Many types of decorator may exist, depending on the framework used. Conditional check nodes are used to check if a condition is satisfied. The leaf nodes are implemented by the agent developer: they are the behaviors themselves or conditional checks, *e.g.* a behavior such as `DefendLeader` will keep returning `Running` while the behavior is executing. Then, if the threat attacking was neutralized, the node return `Success`. Otherwise, if the leader is destroyed, the node returns `Failure`.

2.3. Bayesian Search for Search and Rescue

In the Bayesian search theory approach for search and rescue missions, initial probabilities are assigned to cells based on expert knowledge [Stone 1976]. The search process involves dividing the search area into a grid map, with each cell having low, medium, or high probabilities of the target's presence. The search starts with the cell having the highest probability, and if the target is found, the mission is successful. If not, the probabilities are updated iteratively using the Bayes formula [Box and Tiao 2011] to reflect new information from the search.

The Bayes formula guides the iterative updates of probabilities, incorporating observed data. To update the cell probabilities, variables p (probability of the target being in a cell) and f (probability of finding the target if it is present) are introduced. The posterior probabilities of searched and other cells are updated accordingly, leading to a dynamic adjustment of probabilities and an improved estimation of the target's location. This iterative process continues until the target is located, demonstrating the effectiveness of the Bayesian search method for SAR missions.

3. Loyal Wingman Application

This section describes the application of LW systems using UAVs and decision-making capabilities. The section assesses the effectiveness of LW systems using metrics, and discusses the development of decision-making for Kamikaze and Loyal Wingman UAVs. Experiments are presented to validate the performance and efficiency of the implemented decision-making.

3.1. Loyal Wingman Scenario

To explore the concept of Cooperative Engagement Capability (CEC) within the Manned-Unmanned Team (MUM-T), we present a defense scenario [Giacomossi et al. 2021a, Ricardo et al. 2023] involving two teams comprising fully actuated UAVs. The MUM-T team consists of Loyal Wingman (LW) UAVs that fly in formation alongside a manned leader, providing defense for both the leader and a protected area (PA). In contrast, the adversarial team comprises a swarm of kamikaze threats, as in Figure 1. The primary objective of the MUM-T

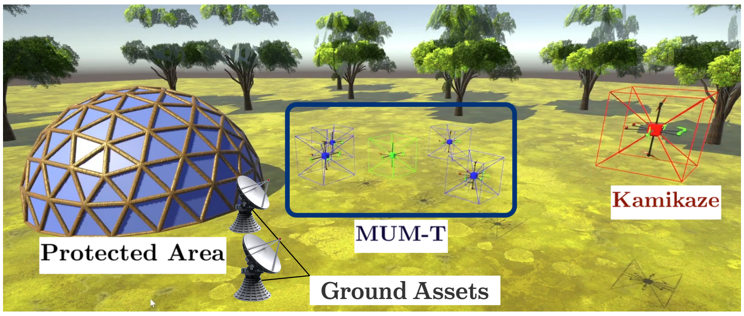


Figura 1. Scenario of interest where LW UAVs, highlighted in blue, supported by ground assets, escort a leader UAV and engage kamikaze threats in order to defend a protected area.

team is to prevent any damage to the leader or the PA. The mission is considered unsuccessful if the leader or the PA are destroyed. Furthermore, to support the MUM-T members, the PA is equipped with ground assets capable of providing aerial surveillance.

The LW is autonomous and capable of making intelligent decisions based on the situational awareness information. To neutralize the threats, the LW is equipped with two hypothetical weapons, a mid-range *freezing gun* and a short-range *vaporizer gun*, both with a limited number of cartridges and a fixed cool-down time interval. The *vaporizer gun* can neutralize the threat, while the *freezing gun* slows down the threat by half of its maximum speed. The weapons' model is simplified, being the hitting success calculated by a given probability. Note that, the *freezing gun* is intended to make the decision space more complex, and the *vaporizer gun* is also an idealized weapon that uses energy to destroy the electronic components.

The leader is remotely controlled by a human and it is in charge of the formation coordination, *i.e.*, it is capable of passing relative coordinate commands to the LW. We assume the loyal wingmen to fly within one predefined formation pattern. In this paper, we consider this unique pattern as an uniform-circular formation along the local horizontal plane with a desired radius. We also assume that the leader is always capable to command the loyal wingmen whenever required.

In contrast, kamikaze UAVs detonate upon collision with the leader, an LW, or the PA, causing damage to the target. Once a kamikaze selects a target, it pursues it until self-explosion or neutralization. The number of kamikazes remains constant, as they immediately re-spawn after being neutralized, ensuring a continuous stream of attacks. Though possessing simplified AI, the kamikazes are faster and more numerous than the MUM-T, compelling the MUM-T to collaborate effectively for neutralization. Situational awareness is shared by the ground assets, providing a vector with the state (position, attitude, and linear velocity) of all entities.

3.2. Problem Breakdown

Robotic cognition involves decomposing agents into software layers for situational awareness, decision-making, and control, as shown in Figure 2. This work focuses on high-level decision-making, cooperation, and low-level path planning, guidance, and control tasks, idealizing situational awareness.

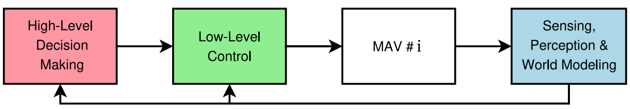


Figura 2. Problem breakdown indicating the main software layers.

The HL decision-making layer processes state information from situational awareness, containing pose and linear velocity of objects in a global coordinate system. AI-based algorithms process the states, generating action commands for rotational and translational movement, and weapon use. The LL control translates movement actions into state references, which are used to calculate control input for the MAV and allocate weapon actions. The MAV module includes vehicle dynamics, control allocator, actuator model, and weapons model, with simplified aiming and projectile dynamics. MAVs share a synchronized internal model, aware of threats and other agents due to ground-asset radars. Each agent is identified by a unique ID. Note that this work addresses only the high-level decision making aspects of the problem breakdown.

3.3. Problem Definition

In this section, we describe the problem to be addressed by LW agents' decision-making. Their main mission is to defend the protected area and the leading drone against multiple kamikaze drone incursions. Effectively engaging and neutralizing threats is crucial for ensuring both safety and mission success. To achieve this, LW UAVs must possess autonomous decision-making capabilities, including flying in formation with cohesion and effectively engaging and neutralizing threats. This requires the use of embedded weaponry, intelligent and strategic weapon deployment, and, if necessary, sacrificing themselves to neutralize a threat.

The main problem of the HL decision-making can now be defined.

Problem I. The main problem is to develop an autonomous intelligence module for the LW MAV to successfully achieve the mission objective, aiming at the smallest loss of LW agents during the attack and defense maneuvers.

Based on this problem it is identified a need for defining a set of expected behaviors for the agents. Therefore, we define the following subproblem:

Subproblem I.1. Design basic behaviors for the LW agents and to develop a decision-making architecture to coordinate these behaviors.

To tackle Subproblem I.1 we need to design the adversary agents, to properly evaluate our solution to Problem I.

Subproblem I.2. Design basic behaviors for the kamikaze agents and to develop a decision-making architecture composed of these behaviors.

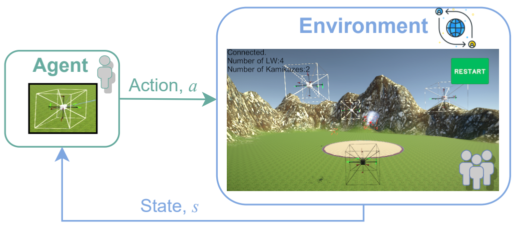


Figura 3. Problem I definition diagram.

Therefore, the main agent in our problem is the LW UAV, who will have to perform actions based on the AI-based decision making, as seen in Figure 3. The action a is performed by the agent based on the agent's state s as feedback. Note that the environment encompasses all agents.

3.4. LW-Kamikaze Assignment Problem

One of the subproblems also addressed is the allocation (or assignment) of tasks among the LW UAVs. By task we mean the neutralization of a specific kamikaze threat. The classical assignment problem, extensively studied [Burkard and Çela 1999, Munkres 1957], deals with optimally assigning workers to tasks based on ratings or costs. The aim here is to facilitate cooperation among MUM-T members, where the leader UAV commands the LWs and distributes tasks. Thus, let us consider the following subproblem.

Problem II. Given a set of workers $\mathcal{W}$, a set of tasks $\mathcal{J}$, and a set of costs $\mathcal{C}$ indicating how effectively each worker $w_i \in \mathcal{W}$, where $i \in \{1, \dots, n\}$, can perform each task $j_i \in \mathcal{J}$ determine the best possible assignment of workers to tasks, such that each task is assigned to one worker and each worker is assigned one task, so the total cost is minimized.

One wishes to choose a set of n independent elements $(c_{i,j})$ of a cost matrix $\mathbf{C}$, where $c_{i,j}$ is the element of the i-th row and j-th column of $\mathbf{C}$, so that the sum of these elements is minimized. This can be expressed as permuting the rows and columns of $\mathbf{C}$ to minimize the trace of a matrix $\min_{\mathbf{L},\mathbf{R}} \mathrm{Tr}(\mathbf{LCR})$, where $\mathbf{L}$ and $\mathbf{R}$ are permutation matrices, and the cost matrix is

$$\mathbf{C} = \begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{array} \begin{array}{c} \begin{array}{cccc} j_1 & j_2 & \cdots & j_n \end{array} \\ \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix} \end{array}. \tag{1}$$

The Kuhn-Munkres algorithm, or the Hungarian algorithm, is a renowned solution to the assignment problem [Munkres 1957], refined by J. Munkres. It operates on a cost matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$, where each element $c_{i,j}$ represents the cost of assigning the jth task to the ith worker.

3.5. Decision-Making Development

To start addressing Problem I, we need the kamikaze agents to be functional via simplified but effective AI. Consequently, we selected the FSM technique to develop the decision-making module for the kamikaze. In this technique, a state s represents a behavior for the agent. In Fig. 4, we present the decision-making developed for the kamikaze UAV, note that each UAV contains an identical decision module.

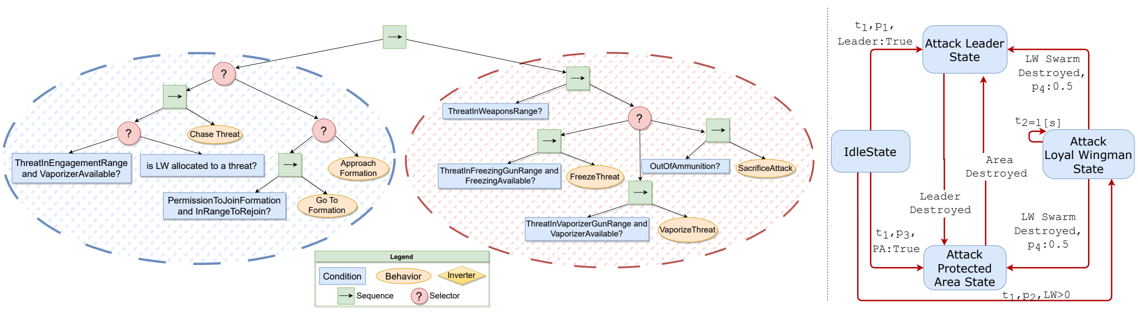


Figura 4. On the left, the Behavior tree for the loyal-wingman UAV. The blue-circled behaviors represent movement behaviors, and the red-circled behaviors depicting offensive behaviors. On the right, the Decision-Making module for a kamikaze MAV using FSM.

The kamikaze has a set $\mathcal{K}$ of four behaviors, the initial behavior is *IdleState*, where the agent is idle for t_1 seconds and then selects a target based on the probabilities p_1, p_2, and p_3, which in this work are equally distributed probabilities used for each target type. Once a target is selected, the FSM will transition to the representative state of the selected target and will remain in that state until the agent is destroyed. The exception is *Attack Loyal Wingman*; this behavior targets the closest LW, and will select the closest LW every $t_2 = 1$ s, to avoid frequent target switching. Once there are no more LW, the FSM transitions to attack the leader or the protected area with a probability of 50% each.

To address the complexities of LW UAV behaviors and their desired capabilities, a more elaborate AI architecture is needed. For Subproblem I.1, the decision-making module employs the BT technique, chosen for its advantages over FSMs [Iovino et al. 2022]. The BT diagram for an LW UAV, shown in Figure 4, includes default behavior of staying in formation (`Go To Formation`) unless threats are detected. When a threat enters the engagement range or the LW is assigned a task, it switches to the `Chase Threat` behavior, utilizing available weapons such as the vaporizer gun or freezing gun. The LW selects a neutralization method based on weapon availability and ranges. The preferred strategy involves freezing the closest threat within mid-range, approaching it safely, and using the vaporizer gun. If both weapons run out of ammunition, the LW resorts to the `SacrificeAttack` behavior. After eliminating the threat, the LW returns to the formation by approaching it from a safe distance $d_f \in \mathbb{R}_{>0}$. It then requests permission to rejoin and awaits the formation coordinates.

3.6. Simulation Results

The effectiveness of the overall method is evaluated using a Monte Carlo simulation with 122 iterations. These simulations are performed considering 4 LW against 3 kamikazes (that respawn once neutralized) with the PA located at the origin and the leader hovering at the point $(5, 0, 5)$ m. The LW have a maximum speed of 1.5 m/s, being the kamikazes 50% faster. The freezing and vaporizer guns have an ammo of 10, a cooling down of 1 s, and a hit probability of 95%. The freezing gun can reduce the kamikaze speed by half during 5 s. Figure 5 shows the simulation results. It can be seen that, on average, the survival time is 169.3 s and the number of kamikazes destroyed is 34, without considering the sacrifice attack behavior, which is an excellent result since the MUM-T can only directly neutralize a maximum of 40 kamikazes given that each LW is equipped with a vaporizer gun with an ammo of 10. This corresponds to 85% of the total neutralizing capability.

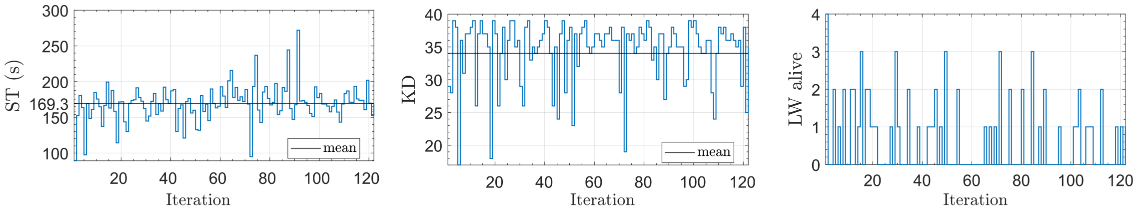


Figura 5. Monte Carlo simulation results. ST stands for survival time and KD stands for kamikazes destroyed.

4. Search and Rescue Application

This study examines UAVs' potential in search and rescue (SAR) missions, focusing on AI techniques for addressing challenges. It introduces a tailored strategy and decision-making process for drones and swarms, emphasizing autonomous and collective capabilities. The experimentation includes two approaches: developing autonomous and collective decision-making and a search strategy.

4.1. Search and Rescue Scenario

Developing a search and rescue scenario using drones provides an opportunity to study the intricacies of the complex problem of locating missing individuals. By simulating scenarios and evaluating the performance of drone swarms, insights into the effectiveness of different search strategies

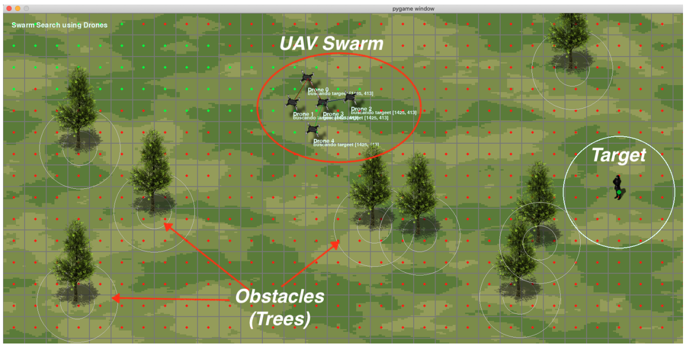


Figura 6. Search and Rescue scenario, a swarm of drones, within the red circle. The environment symbolizes obstacles as trees. The objective is to locate the lost person.

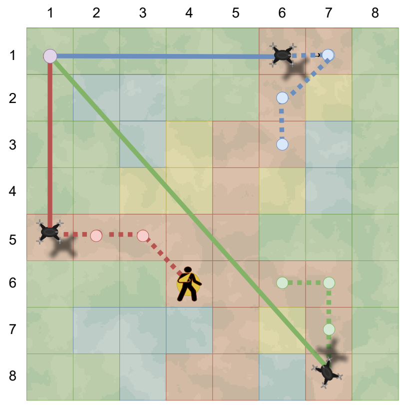


Figura 7. Search process using three drones. Green, yellow, and red cells represent low, medium, and high probability regions, respectively. No flight zones are illustrated as blue cells. The yellow circle with a person represents the object of interest. Each line with a different color represents the path taken by a different drone. Gray circle represents the start point.

and algorithms can be gained. This involves designing environments, incorporating factors like obstacles, and limited visibility to mimic real-world search scenarios. For this, we utilize a 2D swarm simulator [Giacomossi et al. 2021b] with the proposed scenario involving a UAV swarm in a forest region of approximately $60,800\ m^2$ and dividing the area into $(38 \times 16\ m)$ squares. The swarm assumes that the UAVs lack prior information about the missing target's position and must navigate around obstacles. The drones operate within a shared UAV swarm network, allowing the exchange of mission-related data, including the positions of all mission components and the location of the identified target. The target is considered located when a drone is within a distance of less than two squares, as depicted in Figure 6 by a circle around the target.

4.2. Search Process using a Probability Map

At the start of the search process, the drones communicate to select different cells on the map with high probabilities and navigate towards it to begin the search process, as in Figure 7. Once the drone reaches the intended location, it actively initiates the search process using Hill Climbing (HC) search and Bayesian Search (BS) algorithms [Giacomossi et al. 2023]. In Figure 8, we can observe the search process being performed by the drone. The drone follows a strategy where it visits a cell and, in case the object is not identified, it examines the probabilities of the 8 neighboring cells. Then, using the HC algorithm, the drone moves to the cell with the highest probability

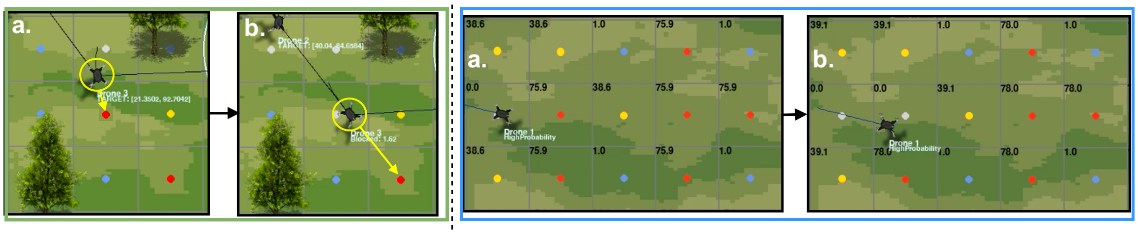


Figura 8. The search selection is shown within the green square, where the drone covers 8-connected cells. High, medium, and low probabilities are represented by Red, Yellow, and Blue cells, respectively. The Gray cells indicate areas already visited by the drone. Inside the blue square, the sequence depicts the drone employing the Bayes algorithm to update the probabilities associated with each cell as it explores them. The numbers in each cell represent these probabilities, which are continually updated.

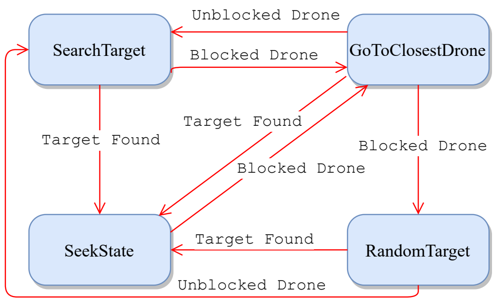
to continue the search.

In the blue sequence in Figure 8 we illustrate the Bayes algorithm updating the cell probabilities in a simulation. The probabilities of all map cells are updated at each step using the solution described in [Giacomossi et al. 2023]. As seen in frames (a) and (b), the probabilities increase as the drone visits new cells without finding the object. In each instance of the simulation, a new probability map is created randomly. The probability for each cell in the map is assigned using an uniform distribution. The location of the individual is also updated to a new high probability cell in the newly generated map. Also, new obstacles are randomly positioned in the map, which helps to evaluate the algorithm's performance under different conditions. The random assignment of probabilities to each cell ensures that the simulation results are unbiased towards any particular scenario.

4.3. Decision-Making for the Search and Rescue Drone Swarm

Our decision-making can be divided into two categories: autonomous intelligence, referring to a single UAV, and collective swarm intelligence, referring to the group. Therefore, decisions are decentralized to each UAV, and cooperative, as the group must cooperate to attain the mission. The implemented intelligence modules are described in Figure 9. The target search module distributes UAVs among scenario according to the strategy developed. Each drone autonomously starts searching the scenario. To control drone behavior, a FSM was developed, representing different behaviors. The FSM for the search and rescue UAV is shown in Figure 9, comprising two main states: `SearchState` and `SeekState`. Two auxiliary states, `GoToClosestDrone` and `RandomTarget`, help prevent blockages caused by local minima. Note that each drone has its own FSM.

The drone's default behavior is to search for the missing target in the `SearchTarget` state. When the target is located, either by the drone itself or another agent, it transitions to the `SeekState`. In this state, the target's coordinates are shared with all agents. If the drone encounters an obstacle hindering its movement, it switches to the `GoToClosestDrone` behavior, navigating towards the nearest agent as a heuristic. If the drone remains blocked, it enters the `RandomTarget` state, selecting a random target to navigate towards. Once unblocked, the FSM returns to the default `SearchTarget` state. Notably, the drone can transition to the `SeekState` from any state in the FSM.



Swarm Intelligence	Module	Description
Autonomous Intelligence	Target Search	Target search is performed on a discrete map.
	Behavior	Intelligent drone behavior through a finite state machine.
	Collision Avoidance	Potential fields for collision avoidance between drones and obstacles.
Collective Intelligence	Swarm Search	Target search in regions mapped as not visited using a common map.
	Blockage Prevention	Prevents drones from getting stuck in local minima present in the scenario.
	Mission Closure	When a drone finds the target the coordinates are shared and all drones move to this position.

Figura 9. Finite State Machine that controls the drone's behavior. On the right, the proposed heuristics description for Swarm Intelligence.

Tabela 2. Results of the experiments.

Exp.	Strategy	Mean Time [s]
1	Lawn Mower - [Giacomossi et al. 2021b]	57.6
2	Probability Map + HC + BS	33.4

4.4. Experiments

The simulator [Giacomossi et al. 2021b] has been enhanced to generate a new probability map during each new instance, which includes updated obstacles and a new location for the missing person. This dynamic probability map enables the drones to adapt their search patterns based on the latest information, enhancing their efficiency. For the experiments, we evaluate the average time taken to complete the mission in 270 runs and compare it to the results obtained in our previous section, which employed a strategy based on the lawn mower search [Giacomossi et al. 2023] and no additional information.

4.4.1. Results

In Table 1 we present the results obtained from the experiments. Notably, Experiment 2 demonstrated a shorter mean time of 24.2 seconds compared to Experiment 1. This indicates that the new strategy utilizing the probability map with the addition of the HC and BS algorithms resulted in an approximately 72% improvement in time efficiency for identifying the missing person. These results suggest that the HC and BS algorithms significantly reduced the time required to complete the task, in comparison to the approach adopted in Experiment 1.

Also, consider that the drones are being simulated in a 2D environment with reduced degrees of freedom, simplified dynamics, and with idealized identification and communication capabilities. So notice that these results can deviate from real world performance. Furthermore, it demonstrates the importance of improving the search method when the goal is to optimize the time efficiency. Note that the Experiment 2 may be more useful as a model for achieving that goal, but the Experiment 1 is still a safe approach when performing a search when there is no information about the region, i.e, no previous map of probabilities is provided.

The results of the comparison between the mission with and without the aid of a probability map provide evidence that using such a map can significantly reduce search time. An interesting observation from the experiments is that when the target is located in a high probability area with most of its cells in the direct path of the drones, as seen in Figure 8, the time taken to locate the target is significantly reduced. This is because the drones can fly almost directly to

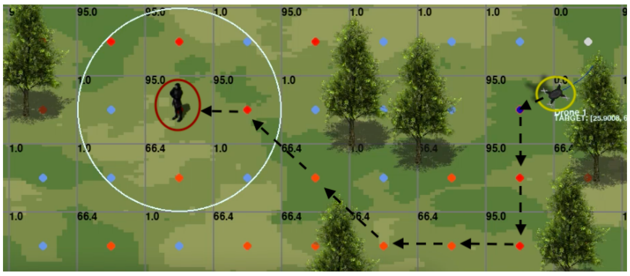


Figura 10. Drone performing the search in the 8-connected cells. Red, Yellow and Blue cells are the high, medium and low probabilities respectively. In Gray, cells already visited. The numbers represent the probability of each cell.

the target, bypassing areas with low probabilities of detection. However, if the target is not in a high probability area, the algorithm may lead the drones to wrong areas, resulting in decreased time efficiency. Therefore, while the use of a probability map can improve search time in optimal scenarios, its effectiveness may vary in situations where the map has incorrect assumptions. Thus, the accuracy of the probability map is a critical factor that has a significant impact on the search.

5. Conclusion

This paper delves into the utilization of decision-making techniques in the context of swarms of UAVs. The research undertakes an evaluation of two distinct applications: the so-called loyal wingman drones within a defense scenario, while the second concentrates on a civil application, targeting search and rescue missions aimed at locating missing individuals. For both applications, AI-based decision-making techniques were employed. Additionally, a tailored set of rules and strategies was developed to address the unique challenges presented by each specific problem.

This study evaluates the Cooperative Engagement Capability (CEC) concept in a Manned-Unmanned Team (MUM-T) operating in a gamified defense scenario. The team consists of a manned-controlled UAV and a group of loyal wingman UAVs with combat capabilities. The goal is to enhance the defense of the leader UAV and protect critical infrastructure from a swarm of Kamikaze UAVs equipped with explosives. The loyal wingmen use vaporizer and freezing guns to counter threats. AI-based decision-making and collaboration within the MUM-T are aimed at achieving higher efficiency in the mission. The study developed AI modules for both LW and kamikaze UAVs, focusing on FSMs and BTs. The Munkres algorithm was employed to evaluate cooperative capabilities within the MUM-T, and a uniform-circular formation was proposed.

This work also investigates the use of UAV swarms for search mission, focusing on locating unknown individuals. A UAV swarm simulator was developed to evaluate various aspects of the research. We developed intelligent search strategies based on autonomous and collective decision-making for UAVs, employing a Finite State Machine (FSM) to control drone behaviors, a discrete map of the search region, and a blockage prevention heuristic to reduce UAVs stuck in local minima. The swarm intelligence approach was found to be effective, reducing the number of UAVs in local minima and reducing the mission completion time. Future works can investigate various communication challenges such as delays and packet loss, state estimation, incorporate more realistic drone dynamics, and adopt a sensor model that better reflects real-world conditions.

Referências

Arkin, R. C., Arkin, R. C., et al. (1998). *Behavior-based robotics*. MIT press.

Box, G. E. and Tiao, G. C. (2011). *Bayesian inference in statistical analysis*. John Wiley & Sons.

Buckland, M. (2004). *Programming Game AI by Example*. Jones & Bartlett Publishers, Burlington, Massachusetts, USA.

Burkard, R. E. and Çela, E. (1999). *Linear Assignment Problems and Extensions*, pages 75–149. Springer US, Boston, MA.

Choset, H., Lynch, K., Hutchinson, S., Kantor, G., and Burgard, W. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Intelligent Robotics and Autonomous Agents series. MIT Press.

Colledanchise, M. and Ogren, P. (2018). *Behavior Trees in Robotics and AI: An Introduction*. Chapman & Hall/CRC Press.

Garcia-Aunon, P., del Cerro, J., and Barrientos, A. (2019). Behavior-based control for an aerial robotic swarm in surveillance missions. *Sensors*, 19(20).

Giacomossi, L., Maximo, M. R., Sundelius, N., Funk, P., Brancalion, J. F., and Sohlberg, R. (2023). Cooperative search and rescue with drone swarm. In *IAI2023 - International Congress and Workshop on Industrial AI*.

Giacomossi, L., Schwanz Dias, S., Brancalion, J., and Maximo, M. (2021a). Cooperative and decentralized decision-making for loyal wingman UAVs. In *IEEE Latin American Robotics Symposium (LARS)*, pages 78–83. IEEE.

Giacomossi, L., Souza, F., Cortes, R. G., Mirko M. Cortez, H., Ferreira, C., Marcondes, C. A. C., Loubach, D. S., Sbruzzi, E. F., Verri, F. A. N., Marques, J. C., Pereira, L. A., Maximo, M. R. O. A., and Curtis, V. V. (2021b). Autonomous and collective intelligence for uav swarm in target search scenario. In *2021 Latin American Robotics Symposium (LARS)*, pages 72–77. IEEE.

Iovino, M., Scukins, E., Styrud, J., Ögren, P., and Smith, C. (2022). A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems*, page 104096.

Mahadevan, S. and Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55(2):311–365.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Ogren, P. (2012). Increasing modularity of uav control systems using computer game behavior trees. *AIAA Guidance, Navigation, and Control Conference*, pages AIAA 2012–4458.

Ricardo, J. A., Giacomossi, L., Trentin, J. F. S., Brancalion, J. F. B., Maximo, M. R. O. A., and Santos, D. A. (2023). Cooperative threat engagement using drone swarms. *IEEE Access*, 11:9529–9546.

Ricardo Jr, J. A. and dos Santos, D. A. (2023). Robust collision-free guidance and control for fully actuated multirotor aerial vehicles. PREPRINT (Version 1) available at Research Square.

Santos, D. A. and Bezerra, J. A. (2022). On the control allocation of fully actuated multirotor aerial vehicles. *Aerospace Science and Technology*, 122:107424.

Stone, L. (1976). *Theory of Optimal Search*. ISSN. Elsevier Science.