

D³ARC: Time-Critical Distributed Disaster Detection for Asynchronous Cooperative Multi-Robot Systems

Nikolaos Koursioupas*, Lina Magoula*, Nancy Alonistioti*, Ramin Khalili**

* Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece

** Huawei Heisenberg Research Center (Munich), Germany

{nkoursioubas*, lina-magoula*, nancy*}@di.uoa.gr

{ramin.khalili**}@huawei.com

Abstract—Climate change is increasing the severity and unpredictability of natural disasters. In time-critical crises such as wildfires, traditional monitoring practices remain limited by coverage, cost, and personnel risk, paving the way for autonomous and adaptive monitoring solutions. Within this context, this paper introduces D³ARC, an asynchronous distributed hierarchical framework for time-aware and reliable wildfire detection. D³ARC integrates multiple robotic agents that cooperate under *uncertainty* through distributed perception, shared situational awareness and coordinated actions. A remote controller asynchronously decides upon each robot’s motion, while each robotic agent senses the environment and decides *where* and *how* to execute the wildfire detection. All robotic operations require time, and as time progresses, wildfires continue to spread, reducing the opportunity for early intervention. As such, all agents share a common objective: to detect a wildfire with a certain performance threshold as fast as possible and within a time limit. D³ARC integrates mechanisms for safe navigation, coverage efficiency, cooperation and reliability. It introduces a *forward-looking* capability that allows agents to anticipate the future by evaluating candidate strategies before execution. The framework is evaluated through realistic robotics simulations, ablation studies, and baseline comparisons, achieving an overall mission success up to 94% with 89.4% detection confidence.

Index Terms—Distributed Agentic Intelligence, Cooperative Robotics, Computer Vision, Time Efficiency, Wildfire Detection

I. INTRODUCTION

Climate change is altering the frequency and intensity of environmental hazards, such as droughts, heatwaves, floods, and wildfires [1]. The increasing severity and unpredictability of these events pose significant risks to human life and ecosystems. Wildfires, in particular, are challenging as they can spread rapidly and uncontrollably across large, often inaccessible regions, making timely detection essential. Accordingly, the European Union Agency for the Space Programme (EUSPA) states that wildfires should be detected within 10 minutes of ignition [2].

Traditional wildfire monitoring practices primarily include ground-based inspections, fixed observation stations, and aerial patrols. These practices are often constrained by limited spatial coverage, high operational costs, and most importantly, risks to personnel operating in hazardous environments. This highlights the need for innovative technological solutions that,

during such critical events, can effectively and safely monitor, perceive, act and adapt.

In this direction, robotics is emerging as a promising technology for addressing such environmental crises [3]. To operate effectively, robots must act as intelligent agents, and as such be equipped with self-learning capabilities and context-aware autonomy [4]. However, considering the spatial scale, dynamicity and overall *uncertainty* of such environments, the deployment of individual robotic agents may be insufficient. Instead, cooperative multi-robot agents can support distributed perception, shared situational awareness, and coordinated actions under rapidly evolving and time-critical conditions. These capabilities form cooperative agentic robotic intelligence as a new research direction, enabling faster response and greater operational effectiveness.

In real-world deployments, agentic capabilities are tightly coupled with the availability of computational, and communication resources. Particularly, each agent spends time for every sensing, computation, communication, and movement operation. Limited onboard computing can delay perception, and reasoning, while fluctuating network conditions introduce delays in coordination with other agents. These accumulated delays across perception, decision-making and action can directly affect a robot’s responsiveness, autonomy, and mobility range. Time awareness is therefore essential as delays may reduce the opportunity for early detection and allow a wildfire to spread further.

Considering these needs and challenges, this paper introduces D³ARC, a hierarchical distributed agentic framework for cooperative robots, consisting of multiple robots and a remote controller (RC), designed for time-aware and reliable wildfire detection under *uncertainty*. More specifically, the RC asynchronously controls the motion and pose of each robot, while each robotic agent decides upon *where* and *how* wildfire detection should be executed. A pool of AI wildfire detection models, differing in complexity and performance, is available to the robotic agents. The *where* decision selects between onboard and remote execution based on resource availability and network conditions, while the *how* decision selects the most suitable AI model from the available pool. When a

robotic agent selects remote execution, it transmits its sensing data to the RC, which *assists* the mission by performing wildfire detection remotely.

D³ARC follows a two phase methodology that combines offline knowledge acquisition with online real-time decision-making. In the online real-time phase, D³ARC integrates four mechanisms that improve safety, coverage efficiency, cooperation and mission reliability. A 360°-obstacle avoidance mechanism prevents unsafe robot motion near obstacles. A coverage efficiency mechanism rates regions based on number of visits and detection performance. This reduces redundant exploration and redirects the robots toward promising or unexplored regions. A call-for-aid mechanism promotes sensing and detection cooperation when at least one robot reports a promising region. An adaptive early stopping mechanism terminates the mission once reliable wildfire detection is sustained, avoiding unnecessary motion, time cost and overall mission duration. Finally, custom penalty functions promote feasible strategies that respect system objective and constraints.

Another key innovation of D³ARC is its *forward-looking* decision-making capability. The RC and robotic agents asynchronously act in a distributed manner but share a common mission reward only after their actions are completed. As such, each agent must *look forward* by estimating the shared mission outcome of its candidate decisions, while the RC must also account for the robots' subsequent actions. D³ARC enables this through distributed neural regression models to evaluate multiple candidate strategies before action execution. The agents then greedily select the strategy expected to provide the best trade-off between detection confidence and mission duration. The key contributions can be summarized as follows:

- An asynchronous distributed hierarchical cooperative decision-making framework between multiple agents that jointly optimize mission duration and effectiveness.
- The design and integration of mechanisms promoting navigation safety, coverage efficiency, cooperation and mission reliability in time-critical missions.
- A *forward-looking* decision-making that evaluates possible futures by anticipating their consequences prior to action selection.
- Realistic modeling of time for any robotic operation.
- A real-time evaluation process using state-of-the-art robotics simulations.
- Integration of heterogeneous robot perception capabilities, where external factors limit the detection performance of some robots.
- Cooperation effectiveness and robust performance, in terms of detection confidence, mission duration, and overall mission success, even under unseen scenarios and setups.
- Comprehensive ablation studies and baseline comparisons that demonstrate the effectiveness and advantages of our proposed approach.

The rest of the paper is organized as follows. Section II presents relevant state-of-the-art literature. Section III provides

the system model and section IV provides the problem formulation. The proposed solution is presented in Section V and evaluated in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

This section presents an overview of the state-of-the-art. These efforts tackle different objectives related to search and rescue (SAR) missions, time-aware disaster management, and wildfire detection and tracking in robotic systems.

SAR: Several works are focused on hazardous SAR scenarios. Cooperative SAR works [5], [6] divide disaster areas among robots and optimize paths to improve coverage speed, balance workload, and reduce environmental uncertainty. Perception-assisted SAR works [7], [8] integrate computer vision, target positioning, and autonomous navigation to detect and localize victims from aerial imagery. Some works [9]–[11] propose decentralized and resource-aware approaches related to path planning, detection verification, communication management, and energy consumption under constrained or unavailable communication infrastructure.

Time-Aware Disaster Management: Time-aware disaster management has been studied in the context of mission planning and resource management. In mission planning [10], [12]–[15], cooperative search, coverage, and task allocation are treated as optimization problems minimizing response, travel, or victim-discovery time while considering task deadlines, energy budgets, operational range, payload, fleet size, and uncertain task locations. In resource management [16], [17], works optimize sensing duration, computation offloading, communication resources, and robot trajectories to satisfy latency constraints while reducing end-to-end delay or energy consumption.

Wildfire Detection: Works [18]–[21] include methods for fire-front tracking and coverage through distributed control, persistent monitoring, evolutionary or reinforcement learning-based approaches. In [22], fire propagation dynamics are considered to coordinate coverage over a planning horizon. Complementary, there are energy-aware approaches that adapt image resolution or select flight and detection parameters to reduce the energy cost of onboard wildfire detection [23], [24].

Table I summarizes the key distinctions of D³ARC compared to the most relevant works focusing on time-awareness and wildfire detection. As it can be inferred from the table, there are studies that directly or indirectly promote coverage efficiency during a mission, while some consider heterogeneous robotic perception capabilities. However, only a few works integrate neural network-based forward-looking reasoning in their decision-making. Asynchronous cooperation is limited, as the majority of the works rely on centralized, synchronized, or unspecified coordination. Most of the works do not integrate an AI-based detection model, as they assume that fire observations are already available. Beyond the characteristics explicitly reported in the table, there are no works that jointly consider mission time, detection reliability, network variability, and resource availability within a unified

framework. These limitations motivate D³ARC, which integrates these aspects through distributed hierarchical cooperative decision-making for fast and reliable wildfire detection under dynamic conditions.

Study	Coverage Efficiency	Forward-Looking Reasoning	Asynchronous Cooperation	AI-based Detection	Heterogeneous Perception
[12], [16]	x	x	x	x	✓
[14], [21]	✓	✓	x	x	x
[18], [19], [22]	✓	x	x	x	x
[13]	x	x	✓	x	x
[10]	✓	x	x	✓	x
[17]	x	✓	x	x	x
[20]	✓	x	x	x	✓
[24]	✓	x	x	✓	x
[23]	x	x	x	✓	x
D ³ ARC	✓	✓	✓	✓	✓

TABLE I: State-of-the-art Comparison Table: Time-Awareness & Wildfire Detection

III. SYSTEM MODEL

This section presents the system model comprising multiple agents, a set of UAV robots $\mathcal{R}$ and a remote controller, denoted by RC. All agents are equipped with AI capabilities, tasked with a mission to cooperatively explore an unknown environment for potential wildfire incidents. Each robot $r \in \mathcal{R}$ is capable of sensing the environment and performing onboard wildfire detection, while the RC asynchronously handles each robot's motion and is also equipped with a wildfire detection algorithm l_{RC} to assist each robot when needed.

Time is represented by asynchronous timesteps indexed by $k \in \mathbb{N}$. A new timestep occurs when one or more robots complete their assigned tasks. Let $\mathbf{t}_k \in \mathbb{R}_+$ denote the wall-clock time at which timestep k starts. Any robot whose task is completed within a tolerance $\mathbf{t}_1$ of $\mathbf{t}_k$ is considered *ready* at the same timestep and does not trigger a new one. We denote the set of robots that are *ready* at k by $\mathcal{R}_k^{\text{ready}}$. The duration of k is the elapsed time from the beginning of k to the beginning of timestep $k+1$ and is defined as $T(k) = \mathbf{t}_{k+1} - \mathbf{t}_k$. The cumulative elapsed time starting from the first timestep ($k=1$) until the end of k is denoted by $T_{\text{total}}(k) = \sum_{\tau=1}^k T(\tau) = \mathbf{t}_{k+1} - \mathbf{t}_1$. When a robot r becomes *ready*, the RC asynchronously assigns its next trajectory and pose, denoted by $\mathcal{P}_{k,r}$, without waiting for the other robots to complete their tasks. For simplicity, trajectory and pose selection will be referred to as motion command. The execution time of this motion command varies across robots and timesteps and is denoted by $t_{k,r} \in \mathbb{R}_+$. Each robot spends time to offer a series of capabilities:

Robot Movement Capabilities: Each robot can move along the three axes, rotate 360°, and is able to actuate its joints to control the orientation of its onboard sensors (e.g. camera gimbal). Let $\{x_{k,r}, y_{k,r}, z_{k,r}\}$ denote robot's r movement along axes, $\mathbf{r}_{k,r}$ denote the robot's body rotation and $\mathbf{so}_{k,r}$ the orientation of all available sensors at k .

Robot Sensing Capabilities: At each timestep, each robot senses the environment for a fixed time period denoted by $\mathbf{s} \in \mathbb{R}_+^1$. Let $\mathcal{S}_r$ denote the set of sensors available at robot

r . Let $I_{s,r}$ denote the sensing sample rate of sensor $s \in \mathcal{S}_r$ and $w_{s,r}$ the size of each data sample.

Robot Computational Capabilities: We assume time-varying computational resources at each robot. As such, let $f_{k,r}, n_{k,r}^{\text{cpu}} \in \mathbb{R}_+$ denote the available computational capacity (i.e., CPU speed) and number of CPU cores of robot r available at timestep k , and $\varsigma_r \in \mathbb{R}_+$ the CPU's effective switched capacitance (ESC). At each timestep, each robot can complete a certain number of Floating Point Operations (FLOPs) per cycle, denoted by $c_{k,r} \in \mathbb{R}_+$.

Robot Perception Capabilities: Each robot perceives the environment to perform wildfire detection. A set of AI algorithms L is available for each robot to select at each k . The computational complexity $\alpha_{l_{k,r}} \in \mathbb{R}_+$ of a selected AI algorithm $l_{k,r} \in L$, depends on the architecture and input size, and is measured in FLOPs. The average performance of an AI algorithm at k is denoted by $\eta_{l_{k,r}}$ and could refer to metrics such as confidence score and accuracy. The time required by robot r to process one sample using $l_{k,r}$ is denoted by $\tau_{\alpha_{l_{k,r}}} \in \mathbb{R}_+$. Regardless of the selected detection algorithm, robots may have heterogeneous detection capabilities. Their perception can be affected by external factors such as image blur, illumination variations, reflections, and viewpoint-related artifacts. Based on their ability to produce reliable detections, robots are therefore classified as either *standard-detection robots* or *limited-detection robots*.

Robot Communication Capabilities: At each k , each robot may request assistance from the RC by offloading its sensing data. Then, the RC executes l_{RC} and achieves an average performance $\eta_{l_{RC,k,r}}$. The communication conditions between a robot and the RC are time-varying and the channel is modeled as a flat-fading with Gaussian noise power density $N_0 \in \mathbb{R}$ and channel gain $g_{k,r} \in \mathbb{R}$, where the fading is assumed constant during the timestep. Also, let $b_{k,r} \in \mathbb{R}_+$ and $p_{k,r} \in \mathbb{R}_+$ be the available bandwidth and transmission power of robot r , respectively. By $d_{k,r} \in \mathbb{R}_+$ we denote the achievable data rate of the communication channel between the RC and robot r at k . Table II summarizes all the notations of the system model.

IV. PROBLEM FORMULATION

Both the RC and the robots are tasked with cooperatively detecting a wildfire incident with a required level of confidence. However, satisfying a detection confidence requirement alone is not sufficient, since robot operation involves time-consuming activities, including mobility, sensing, computation, and communication. The objective is to detect a wildfire incident with a certain confidence as fast as possible under an imposed time limit.

Achieving the objective requires a chain of hierarchical, distributed, and interdependent decisions taken by both the RC and the robots. Each agent controls complementary parameters and operates under a different view of the same environment, necessitating coordinated decision-making.

Decision-Making Stage 1 (@RC): At the beginning of each k , the first stage of decision-making takes place at the RC. The RC asynchronously decides upon a motion command ($\mathcal{P}_{k,r}$)

¹We assume that $\mathbf{s}$ is sufficient to acquire newly generated sensing data.

Parameter	Description
k	The k^{th} global timestep
t_k	The wall-clock time at which timestep k is triggered
RC	The AI-enabled remote controller
$\mathcal{R}$	Full set of robots
$\mathcal{R}_k^{ready}$	Set of robots ready to receive new RC command at k
$\mathcal{P}_{k,r}$	The motion command of robot r at k
$t_{k,r}$	The time required for robot r to take an assigned pose at k
$\mathbf{s}$	The sensing period for all robots
β_r^{max}	Robot's r maximum battery capacity
$\beta_{k,r}$	Robot's r available battery capacity at k
$f_{k,r}$	Robot's r available computational capacity at k
$n_{k,r}^{cpu}$	The number of robot's r available CPU cores at k
$c_{k,r}$	The number of FLOPs per cycle robot r can complete at k
c_r	Robot's r CPU effective switched capacitance
S_r	The set of installed robot's r sensors
$I_{s,r}$	The sensing sample rate of robot's r sensor s
$w_{s,r}$	The data sample size of robot's r sensor s
L	The set of AI algorithms available for all robots
$\alpha_{l_{k,r}}$	The computational complexity of the AI algorithm $l_{k,r}$ at k
l_{RC}	The AI algorithm at the remote controller RC
$\alpha_{l_{RC},k,r}$	The computational complexity of the RC's l_{RC} using robot's r sensing data at k
$\eta_{l_{k,r}}$	The average performance of $l_{k,r}$ executed at k
$\eta_{l_{RC},k,r}$	The average performance of l_{RC} executed at k
$\tau_{\alpha_{l_{k,r}}}$	The computation time to execute $l_{k,r}$ at k
$d_{k,r}$	Robot's r achievable transmission data rate at k
N_0	The white Gaussian noise power spectral density
$g_{k,r}$	The gain of the wireless channel robot r has access to at k
$b_{k,r}$	Robot's r allocated bandwidth at k
$p_{k,r}$	The transmission power of robot r at k

TABLE II: Notation Table

for each $r \in \mathcal{R}_k^{ready}$. The assigned command will influence all subsequent robot's r decisions, as motion precision affects the robot's perspective and can facilitate progress toward achieving the end goal and vice versa.

A. Requirements for Robot Movement and Sensing

To follow the RC's motion command, each robot executes multiple movements. Three motion types are considered: hovering, horizontal, and vertical movement, each with distinct time costs. More specifically:

The total time required by each robot $r \in \mathcal{R}_k^{ready}$ to complete a motion command at k , denoted by $t_{k,r}$, is expressed as:

$$t_{k,r} = \sum_{m \in \mathcal{M}} t_{m,k,r}, \quad r \in \mathcal{R}_k^{ready} \quad (1)$$

where:

$$t_{m,k,r} = \frac{\delta_{k,r}^m}{v_{k,r}^m}, \quad (2)$$

$m \in \mathcal{M} = \{\text{hovering}, \text{horizontal}, \text{vertical}\}$ represents the set of movement types, $t_{m,k,r}$ denotes the time required by robot r to complete each movement at k , $\delta_{k,r}^m$ is the distance traveled by robot r along movement type m at step k , and $v_{k,r}^m$ is the corresponding speed of robot r .

Decision-Making Stage 2 (@Robot): Following the motion command (at $t_{k,r}$) and before sensing the environment, robot r decides on a binary variable $o_{k,r}$ indicating whether to offload the detection task or proceed with local execution. In the case of local execution, robot r also decides on an appropriate AI algorithm $l_{k,r}$.

Figure 1 illustrates the timeline of a robot $r \in \mathcal{R}_k^{ready}$, showing the chain of decisions between the RC and the robot, their dependencies, and when each decision is made.

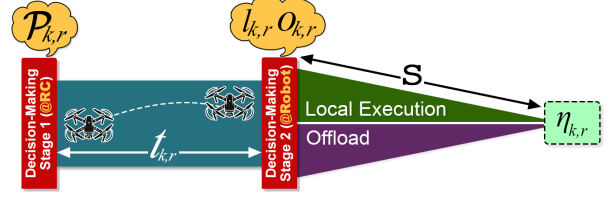


Fig. 1: Robot Decision-Making Stages

Each robot, after $t_{k,r}$, senses the environment for a fixed time period $\mathbf{s}$ using its set of sensors S_r .

B. Requirements for Local Execution

In case of local execution, each robot processes the sensing data locally using the selected AI algorithm $l_{k,r}$. The computation time required by robot r ($r \in \mathcal{R}_k^{ready}$) for executing $l_{k,r}$, assuming perfect parallelization across $n_{k,r}^{cpu}$ cores, is denoted by $T_{computation}$ is given by:

$$T_{computation}(\alpha_{l_{k,r}}) = \frac{N(I_{s,r}, \mathbf{s}) \cdot \alpha_{l_{k,r}}}{c_{k,r} \cdot n_{k,r}^{cpu} \cdot f_{k,r}} \quad (seconds), \quad (3)$$

where:

$$N(I_{s,r}, \mathbf{s}) = \sum_{s \in S_r} I_{s,r} \cdot \mathbf{s}, \quad (4)$$

denotes the number of data samples collected during the sensing period $\mathbf{s}$ to be processed by $l_{k,r}$.

C. Requirements for Offloading

In case of offloading, each robot $r \in \mathcal{R}_k^{ready}$ transmits the sensing data for remote wildfire detection. We assume that the RC has high computational capacity, and thus its processing time is negligible. Hence, only the data transmission time is considered, denoted by $T_{transmission}(k, r)$, and defined as:

$$T_{transmission}(k, r) = \frac{\sum_{s \in S_r} N(I_{s,r}, \mathbf{s}) \cdot w_{s,r}}{d_{k,r}} \quad (seconds), \quad (5)$$

where the modeling of the achievable data rate $d_{k,r}$ at k is provided in the Appendix.

D. Detection Confidence

Each robot senses the environment over a time window $\mathbf{s}$ and performs wildfire detection. The detection confidence level may fluctuate over time due to variations in sensing quality, introducing detection uncertainty (Section III, **Robot Perception Capabilities**). To capture this, we introduce a temporal consistency measure, named the Stable Confidence Score (SCS), to quantify detection stability over $\mathbf{s}$. The SCS of a robot r denotes the confidence level obtained at the end of k by performing either local detection ($\eta_{l_{k,r}}$), or remote detection ($\eta_{l_{RC},k,r}$), depending on the robot's decision. For simplicity, the selected score is denoted by $\eta_{k,r}$ and is defined as follows:

$$\eta_{k,r} = \max\left(0, \overline{cs}_{k,r} \cdot \left(1 - \frac{\sigma_{cs_{k,r}}}{\overline{cs}_{k,r}}\right)\right), \quad (6)$$

where $\overline{cs}_{k,r}$ denotes the average confidence over $\mathbf{s}$, and $\sigma_{cs_{k,r}}$ represents the corresponding standard deviation. The ratio $\frac{\sigma_{cs_{k,r}}}{\overline{cs}_{k,r}}$ is the coefficient of variation, measuring dispersion around the mean.

As such, at the end of k , individual detection confidences are combined to compute the cooperative detection confidence η_k as follows:

$$\eta_k = \max_{i \in \mathcal{R}_k^{\text{ready}} \cup \{RC\}} \eta_{k,i}. \quad (7)$$

Figure 2 shows an example of 3 robots across an timeline excerpt of 4 asynchronous timesteps. The cooperative confidence η_k is computed from Robots 1 and 3 at k , Robot 3 at $k+1$, and all robots at $k+2$. The total duration until the end of $k+2$ is $T_{\text{total}}(k+2) = \sum_{\tau=1}^{k+2} T(\tau) = t_{k+3} - t_1$.

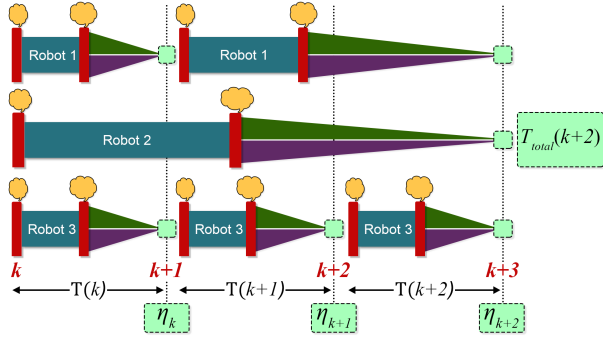


Fig. 2: Illustrative Timeline Excerpt for 3 Robots

E. Problem Objective

The objective is to detect a wildfire with a certain confidence as early as possible under a time limit requirement. As such, the optimization aims to minimize the total mission time $T_{\text{total}}(k^*)$, where k^* is the first timestep at which the cooperative detection confidence η_{k^*} reaches or exceeds the required confidence threshold $\eta_{\min}$. The resulting formulation is given below:

$$\min_{\{\mathbf{P}_k, \mathbf{o}_k, \mathbf{l}_k\}_{k=1}^{k^*}} T_{\text{total}}(k^*) \quad (8)$$

where

$$k^* = \min\{k \mid \eta_k \geq \eta_{\min}\} \quad (9)$$

$$T_{\text{total}}(k^*) = \sum_{\tau=1}^{k^*} T(\tau) = t_{k^*+1} - t_1, \quad (10)$$

$$\text{s.t. } \eta_{k^*} \geq \eta_{\min}, \quad (11)$$

$$\mathbf{p}_{k,i,r}^{\min} \leq \mathbf{p}_{k,i,r} \leq \mathbf{p}_{k,i,r}^{\max}, \quad \forall \mathbf{p}_{k,i,r} \in \mathcal{P}_{k,r}, \quad \forall i \in \{1, \dots, |\mathcal{P}_{k,r}|\}, \quad (12)$$

$$T_{\text{total}}(k^*) \leq t_{thr}. \quad (13)$$

The vectors $\mathbf{P}_k = (\mathcal{P}_{k,r})_{r \in \mathcal{R}_k^{\text{ready}}}$, $\mathbf{o}_k = (o_{k,r})_{r \in \mathcal{R}_k^{\text{ready}}}$ and $\mathbf{l}_k = (l_{k,r})_{r \in \mathcal{R}_k^{\text{ready}}}$ denote the motion commands selected by the RC, and the decisions made by the robots during k . Constraint (11) ensures a minimum acceptable cooperative

performance. Constraint (12) ensures that the assigned motion command is bounded by the hardware specifications of robot r . Constraint (13) ensures that the elapsed wall-clock mission time up to timestep k^* does not exceed the imposed threshold t_{thr} .

To enforce the constraints and accelerate execution, we introduce a penalty function for violations of the minimum detection confidence, motion bounds, and duration of each timestep. The penalty is defined as follows:

$$P_{\text{constr}}(k) = \max(0, \eta_{\min} - \eta_k) + \sum_{r \in \mathcal{R}_k^{\text{ready}}} \sum_{i=1}^{|\mathcal{P}_{k,r}|} \max(0, p_{k,i,r} - p_{k,i,r}^{\max}, p_{k,i,r}^{\min} - p_{k,i,r}), \quad (14)$$

$$P_T(k) = T(k) \quad (15)$$

Overall, the complete objective function that takes into account any introduced penalties is defined as:

$$\min_{\{\mathbf{P}_k, \mathbf{o}_k, \mathbf{l}_k\}_{k=1}^{k^*}} \left(\mu_1 \cdot T_{\text{total}}(k^*) + \sum_{k=1}^{k^*} (\mu_2 \cdot P_{\text{constr}}(k) + \mu_3 \cdot P_T(k)) \right) \quad (16)$$

where μ_1 , μ_2 and μ_3 are the weights of the equation, controlling the impact of each term. It should be noted that each term of the objective and penalty functions is min-max normalized and then divided by its mean to ensure comparable average contributions before applying the objective weights.

V. D³ARC: DISTRIBUTED HIERARCHICAL INTELLIGENCE FOR COOPERATIVE ROBOTS

This section introduces D³ARC, an asynchronous distributed hierarchical cooperative decision-making framework designed to solve our time-aware optimization problem. The framework comprises two phases: offline warmup and online real-time execution. The RC and the robots have different views of the environment and take different yet interdependent actions that are jointly evaluated using a reward function.

A. State, Action, Reward

State of the Environment: Each agent has a different, yet complementary, view of the environment, defined as the state (S). The RC maintains a higher-level perspective as it does not directly interact with the environment, whereas the robots operate with a concrete, interaction-driven view.

RC State: The RC, at each timestep, inspects a set of variables $S_{\text{RC},k}$ that guides its decision-making. This set includes the robot's position along the three axes ($x_{k-1,r}$, $y_{k-1,r}$, $z_{k-1,r}$) and pose, i.e. body's and sensors' orientation ($\mathbf{r}_{k-1,r}$ and $\mathbf{so}_{k-1,r}$), its distance from robot r ($\delta_{k-1,\text{RC},r}$), as well as the cooperative detection confidence SCS (η_{k-1}) and elapsed time ($T(k-1)$) observed at the previous timestep, as an outcome of the preceding joint decisions of the involved agents. Overall:

$$S_{\text{RC},k} = \{x_{k-1,r}, y_{k-1,r}, z_{k-1,r}, \mathbf{r}_{k-1,r}, \mathbf{so}_{k-1,r}, \delta_{k-1,\text{RC},r}, \eta_{k-1}, T(k-1)\}.$$

Robot State: Each robot after executing RC's motion command, inspects its own set of parameters $S_{k,r}$. These parameters include its pose ($\mathbf{r}_{k,r}, \mathbf{so}_{k,r}$), the cpu frequency ratio ($f_{k,r}^{\text{ratio}}$), its distance from the RC ($\delta_{k,\text{RC},r}$), the allocated bandwidth ($b_{k,r}$), the achievable transmission data rate ($\mathbf{d}_{k,r}$), as well as the SCS (η_{k-1}) and elapsed time ($T(k-1)$) observed at the previous timestep, as an outcome of the preceding joint decisions of the involved agents. Overall:

$$S_{k,r} = \{ \mathbf{r}_{k,r}, \mathbf{so}_{k,r}, f_{k,r}^{\text{ratio}}, \delta_{k,\text{RC},r}, b_{k,r}, \mathbf{d}_{k,r}, \eta_{k-1}, T(k-1) \}, \quad (17)$$

where:

$$f_{k,r}^{\text{ratio}} = \frac{f_{k,r}}{f_r^{\text{max}}}, \quad (18)$$

and f_r^{max} denotes the maximum CPU frequency of robot r .

Action Space: The action space, is comprised of all decisions made by all agents at timestep k .

RC Actions: As $A_{\text{RC},k,r}$ we denote the action space of the RC, that represents assigned actions related to the movement of robot r along the three axes ($\delta_{k,r}^{(x)}, \delta_{k,r}^{(y)}, \delta_{k,r}^{(z)}$), its body's and sensors' rotation ($\delta_{k,r}^{(x)}, \delta_{k,r}^{(\text{so})}$). Overall:

$$A_{\text{RC},k,r} = \{ \delta_{k,r}^{(x)}, \delta_{k,r}^{(y)}, \delta_{k,r}^{(z)}, \delta_{k,r}^{(x)}, \delta_{k,r}^{(\text{so})} \}.$$

Robot Actions: As $A_{k,r}$ we denote the decision-making of each robot r regarding the offloading of the detection task $o_{k,r}$ and the model selection $l_{k,r}$. Overall:

$$A_{k,r} = \{ o_{k,r}, l_{k,r} \}.$$

Mutual Reward: The reward retrieved at the end of each timestep is mutual for all agents and is derived from the corresponding optimization objective (Eq. (16)). The reward is designed to promote the main objective while discouraging infeasible or undesirable decisions. As such, the mutual reward is defined as:

$$rr_k = \mu_1 \cdot \eta_k - \mu_2 \cdot P_{\text{constr}}(k) - \mu_3 \cdot P_T(k). \quad (19)$$

Thus, maximizing rr_k directly promotes higher cooperative detection confidence towards reaching the imposed threshold as fast as possible, while penalizing constraint violations associated with the selected actions.

B. Offline Warmup Phase

This phase enables all agents to acquire initial experience, and prevents a cold-start. During warmup, and at each timestep, the RC and ready robots randomly select actions, evaluate the resulting decisions using rr_k , and store the corresponding states, actions, and rewards until sufficient training data are collected.

Agent Forward-Looking and Training: Observing the timeline within a timestep k (Figure 1), a pair of agents have to take sequential and complementary actions at different times. However, a single shared reward is retrieved only at the end of k , forcing each agent to *look forward* by estimating the shared outcome of its actions before the complete decision sequence is executed. Most importantly, the RC, lacking knowledge of

each robot's subsequent actions, evaluates the outcome of its own decision while anticipating the expected effect of the robot's subsequent decisions on the final shared reward. This *forward-looking* is the ability of all agents to anticipate the consequences of alternative actions by estimating outcomes that have not yet been observed at decision time [25].

To enable *forward-looking*, one regression feedforward neural network (RFNN) model is trained for the RC and one for all robots in a supervised manner. Each model learns to estimate the resulting shared reward (model output) using as input features the agent's respective state and actions, i.e. $S_{k,r} \cup A_{k,r}$ for all robots and $S_{\text{RC},k,r} \cup A_{\text{RC},k,r}$ for the RC. The rewards collected during this phase serve as ground truth for training. These trained models are used during the online phase for real-time execution.

C. Online Real-Time Execution Phase

This phase leverages the experience obtained during the offline training to enable cooperative, real-time, chained decision-making between agents. The agents use their pre-trained RFNN models and follow a greedy-based strategy. At each k , the RC evaluates a set of N candidate alternative future actions, which are sampled independently from a uniform distribution over the feasible action space², while each robot evaluates all possible actions. As such, each agent exploits its acquired *forward-looking* knowledge: it estimates the anticipated reward for each candidate action, and greedily selects the one that maximizes this reward.

To complement and accelerate cooperative decision-making during the online phase, D³ARC incorporates four additional mechanisms that support safe operation while promoting exploration, cooperation and system efficiency. These mechanisms are detailed below.

Obstacle Avoidance Mechanism (@Robot): The monitored environment is not obstacle-free. While the RC maintains a higher-level view of the environment, each robot has access to a more concrete local perspective and can detect nearby obstacles through onboard sensing, e.g., LiDAR. As such, it would be impractical for a robot to blindly follow every RC's motion command. D³ARC incorporates a 360° obstacle avoidance mechanism at the robot side to ensure safe and reliable exploration of the surrounding environment. This mechanism suppresses motion commands that would drive the robot toward obstacles located within a minimum safety distance. Specifically, when an obstacle is detected along the direction of motion and its distance from the robot falls below the minimum safety threshold sd_{min} , the robot autonomously halts movement in this direction.

Coverage Efficiency Mechanism (@RC): Since, the RC greedily selects among a number of candidate actions at each k , it may repeatedly select neighboring motion commands confining the robot's motion to a nearby vicinity. To enhance environment coverage efficiency, D³ARC introduces a coverage efficiency mechanism at the RC side, driven by η_k . The

²In large action spaces, exhaustive evaluation is infeasible, and as such a limited candidate set enables efficient local search.

environment is partitioned into fixed-size cubic regions of size crs . As each robot r navigates across these regions, the RC maintains a region-wise logging history $h_{cr,k}$, which records the number of robot visits $vs_{cr,k}$, and maximum SCS $\eta_{cr,k}^{max}$ observed across visits until k . Based on this history, each region is categorized as:

- **Unexplored:** Unvisited regions ($cr \notin h_{cr,k}$)
- **Uninformative:** Visited regions with zero average SCS ($cr \in h_{cr,k}$, $\eta_{cr,k}^{max} = 0$)
- **Promising:** Visited regions with non-zero maximum SCS ($cr \in h_{cr,k}$, $\eta_{cr,k}^{max} > 0$)

To improve coverage efficiency, the RC assigns an area prioritization penalty $AP_{cr,r}(k)$ to each candidate action for robot r . This penalty is based on the history of the region that will be reached following that action. The penalty is defined as:

$$AP_{cr,r}(k) = \begin{cases} 1, & cr \notin h_{cr,k}, \\ vs_{cr,k} \cdot \rho, & cr \in h_{cr,k}, \eta_{cr,k}^{max} = 0, \\ \max(0, \eta_{min} - \eta_{cr,k}^{max}), & cr \in h_{cr,k}, \eta_{cr,k}^{max} > 0. \end{cases} \quad (20)$$

where $\rho \geq 1$ controls how strongly repeated visits to *uninformative* regions are penalized. As such the reward function is updated as follows:

$$rr_k = \mu_1 \cdot \eta_k - \mu_2 \cdot P_{constr}(k) - \mu_3 \cdot P_T(k) - \mu_4 \cdot \sum_{r \in \mathcal{R}_k^{ready}} AP_{cr,r}(k), \quad (21)$$

where μ_4 is a weight controlling the impact of the area prioritization term. Since RC aims to minimize the $AP_{cr,r}(k)$ during candidate action selection, *promising* regions with high previously observed SCS receive the lowest penalties and are prioritized first. *Unexplored* regions receive an intermediate penalty, while repeatedly visited *uninformative* regions receive increasingly larger penalties. Once again, each term is normalized according to its own scale.

Call-for-Aid (@Robot): To further improve cooperative sensing around *promising* regions, D³ARC incorporates at the robot side a Call-for-Aid (CFA) mechanism. This mechanism directs nearby robots toward a common region of interest, allowing them to collect complementary observations and jointly reduce detection uncertainty. More specifically, a robot broadcasts a CFA message to the RC and all neighboring robots within a communication radius (δ_{CFA}), when it detects a wildfire with the highest confidence observed so far, yet insufficient to declare a detection. This confidence must exceed a threshold η_{cfa} ($\eta_{cfa} < \eta_{min}$). This threshold ensures that the CFA is triggered only when the detection confidence is high enough, preventing false interruptions to other robots' exploration and avoiding unnecessary time loss. The broadcast message includes the position and body rotation associated with the confidence observation.

After receiving the CFA message, the RC stops providing motion commands to the CFA robots to avoid conflicting with their cooperative actions, while relaying information about

the *promising* region to robots that may not have received the original broadcast. CFA robots determine their subsequent movement based on the reported high-confidence position and body rotation. Robots located too far from the region, for which participation would introduce excessive travel time, remain under the RC's control and continue their current operations. Each CFA robot evaluates a set of N candidate actions, sampled once again independently from a uniform distribution over the feasible action space. The evaluation favors actions that move the robot toward the *promising* region, reduce its distance to the reported position, and align each robot's body rotation with that rotation of the robot that broadcast the CFA message.

To promote position and rotation alignment with the robot reported the CFA message r_{ref} , a new penalty $P_{CFA}(k, r, r_{ref})$ is introduced during CFA, defined as follows:

$$P_{CFA}(k, r, r_{ref}) = \lambda_1 \cdot \delta_{k,r,r_{ref}} + \lambda_2 \cdot al_{k,r,r_{ref}} + \lambda_3 \cdot y_{k,r,r_{ref}}, \quad (22)$$

where $\delta_{k,r,r_{ref}}$ is the distance between the robot's r candidate position and the reported position by robot r_{ref} , $al_{k,r,r_{ref}}$ is the alignment between the candidate motion direction and the direction toward the reported position, and $y_{k,r,r_{ref}}$ is the rotation distance between the robot's candidate rotation assignment and the reported rotation by robot r_{ref} . The weights λ_1 , λ_2 and λ_3 are fixed design parameters that balance the relative contribution of distance, direction, and rotation alignment, respectively. Overall, lower values of $P_{CFA}(k, r, r_{ref})$ correspond to actions that better support the CFA request.

During CFA, participating robots temporarily prioritize reaching the *promising* region over sensing at intermediate locations. Accordingly, the robots increase their motion speed and reduce their sensing duration by a factor fv_{CFA} . This allows them to reach the *promising* region faster while still sensing the environment along the way, in case a better view-point of the event is found. Once a robot reaches the *promising* region, it exits the CFA mechanism individually and returns to normal RC controlled operation, restoring its normal motion speed and sensing duration. The CFA mechanism terminates when all participating robots have reached the region.

Adaptive Early Mission Completion Mechanism (@RC): Even upon wildfire detection, the RC may select actions, which can cause abrupt perspective changes and unnecessary motion. To counteract this, D³ARC employs an adaptive early mission completion mechanism, improving efficiency through motion stabilization or mission termination, when the cooperative SCS indicates that further motion is unnecessary. Specifically, under reliable detections by at least one robot (line 3), the RC commands the respective robots to hover in place while continuing their perception tasks. Other robots, remain under normal RC control and continue their assigned tasks. If at least one hovering robot maintains a confidence greater than or equal to η_{min} for *esp* consecutive timesteps, the wildfire detection is considered confirmed and the mission is successfully terminated for all robots (line 5). For intermediate confidence levels (line 8), the RC reduces the action value ranges $\mathcal{A}_{RC,k,r}$ by a fixed value $fv_a \in \mathbf{fv}, \forall a \in \mathcal{A}_{RC,k,r}$

(line 11). This limits abrupt changes in robot r 's perspective and encourages finer exploration around the current viewpoint, which may contain useful evidence of a wildfire. If intermediate confidence persists for c_{thr} consecutive timesteps, this indicates that the finer exploration has not produced stronger evidence. Therefore, more drastic corrective measures are applied by expanding the action set using the same value (line 13), allowing the robot to explore more diverse viewpoints and potentially reduce this uncertainty. If the cooperative SCS falls below the intermediate confidence threshold, the detection is no longer considered sufficient. As such, the action value ranges are reset and standard D³ARC operation resumes (line 16).

Algorithm: D³ARC Adaptive Early Mission Completion

Input: $\eta_{min}, \eta_{int}, esp, c_{thr}, \mathbf{fv}$
1: Initialize $c_{stop,r} \leftarrow 0, c_{low,r} \leftarrow 0$
2: **for** each timestep k **do**
3: **if** $\eta_{k-1,r} \geq \eta_{min}$ **then**
4: $hover; c_{stop,r} \leftarrow c_{stop,r} + 1; c_{low,r} \leftarrow 0$
5: **if** $c_{stop,r} \geq esp$ **then**
6: $complete\ mission$
7: **end if**
8: **else if** $\eta_{int} \leq \eta_{k-1,r} < \eta_{min}$ **then**
9: $c_{stop,r} \leftarrow 0; c_{low,r} \leftarrow c_{low,r} + 1$
10: **if** $c_{low,r} < c_{thr}$ **then**
11: $\mathcal{A}_{RC,k,r} \leftarrow \{a/fv_a \mid fv_a \in \mathbf{fv}, a \in \mathcal{A}_{RC,k,r}\}$
12: **else**
13: $\mathcal{A}_{RC,k,r} \leftarrow \{a + fv_a \mid fv_a \in \mathbf{fv}, a \in \mathcal{A}_{RC,k,r}\}$
14: **end if**
15: **else**
16: $reset\ \mathcal{A}_{RC,k,r}; c_{stop,r} \leftarrow 0; c_{low,r} \leftarrow 0$
17: **end if**
18: **end for**

VI. PERFORMANCE EVALUATION

Mission: The mission incorporates up to 4 UAV robots. Each UAV robot is equipped with a set of sensors for wildfire detection and safe navigation, i.e. optical camera, LiDAR, GPS, Inertial Measurement Unit (IMU), and two computer vision (CV) detection models. All robots cooperate with a single RC, having its own CV model, and is capable of controlling the robots' motion along axes, rotation, and camera roll. Together, these agents are tasked with detecting a wildfire incident with the objective to meet a required detection confidence as fast as possible, while ensuring that the detection task is completed within a predetermined time limit.

Accordingly, we structure the evaluation around three key points: **1)** the trade-off between time and mission reliability, **2)** the importance of hierarchical coordination and robotic cooperation and **3)** the effectiveness of D³ARC compared to non-cooperative and heuristic baselines.

Evaluation Metrics: To quantitatively evaluate the performance of D³ARC, Table III summarizes the evaluated metrics. All metrics are averaged across missions and robots. The look-forward accuracy (LFA) is expressed as:

$$LFA = \frac{\sum_{r \in \mathcal{R}} (1 - \frac{MAE_r}{MAE_r^{\max}})}{|\mathcal{R}|} \cdot 100\%,$$

where the MAE_r expresses the mean absolute error between the anticipated and actual reward for robot r in a single mission, and $MAE_r^{\max}$ expresses the maximum MAE for robot r across missions.

Metric	Definition
Total Time (min)	Total mission time
Computation Time (min)	Time for CV execution
Transmission Time (min)	Time during sensing data transmission
Movement Time (min)	Time taken during movement
MTLR (%)	Mission Time Limit Reached
TUFD (min)	Time Until First valid Detection
MD_{AVG} (min)	Average Mission Duration
MD_{Median} (min)	Median Mission Duration
MTP (%)	Movement Time Percentage
CTP (%)	Computation Time Percentage
TTP (%)	Transmission Time Percentage
FPDs (%)	False Positive Detections Rate
MSCS (%)	Stable Confidence Score over a mission
MLFA (%)	Look-Forward Accuracy
MSR (%)	Mission Success Rate, excluding FPDs

TABLE III: Evaluation Metrics Definition

A. Simulation Setup

This section outlines our simulation ecosystem, including: **1)** the 3D environment, **2)** the UAV robots, **3)** the CV detection models, and **4)** the wireless communication network.

3D Environment: The simulation environment is built upon Gazebo 11 [26] and Robot Operating System 2 (ROS2) [27], incorporating rigid-body dynamics and collision physics through the Open Dynamics Engine [28]. These components provide a physics-aware simulation setting for evaluating robotic behavior under realistic terrain and environmental conditions. Figure 3 illustrates the environment that includes a custom 530 x 530 x 165 m (width x depth x height) mountainous forest scene along with a 46 m high wildfire (including smoke), all designed in Blender [29].

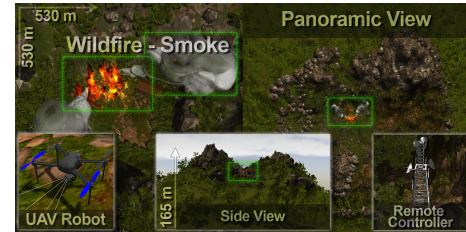


Fig. 3: Simulation Environment inside Gazebo

UAV Robots: The simulated UAV robots' properties were highly inspired by the 3DR Iris quadcopter to provide a realistic reference for the robot models [30]. Table VII given in the Appendix, summarizes the UAV robot configuration (same configuration is applied to all robots), including the set of sensors, hardware specifications, computational and communication capabilities, and key physical properties.

CV Detection Models: Two custom convolutional neural network (CNN) models are available for wildfire detection: a Full CNN and a Quantized CNN, differing in computational complexity. Both models are available for all UAV robots, while the RC uses only the Full one. Details regarding their

training and configuration are provided in the Appendix (Table VIII).

Wireless Communication Network: The communication model considers a single Base Station (BS). The RC is assumed to be collocated with the BS, and as such share the same physical location and jointly represent the communication endpoint of each UAV robot. A 2.4 GHz wireless link connecting each robot and the RC is characterized by distance-dependent path loss, shadow fading, bounded bandwidth, and spectral efficiency. The transmit power is determined with respect to a target received signal strength, while the complete network parameterization is provided in Appendix (Table IX).

Ecosystem Dynamicity: To simulate realistic conditions, introduce dynamicity and uncertainty into the environment and evaluate the generalization ability of D³ARC, several key parameters are varied at each timestep. These variations capture changes in: **1)** robot’s resource availability, **2)** wildfire locations and **3)** detection capabilities.

- **Resource availability:** The computational and communication resources vary across timesteps and robots (CPU speed, assigned bandwidth). This also applies to the wireless communication quality, including degradation with distance and fluctuations in the achievable data rate and transmission power requirements (Section IV).
- **Wildfire locations:** The environment includes 14 distinct wildfire locations, with one wildfire incident per mission, and two possible BS locations. The distance between each wildfire location and the corresponding BS ranges from 18 to 275 m. The wildfire locations introduce detection difficulty (e.g. distant and obscured), while alternative BS placements impact connectivity and detection reliability.
- **Detection capabilities:** To emulate the heterogeneous perception capabilities introduced in Section III, at each mission we randomly select a subset of robots (up to 50% of the total robots) as *limited-detection* UAVs and cap their detection confidence below the mission threshold ($\eta_{\min}$). The remaining robots operate as *standard-detection* UAVs without this restriction. This prevents all robots from independently achieving reliable detections under ideal sensing conditions and promotes scenarios where successful mission completion depends on stronger coordination and cooperative sensing and detection.

B. D³ARC Configuration Setup

Table V summarizes the configuration of D³ARC regarding the region size, motion command ranges, thresholds and key algorithmic parameters. Table IV summarizes the configuration for the RFNN models used by each UAV Robot and the RC. Overall, 115K samples, a batch size of 128, the Adam optimizer with learning rate 5e-06 and 2000 epochs were used for the offline training phase with 85%, 10%, and 5% of the samples used for the training, validation and evaluation stages. To avoid overfitting, dropout layers were used with 10% probability of zeroing neural connections along with an early stopping where the patience was set to 10 epochs. All hyperparameters are empirically tuned during the offline

warmup phase based on mission success, detection reliability, and training stability. Once set, all parameter values remain fixed across the online evaluation missions.

Entity RFNN	# Input Features	# RFNN Layers: [Shape]
UAV Robot	11 ($S_{k,r} \cup A_{k,r}$)	3: [128 x 64 x 32]
RC	13 ($S_{RC,k,r} \cup A_{RC,k,r}$)	4: [128 x 64 x 64 x 32]

TABLE IV: D³ARC: RFNNs for the UAV Robot & RC

Parameter	Value
Cubic Region Size (crs)	60x60x60m
Axes Movement Range ($\delta^{(x)}, \delta^{(y)}, \delta^{(z)}$)	[-20,20]m
Body Rotation Range ($\delta^{(y)}$)	[-90,90]°
Camera Roll Range ($\delta^{(so)}$)	[0,90]°
Time tolerance (τ_1)	200ms
# Candidate Actions (N)	1000
Penalty Weights ($\lambda_1, \lambda_2, \lambda_3$)	1,1,1
Objective Weights ($\mu_1, \mu_2, \mu_3, \mu_4$)	1,2,1,2
Minimum Confidence Threshold ($\eta_{\min}$)	75%
Intermediate Confidence Threshold (η_{int})	20%
Decrease Factors (fv)	[3,10,10]%
CFA Distance Radius (δ_{CFA})	100m
CFA Factor (fv_{CFA})	2
Early Mission Completion Patience (esp)	2
Consecutive Low Confidence Threshold (c_{thr})	2

TABLE V: D³ARC: Configuration Setup

C. Time-Aware Multi-Robot: Evaluation Results

This section presents the evaluation of D³ARC under time-sensitive wildfire detection requirements. The evaluation follows a progressive analysis of 3 rounds, with 300 missions per setup, to examine the main factors that affect mission reliability, cooperative sensing, detection performance, and scalability. First, a single robot case is evaluated under different mission time limits to study whether increasing the available time is sufficient to achieve reliable detection. Then, an ablation study is conducted by varying the number of robots and comparing D³ARC under cooperative and non cooperative configurations. This aims to show that the benefits do not come solely from increasing the number of robots, but also from targeted cooperation among them. Finally, D³ARC is compared against a distributed heuristic cooperative strategy to evaluate its benefit over a baseline.

Results 1: Single Robot Time Limit Sensitivity

This first round of experiments evaluates the performance of a single *standard-detection* UAV robot under different mission time limits (3, 5, 10 min). The goal is to examine whether increasing the available mission time is sufficient to improve wildfire detection reliability, or whether the single robot setup remains limited by its own sensing and exploration capabilities.

Results 1 (Figure 4) - Observations:

- The mean LF accuracy (MLFA) remains stable around 80% across all thresholds, indicating high decision-making quality for both agents.
- The average mission duration (MD_{avg}) and time until first detection (TUF_D) remain on average at ~48% and ~35% of their respective thresholds, indicating that most

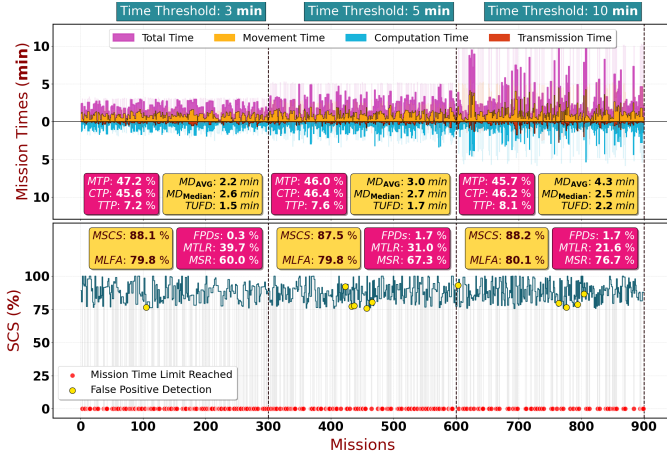


Fig. 4: [Single Robot]: Performance for [3, 5, 10] minutes

missions are completed early. Higher time thresholds increase mission duration as they allow further exploration of harder cases.

- Movement and computation dominate the total mission time, with each accounting for $\sim 46\%$ of the mission duration, while transmission remains below $\sim 8\%$.
- As time threshold increases, the percentage of missions reaching the time limit (MTLR) decreases. This indicates that the robot is allowed to search more regions and increase the probability to meet the detection performance requirement.
- However, increasing the time threshold by more than three times, reaching the maximum acceptable by EUSPA (10 min) [2], the mission success ratio (MSR) still remains under 77%, although detection performance remains high ($\uparrow$ MSCS, $\downarrow$ FPDs) and consistent across all time limits.

Results 1 - Conclusion: The single robot results show that increasing the time limit improves overall performance. However, all remarks reveal an important limitation of the single robot setup. This limitation is not the quality of the detection once the robot reaches a promising region, but the robot's ability to consistently reach such regions within the required time. In simple words, *the robot can detect reliably, yet time is not enough*. This motivates the transition to a multi-robot setup, where cooperation can provide additional sensing perspectives, faster coverage, and stronger support for uncertain detections.

Results 2: Impact of Multi-Robot Cooperation under a Strict Time Limit

Transitioning into a multi-robot setup, this round of results quantifies the impact of cooperation under the strictest mission time limit of 3 min, and for up to 4 UAV robots. The single robot is included for reference. The goal is to examine whether increasing only the number of robots improves mission reliability, or explicit cooperation is required. To compare D³ARC with and without cooperation, we enable/disable the two cooperative mechanisms, i.e. the coverage efficiency and

call-for-aid (Section V). In each multi-robot case, 50% of the robots are configured as *limited-detection* UAVs.

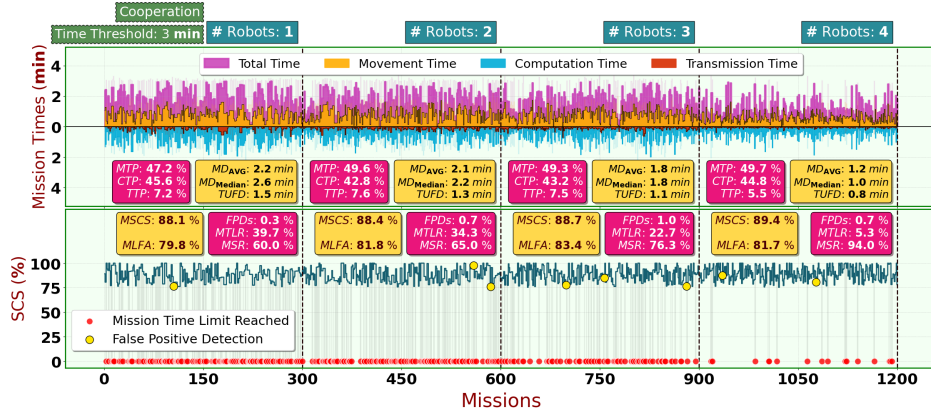
Results 2 (Figure 5) - Observations:

- Looking at the MSR, the benefit of increasing the number of robots becomes evident. Compared with the single robot case, cooperation improves the MSR by 5%, 16.3%, and 34%, for 2, 3, and 4 robots, respectively, while without cooperation, by 0.3%, 3%, and 22%.
- Cooperation provides an additional gain in the MSR of 4.7%, 13.3%, and 12% over the non-cooperative setup. This highlights that even when multiple robots are available, coordinated support is necessary to use the robots' capabilities effectively and avoid relying on independent detections alone.
- Cooperation also reduces time limit violations, decreasing the MTLR by 4.4%, 13.3%, and 11% as robots increase, compared with the non-cooperative setup. This shows that cooperation does not only improve mission success rates, but facilitates robots to satisfy the detection objective earlier and more reliably.
- Cooperation reduces MD_{avg} as the number of robots increases, with more evident reductions of 14% and 20% for 3 and 4 robots, respectively. It also significantly lowers TUFD, reaching 0.8 min with 4 robots, showing that cooperation guides robots toward promising regions faster.
- The cooperative setup maintains high MSCS and low FPDs, reaching 89.4% and 0.7%, respectively, with 4 robots, compared with 87.6% and 1.7% without cooperation. This shows that cooperation accelerates mission completion while also preserving stronger detection confidence and almost eliminating incorrect wildfire detections.
- The MLFA improves under cooperation, indicating that cooperative information facilitate the agents selecting more effective actions toward promising regions.
- The time distribution also changes with cooperation. Cooperative configurations spend a higher amount of time on movement, especially in the case of 4 robots, where MTP is 23% higher, while the non-cooperative setup spends more time on computation. This is mainly a result of D³ARC's coverage efficiency mechanism, which discourages repeated visits to *uninformative* regions and incentivizes the robots to continue exploring the environment.

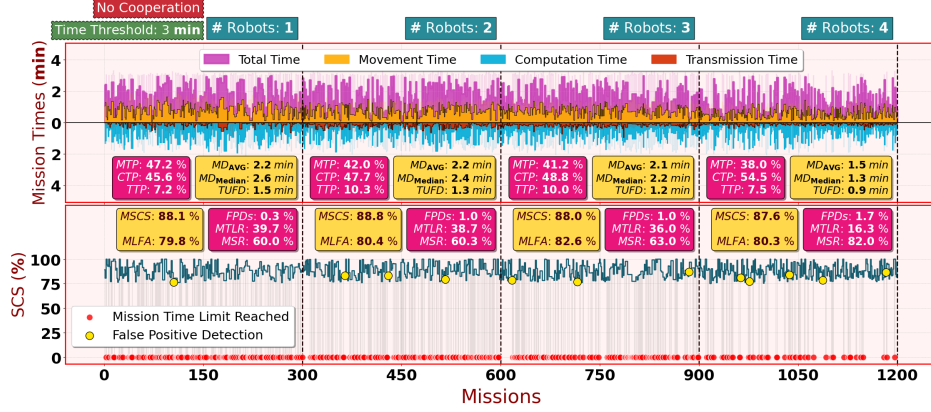
Results 2 - Conclusion: The results show that cooperation can convert additional robots into effective sensing support under strict time constraints. This results in higher mission success, fewer time limit violations, shorter mission duration, higher coverage, and earlier detections compared to the non-cooperative configuration. In simple words, *multiple robots may travel far, yet cooperative ones arrive with purpose*.

Results 3: Baseline Comparison

This final round of evaluation results compares the cooperative D³ARC and non-cooperative (marked as D³ARC) against CH-RBS, a Rule-Based Strategy (see Appendix A) that



(a) Cooperative Setup



(b) Non-Cooperative Setup

Fig. 5: [Multi-Robot]: Performance for the strictest time threshold of 3 minutes.

integrates the cooperative mechanisms of D³ARC. The D³ARC is included just for reference. This time we examine the merits of adaptive intelligence realized through the hierarchical distributed agentic decision-making of D³ARC. The comparison is conducted using the 3 robot setup under the strict 3 min time threshold. This setting represents a challenging middle-ground scenario that maintains coordination and cooperation complexity.

# Robots: 3 - Time Threshold: 3 minutes				
Strategy	Avg. Mission Duration (min)	Avg. TUFD (min)	MSCS (%)	Avg. MSR (%)
D ³ ARC	1.8	1.1	88.7	76.3
D ³ AR	2.1	1.2	88.0	63.0
CH-RBS	2.3	1.6	87.5	44.0

TABLE VI: Comparison of D³ARC against Baselines

Results 3 (Table VI) - Observations:

- D³ARC achieves the highest MSR, reaching 76.3%, compared to 44% of CH-RBS. This large gap indicates that heuristic-driven cooperation rules are not sufficient under dynamic ecosystem conditions. D³ARC adapts agentic decision-making, allowing the robots to use cooperation more effectively.

- D³ARC achieves the shortest average mission duration, reducing it from 2.3 min in CH-RBS to 1.8 min. This shows that D³ARC does not only increase mission success, but also enables faster completion under the same time threshold.
- The average TUFD is also 32% lower compared to CH-RBS. This indicates that D³ARC supports robots in detecting informative wildfire evidence earlier in the mission.
- Although the differences in MSCS are moderate, they show that the faster mission completion and highest MSR of D³ARC is not achieved at the cost of detection confidence.

Results 3 - Conclusion: The results confirm that by combining multiple robots, cooperative sensing and hierarchical distributed agentic decision-making, D³ARC dominates. In simple words, *cooperation matters, yet adaptive intelligence makes the difference.*

VII. CONCLUSIONS

This paper introduces D³ARC, a time-aware distributed, hierarchical, and cooperative framework integrating multiple robots and a remote controller for wildfire detection under *uncertainty*. The controller decides each robot's motion, while

each robot determines *where* and *how* to perform detection. All agents aim to detect wildfire with a certain performance as fast as possible under an imposed time limit. D³ARC integrates an innovative forward-looking approach, along with four key mechanisms that support safe operation while promoting exploration, cooperation and system efficiency.

The results demonstrate the effectiveness of D³ARC. Single robot experiments show reliable detections, but with a time cost. Multi-robot cooperation improves coverage speed and mission duration, enables earlier and reliable detections, and achieves mission success up to 94% with 89.4% detection confidence. Comparisons with the non-cooperative D³AR and the cooperative heuristic CH-RBS confirm the merits of D³ARC's adaptive, confidence-driven, and time-aware distributed yet cooperative decision-making.

REFERENCES

- [1] World Meteorological Organization, "State of the Global Climate 2025," World Meteorological Organization, Geneva, Switzerland, Tech. Rep. WMO-No. 1391, Mar. 2026. [Online]. Available: <https://library.wmo.int/records/item/69807-state-of-the-global-climate-2025>
- [2] "User needs and requirements: Report on emergency management and humanitarian aid," European Union Agency for the Space Programme (EUSPA), Tech. Rep., 2024. [Online]. Available: www.euspa.europa.eu/publications-multimedia/publications/user-needs-and-requirements
- [3] M. Ghassemian, J. Erfanian, K. Althoefer, K. Ahmadi, J. Eichinger, A. Meseguer Valenzuela, P. Barattini, M. Bahaei, T. Booth, A. Bouttier, C. Ciochina, M. Mestoukirdi, P. Gonçalves, E. Natalizio, J. Simonjan, M. Basaran, F. Cogen, Y. Ozsahin, A. Huete, and T. Mahmoodi, "6g architectural foundations and ai-native solutions for future connected robotics." 03 2026.
- [4] 6G-IA Vision Working Group, "European vision for the 6g network ecosystem," 6G-IA, Tech. Rep. Version 2.0, 2024. [Online]. Available: <https://6g-ia.eu/wp-content/uploads/2024/11/european-vision-for-the-6g-network-ecosystem.pdf>
- [5] Y. Xiong, Y. Zhou, J. She, and A. Yu, "Collaborative coverage path planning for uav swarm for multi-region post-disaster assessment," *Vehicular Communications*, vol. 53, p. 100915, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214209625000427>
- [6] A. Aminzadeh and A. M. Khoshnood, "Multi-uav cooperative search and coverage control in post-disaster assessment: Experimental implementation," *Intelligent Service Robotics*, vol. 16, no. 4, pp. 415–430, 2023. [Online]. Available: <https://doi.org/10.1007/s11370-023-00476-4>
- [7] K. Rashida Farsath, K. Jitha, V. Mohammed Marwan, A. Muhammed Ali Jouhar, K. Muhammed Farseen, and K. Musrifa, "Ai-enhanced unmanned aerial vehicles for search and rescue operations," in *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT)*, 2024, pp. 1–10.
- [8] L. Xing, X. Fan, Y. Dong, Z. Xiong, L. Xing, Y. Yang, H. Bai, and C. Zhou, "Multi-uav cooperative system for search and rescue based on yolov5," *International Journal of Disaster Risk Reduction*, vol. 76, p. 102972, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214209622001911>
- [9] J. Horyna, T. Baca, V. Walter, D. Albani, D. Hert, E. Ferrante, and M. Saska, "Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication," *Auton. Robots*, vol. 47, no. 1, p. 77–93, Oct. 2022. [Online]. Available: <https://doi.org/10.1007/s10514-022-10066-5>
- [10] A. Romero, C. Delgado, L. Zanzi, R. Suárez, and X. Costa-Pérez, "Cellular-enabled collaborative robots planning and operations for search-and-rescue scenarios," 2024. [Online]. Available: <https://arxiv.org/abs/2403.09177>
- [11] I. Medeiros, A. Boukerche, and E. Cerqueira, "Swarm-based and energy-aware unmanned aerial vehicle system for video delivery of mobile objects," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 766–779, 2022.
- [12] Z. Beck, W. L. Teacy, A. Rogers, and N. R. Jennings, "Collaborative online planning for automated victim search in disaster response," *Robotics and Autonomous Systems*, vol. 100, pp. 251–266, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889016307515>
- [13] P. Ghassemi and S. Chowdhury, "Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints," *Robotics and Autonomous Systems*, vol. 147, p. 103905, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889021001901>
- [14] A. Khanal, J. P. Mathew, C. Nowzari, and G. J. Stein, "Learning-augmented model-based multi-robot planning for time-critical search and inspection under uncertainty," 2025. [Online]. Available: <https://arxiv.org/abs/2507.06129>
- [15] D. Han, H. Jiang, L. Wang, X. Zhu, Y. Chen, and Q. Yu, "Collaborative task allocation and optimization solution for unmanned aerial vehicles in search and rescue," *Drones*, vol. 8, no. 4, 2024. [Online]. Available: <https://www.mdpi.com/2504-446X/8/4/138>
- [16] R. Yin, Y. Shen, H. Zhu, X. Chen, and C. Wu, "Time-critical tasks implementation in mec based multi-robot cooperation systems," *China Communications*, vol. 19, no. 4, pp. 199–215, 2022.
- [17] Y. Lu, C. Xu, and Y. Wang, "Joint computation offloading and trajectory optimization for edge computing uav: A knn-ddpg algorithm," *Drones*, vol. 8, no. 10, 2024. [Online]. Available: <https://www.mdpi.com/2504-446X/8/10/564>
- [18] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1537–1548, 2020.
- [19] K. Shrestha, R. Dubey, A. Singandhupe, S. Louis, and H. La, "Multi objective uav network deployment for dynamic fire coverage," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 1280–1287.
- [20] N. Patrinoopoulou, I. Daramouskas, D. Meimetis, V. Lappas, and V. Kostopoulos, "A distributed framework for persistent wildfire monitoring with fixed wing uavs," *Drones and Autonomous Vehicles*, vol. 1, no. 3, p. 10009, 2024.
- [21] K. D. Julian and M. J. Kochenderfer, "Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning," 2018. [Online]. Available: <https://arxiv.org/abs/1810.04244>
- [22] E. Seraj, A. Silva, and M. Gombolay, "Multi-uav planning for cooperative wildfire coverage and tracking with quality-of-service guarantees," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 2, p. 39, 2022. [Online]. Available: <https://doi.org/10.1007/s10458-022-09566-6>
- [23] B. Akpomedaye, A. Shalan, N. A. Walee, and M. Rahman, "Energy-efficient uav surveillance for early wildfire detection using ai-driven image analysis," in *2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA)*, 2025, pp. 1–7.
- [24] J. Suo, X. Zhang, W. Shi, and W. Zhou, "E3-uav: An edge-based energy-efficient object detection system for unmanned aerial vehicles," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4398–4413, 2024.
- [25] X. Luo, A. Rechardt, G. Sun, K. K. Nejad, F. Yáñez, B. Yilmaz, K. Lee, A. O. Cohen, V. Borghesani, A. Pashkov, D. Marinazzo, J. Nicholas, A. Salatiello, I. Sucholutsky, P. Minervini, S. Razavi, R. Rocca, E. Yusifov, T. Okalova, N. Gu, M. Ferienc, M. Khona, K. R. Patil, P.-S. Lee, R. Mata, N. E. Myers, J. K. Bizley, S. Musslick, I. P. Bilgin, G. Niso, J. M. Ales, M. Gaebler, N. A. R. Murty, L. Loued-Khenissi, A. Behler, C. M. Hall, J. Dafflon, S. D. Bao, and B. C. Love, "Large language models surpass human experts in predicting neuroscience results," *Nature Human Behaviour*, 2024. [Online]. Available: <https://doi.org/10.1038/s41562-024-02046-9>
- [26] Official gazebo website. [Online]. Available: <https://gazebo.sim.org>
- [27] Official ros website. [Online]. Available: <https://www.ros.org/>
- [28] Open dynamics engine. [Online]. Available: [https://www.ode.org/](https://www ode.org/)
- [29] Blender free & open source 3d software. [Online]. Available: <https://www.blender.org/download/>
- [30] 3dr iris quadcopter specifications. [Online]. Available: <https://www.ardupilot.org/co.uk/iris-quadcopter-uav.html>
- [31] Dfire dataset official github. [Online]. Available: <https://github.com/gaiasd/DFireDataset>
- [32] P. V. A. B. de Venâncio, R. J. Campos, T. M. Rezende, A. C. Lisboa, and A. V. Barbosa, "A hybrid method for fire detection based on spatial and temporal patterns," *Neural Computing and*

Applications, vol. 35, no. 13, pp. 9349–9361, 2023. [Online]. Available: <https://doi.org/10.1007/s00521-023-08260-2>

- [33] M4sfwd dataset official github. [Online]. Available: <https://github.com/Philharmy-Wang/M4SFWD>
- [34] G. Wang, “Multiple scenarios, multiple weather conditions, multiple lighting conditions and multiple wildfire objects synthetic forest wildfire dataset (m4sfwd),” 2024. [Online]. Available: <https://dx.doi.org/10.21227/m9kz-bw61>

APPENDIX

A. Achievable Data Rate Modeling

The achievable data rate $d_{k,r}$ of each robot $r \in \mathcal{R}_k^{ready}$ is given by:

$$d_{k,r} = b_{k,r} \cdot \log_2 \left(1 + \frac{g_{k,r} \cdot p_{k,r}}{b_{k,r} \cdot N_0} \right), \quad (23)$$

where:

$$g_{k,r} = 10^{-\frac{L_0 + 10 \cdot e \cdot \log_{10}(\frac{\delta_{k,RC,r}}{\delta_0})}{10}}, \quad (24)$$

$$L_0 = 20 \cdot \log_{10}(\delta_0) + 20 \cdot \log_{10}(f_k) + 20 \cdot \log_{10}\left(\frac{4 \cdot \pi}{c}\right), \quad (25)$$

$$p_k = tp_k + L_0. \quad (26)$$

tp_k is the target signal strength (dBm) L_0 is the path loss (dB) at the reference distance δ_0 (meters), $\delta_{k,RC,r}$ is the distance between the robot and the RC at k (meters), f_k denotes the transmission carrier frequency (Hz), e is the path loss exponent and c the speed of light (m/s).

B. UAV Robot Specifications

Table VII summarizes the UAV robot’s configuration. Several abbreviations are used in the table: Frames Per Second (FPS), Width (×) Height (W×H), and Height/Width/Depth (H/W/D).

Parameter	Value
# of LiDARs	6
LiDAR Safety Distance (sd_{min})	2 m
Camera FPS (I_{camera})	11
Camera Resolution (W×H)	1920 × 1080
Camera Frame Size (w_{camera})	519 KByte
Sensing Period (s)	3 s

(a) Sensing & Perception

Parameter	Value
CPU Speed (f_{min}/f_{max})	1.5/3.2 GHz
CPU Cores (n^{cpu})	6
CPU ESC (ς)	10^{-28}
CPU FLOPs/Cycle (c)	8
Transmission Power (p^{min}/p^{max})	14/33 dBm

(b) Computation & Communication

Parameter	Value
Mass	1.816 kg
# of Propellers	4
Propeller Radius	100 mm
Speed (v)	5 m/s

(c) Physical Specifications

TABLE VII: UAV Robot Configuration Setup

C. Computer Vision Detection Model Configuration

The Full and Quantized CV models were trained offline on publicly available datasets, namely DFire [31], [32] and M⁴SFWD [33], [34], which together contain approximately 25K diverse images. Their configurations are summarized in Table VIII.

Configuration	Full CNN	Quantized CNN
# of Layers	11 convolutional + 2 fully connected	
Activation Function	LeakyReLU	
Learning Rate	2×10^{-4}	
Precision	float32	qint8
Trainable Parameters	36.5M	
Complexity (GFLOPs)	20.23	5.05

TABLE VIII: CV models used for wildfire event detection

D. Wireless Communication Network Specifications

Table IX provides the configuration of the network model.

Parameter	Value
Target Signal Strength (tp_k)	-80 dBm
RC Antenna Frequency (f_k)	2.4 GHz
Reference Distance (δ_0)	1 m
Bandwidth min/max (b_{min}/b_{max})	15/30 MHz
Spectral Efficiency min/max (se_{min}/se_{max})	0.12/4
Noise Spectral Density (N_0)	-158 dBm/Hz
Path Loss Exponent (e)	3.8
Shadow Fading	8 dB

TABLE IX: Network Configuration Setup

E. Heuristic Rule-based Strategy (H-RBS)

H-RBS performs a random-walk inspection at each timestep and determines whether to execute the task locally or offload it by comparing the transmission ratio $\frac{d_{k,r}}{d_{k,r}^{max}}$ and the processing ratio $\frac{f_{k,r}}{f_{k,r}^{max}}$, of each robot $r \in \mathcal{R}_k^{ready}$, against predefined thresholds $thr_1 = 0.85$ and $thr_2 = 0.65$. H-RBS selects the strategy whose corresponding ratio exceeds its threshold. If both exceed, the choice is random, while if neither does, the strategy with the minimum distance from its respective threshold is selected. H-RBS integrates the early mission completion mechanism of D³ARC.