Market-Based Replanning for Safety-Critical UAV Swarms in Search and Rescue Missions

Luiz Giacomossi, Andrea Haglund, Claire Namatovu, Emily Zainali, Esaias Målqvist,
Yonatan M. Beyene, Ivan Tomasic, Baran Çyrüklü, and Håkan Forsberg
Mälardalen University (MDU)
Västerås, Sweden
luiz.giacomossi@mdh.se

***Abstract*—Reliable autonomous UAV swarms in Search and Rescue (SAR) missions require fault-tolerant coordination capable of sustaining operations despite agent degradation. This paper introduces the Intelligent Replanning Drone Swarm, a distributed coordination architecture designed for resource-constrained environments. The proposed framework employs a *Reverse-Auction* market mechanism where agents bid to service search sectors based on a distance-weighted cost function, coupled with a geometric consensus protocol for target verification. We evaluate the approach through physics-based simulations ($N = 8$ agents, 8×8 grid) subjected to stochastic fault injection. Results indicate that the swarm autonomously reallocates tasks from failed agents with low latency relative to the total mission duration, maintaining a mission success rate of 93% under 25% workforce degradation. The proposed framework demonstrates a robust, empirically tested method for self-healing aerial robotic coordination.**

***Keywords*—Fault Tolerance, Market-Based Coordination, Multi-Agent Systems, Search and Rescue, UAV Swarms.**

I. Introduction

Autonomous Unmanned Aerial Vehicle (UAV) swarms are increasingly utilized for Search and Rescue (SAR) [1], [2], defense operations [3], and disaster response due to their potential for redundancy and parallel task execution [4]. However, deploying these systems in uncontrolled real-world scenarios presents reliability challenges. A primary requirement is *Fault Tolerance*; specifically, the ability to sustain mission-critical functions despite individual agent failures (termed *Fail-Operational* at the mission level).

Centralized control architectures create a single point of failure and are limited by communication latency in large-scale deployments. Conversely, distributed approaches may lack the predictable safety properties required for critical operations. This paper addresses the problem of maintaining swarm integrity and mission continuity during agent degradation (e.g., battery depletion or crash).

We present *Intelligent Replanning Drone Swarm* (IRDS), a decentralized framework for resilient task allocation designed for resource-constrained environments (low-bandwidth, limited-compute and energy). The proposed method integrates a *Reverse-Auction Market*—where agents minimize the collective cost of service—with a spatial consensus protocol for target verification. This enables the swarm to detect node failures, modeled as "Liquidation events" (immediate task release), update the global system state, and redistribute search sectors to maintain coverage.

Our contributions are:

1) **Reverse-Auction Protocol:** We formalize a decentralized task allocation mechanism using distance-weighted pricing to minimize collective travel distance and improve target coverage.
2) **Resilient Architecture:** We propose a fault-tolerant state machine that treats agent failure as a market event, triggering automatic task reallocation without central intervention.
3) **Empirical Validation:** We validate robustness through physics-based simulations (PyBullet) with stochastic fault injection. Results demonstrate that the framework maintains mission success rates of 93% even under significant workforce loss.

The paper is organized as follows: Section II reviews related work; Section III formalizes the problem; Section IV details the IRDS architecture; Sections V and VI present and discuss results; Section VII concludes.

II. Related Work

Current research in multi-agent swarm coordination can be classified into centralized optimization, heuristic search strategies, and market-based approaches.

A. Heuristic and Probabilistic Search

Coordinate search is often achieved through shared world models or emergent behaviors. Probabilistic approaches typically integrate Bayesian updates with local gradients to guide agents toward high-entropy areas. For instance, Finite State Machines (FSM) coupled with potential fields have been shown to effectively coordinate collision-free swarms [2]. More advanced methods utilize probabilistic priors to reduce search times compared to blind lawn-mower patterns [1]. However, these heuristic approaches typically depend on continuous state synchronization and often lack built-in fault tolerance to automatically redistribute tasks when an agent fails.

B. Task Allocation: Optimization vs. Markets

The allocation of agents to tasks is a known problem in swarm robotics, often formulated as the Multi-Robot Task Allocation (MRTA) problem [5]. In defense contexts, global optimization techniques such as the *Hungarian*

Algorithm (Kuhn-Munkres) can guarantee optimal assignment of assets to threats [3]. However, global optimization typically entails a computational complexity of $O(N^3)$, which scales poorly for large swarms and creates a centralized bottleneck.

Market-based approaches offer a decentralized alternative by employing virtual economies where agents bid for tasks based on local cost functions [5], [6]. By prioritizing local cost minimization, auction mechanisms reduce algorithmic complexity to approximately $O(N \log N)$. This enables the high-frequency replanning required for dynamic SAR environments, avoiding the computational penalty of combinatorial optimization [7]. IRDS adopts this paradigm but simplifies the bidding process to distance-weighted metrics to minimize communication overhead.

C. Consensus in Ad-Hoc Networks

Reliable target verification in swarms requires distributed consensus [8] to filter false positives. While blockchain and Paxos algorithms offer cryptographically secure consistency, they impose bandwidth overheads often untenable for low-power drone meshes [9]. Consequently, IRDS utilizes a lightweight geometric voting protocol that decouples exploration from verification, ensuring that sensor noise does not derail the global mission.

III. Problem Formulation & Mathematical Model

We consider a coordination of N autonomous agents, $\mathcal{D} = \{d_1, \dots, d_N\}$, to search a bounded area $\mathcal{A} \subset \mathbb{R}^2$ discretized into M non-overlapping sectors $\mathcal{S} = \{s_1, \dots, s_M\}$. The state of agent i at time t is $\mathbf{x}_i(t) = [\mathbf{p}_i(t), b_i(t)]^T$, representing position and battery level. The objective is to maximize the covered area while minimizing collective travel distance, subject to dynamic constraints imposed by agent failures and communication limits.

A. Cost Function Definition

To quantify travel overhead of task execution, we define a *Service Cost* function based on proximity. The bid price $B_{ij}(t)$ for agent d_i to acquire sector s_j is modeled as:

$$B_{ij}(t) = C_{base} \cdot \left(1 + \frac{\|\mathbf{p}_i(t) - \mathbf{p}_{s_j}\|}{\delta}\right) \tag{1}$$

where C_{base} is the nominal base cost, $\mathbf{p}_{s_j}$ is the sector centroid, and δ is the distance scaling factor. This serves as the basis for the reverse-auction protocol (Sec.IV).

B. Dynamic Consensus Condition

Target verification is modeled as a health-aware spatial voting problem. Let $\mathcal{V}(t) \subseteq \mathcal{D}$ be the set of currently active recruited verifiers (neighboring agents temporarily assigned to confirm a detected target). The set is updated dynamically based on agent health $h_i(t)$:

$$\mathcal{V}(t) = \{d_v \in \mathcal{V}_{init} \mid h_v(t) = \text{OK}\} \tag{2}$$

A target is confirmed if and only if the number of positive confirmations N_{pos} satisfies the dynamic unanimity rule:

$$N_{pos}(t) = |\mathcal{V}(t)| \quad \text{where} \quad |\mathcal{V}(t)| > 0 \tag{3}$$

This mechanism removes crashed verifiers from the quorum during voting, preventing consensus deadlock caused by agent unavailability.

C. Distributed Collision Avoidance

Safety is enforced via a decentralized, modified heuristic Artificial Potential Field (APF) [10]. The repulsive force $\mathbf{F}_{ij}$ acting on agent d_i from neighbor d_j applies a squared inverse-distance barrier to penalize proximity:

$$\mathbf{F}_{ij} = \begin{cases} k_{rep}\left(\frac{1}{d_{ij}} - \frac{1}{r_{safe}}\right)^2 \frac{\mathbf{p}_{ij}}{d_{ij}} & \text{if } d_{ij} < r_{safe} \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{4}$$

where $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ is the Euclidean distance, $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ is the relative position vector, k_{rep} is the repulsion gain, and r_{safe} is the safety radius interaction horizon.

IV. Proposed Method: The IRDS Architecture

The IRDS architecture employs a hierarchical control strategy decomposing the mission into three layers: Global Task Allocation, Local Trajectory Generation, and Reactive Control. The system architecture is visualized in Fig. 1.

A. Layer 1: Market-Based Task Allocation

The search space $\mathcal{S}$ is treated as a set of discrete commodities. The allocation process is divided into two distinct phases.

1) Phase I: Static Initialization: At $t = 0$, to prevent initial clustering, the system executes a round-robin assignment. All sectors are assigned a fixed nominal cost C_{base}. Agents iteratively acquire an initial set of unassigned sectors to ensure a uniform spatial distribution across the search area.

2) Phase II: Dynamic Reverse Auction: During mission execution, task allocation shifts to a distance-weighted reverse auction. When a sector becomes available (i.e., is unassigned or released via liquidation), valid bids B_{ij} are established via the cost function in Eq. (1). The allocation logic clears the market by identifying the lowest global bidder:

$$(d^*, s^*) = \arg\min_{d_i \in \mathcal{D}} B_{ij}(t) \quad \text{s.t.} \quad W_i(t) \geq B_{ij}(t) \tag{5}$$

3) Fault Tolerance (Liquidation): Upon detection of a fault (e.g., `BAD_BATTERY`), the system executes a Liquidation Protocol: (1) All sectors owned by the faulty agent are released; (2) These sectors are re-auctioned immediately. Neighbors naturally win these auctions due to proximity, facilitating rapid autonomous takeover.

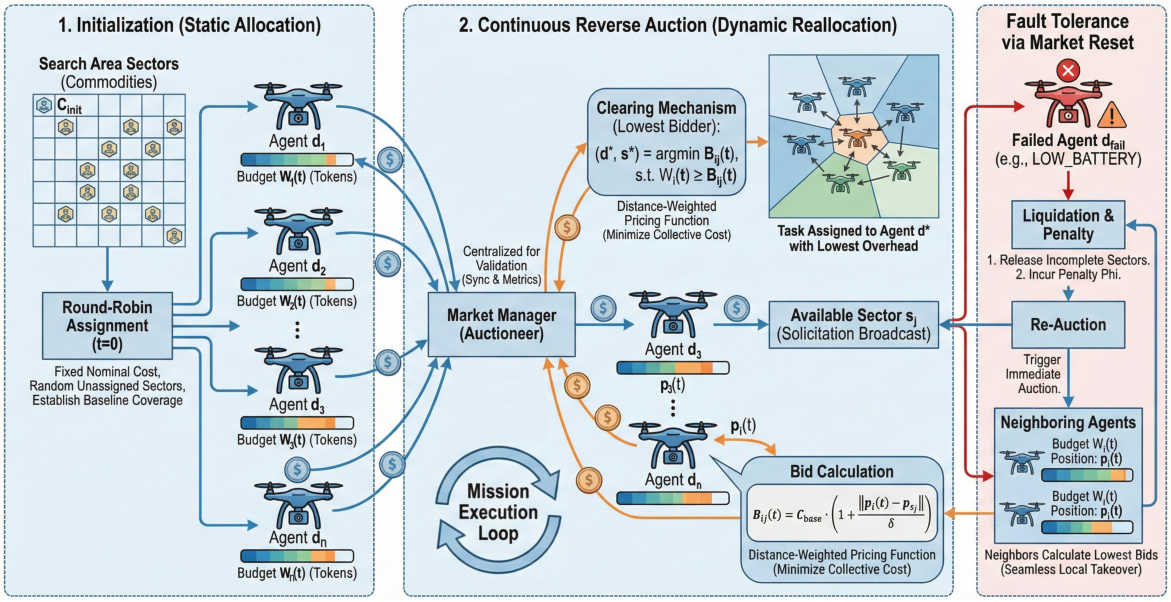

Fig. 1. Architectural overview of the IRDS Token-Based Reverse Auction. The system operates in three phases: (1) Static initialization; (2) Dynamic reallocation where agents minimize service cost; and (3) Fault recovery via market liquidation.

B. Layer 2: Coverage & Trajectory

Upon securing a task, the agent employs a deterministic *Boustrophedon Decomposition* [11]. This decomposition arranges trajectories in a "lawnmower" pattern to guarantee complete coverage of the assigned region. The planner generates a reference trajectory $\mathcal{P}_j$ relative to the section centroid, with sweep spacing determined by the sensor's Field of View (FOV). Simultaneously, the agent continuously computes the repulsive forces $\mathbf{F}_{ij}$ defined in Eq. (4) to modify the control input for real-time collision avoidance.

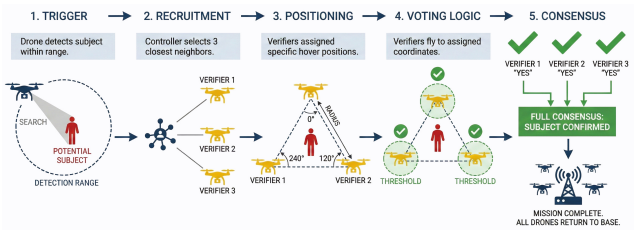

Fig. 2. Visual representation of the Multi-Agent Consensus Protocol. (1) Trigger: Detector identifies target. (2) Recruitment: The $k = 3$ nearest neighbors are tasked as verifiers. (3) Positioning: Verifiers form an equilateral triangle (geometric lock) around the target. (4) Consensus: Unanimity is required to confirm the target and terminate the mission.

C. Layer 3: Geometric Consensus Protocol

To validate target detection in noisy environments, IRDS implements the multi-stage voting process illustrated in Fig. 2.

1) **Recruitment:** Upon initial detection, the detecting agent d_{det} halts its search and recruits the $k = 3$ nearest neighbors to establish the candidate set $\mathcal{V}_{init}$.
2) **Geometric Positioning:** To ensure diverse vantage points, the verifiers are commanded to form an *Equilateral Triangle* formation (in the nominal $k = 3$ case) around the estimated target coordinates (0°, 120°, 240°).
3) **Voting Logic:** Once in position, each verifier casts a binary vote based on its local sensor reading.
4) **Consensus:** The mission concludes only if the dynamic unanimity condition ($N_{pos} = |\mathcal{V}(t)|$) is met. Agent failures during this phase automatically reduce $|\mathcal{V}(t)|$ (Eq. 3) to prevent deadlocks. If the condition is subsequently not met (i.e., healthy voters disagree), the target is discarded as a false positive and verifiers return to the market.

D. Implementation Note

The experiments utilize a central `MarketSystem` class within the simulation loop to manage currency and synchronization, acting as a proxy for a distributed ledger. The bidding logic, liquidation protocol, and consensus voting are computed locally by each agent and are decentralized by construction (only arbitration and metric collection are centralized in simulation). In a physical network, the auctioneer role maps to a distributed state machine synchronized via ad-hoc gossip protocols. Thus, real-world RF latency and packet loss can increase reallocation delays and introduce inconsistencies in the global belief state.

V. Experimental Evaluation

To validate the dependability and efficiency of the IRDS architecture, we conducted extensive simulations in a high-fidelity physics environment.

A. Operational Scenario Description

To evaluate the system's efficacy, we simulated a complete SAR mission. The operational sequence, seen in Fig. 3, proceeds through four distinct phases:

- **Deployment (Fig. 3a):** The swarm initializes at a central helipad (H). During this phase, the *Phase I* static market allocation distributes the initial set of target sectors to ensure uniform spatial distribution.
- **Distributed Search (Fig. 3b):** Upon reaching their assigned sectors, agents execute the *Layer 2* local path planner. The inset highlights the generated trajectories within the assigned $1.5m^2$ grid cells.
- **State Synchronization (Fig. 3c):** As sectors are cleared, agents update the global belief state. Green cells indicate fully searched regions; white cells remain effectively "on the market." This visual feedback confirms the functionality of the decentralized ledger in preventing redundant searching.
- **Target Verification (Fig. 3d):** Upon detecting a potential victim (red dotted circle), the searching agent triggers the *Layer 3* Consensus Protocol. The figure captures the "Geometric Lock" moment where neighboring agents (highlighted in yellow/blue dashed circles) converge to form the verification triangle, satisfying the unanimity condition required to confirm the target and conclude the mission.

B. Experimental Setup

The simulation is built upon the `gym-pybullet-drones` framework[1]. The search domain was defined as an 8×8 discretized grid (64 sectors total), where each sector represents a $1.5m \times 1.5m$ area. The target is randomly placed within the grid at the start of each episode.

- **Agents:** Bitcraze Crazyflie 2.x dynamics.
- **Market Config:** Base Cost $C_{base} = 2.0$, Distance Factor $\delta = 10.0$.
- **Statistical Rigor:** Each configuration is executed for $K = 100$ Monte Carlo trials for statistical significance given the stochastic nature of target placement.

C. Performance Metrics

Tab.I summarizes the Key Performance Indicators (KPIs) used to evaluate mission efficiency, system reliability, and market dynamics across all experiments.

D. Experimental Scenarios

1) Scenario A: Scalability Analysis: We evaluate performance limits by increasing swarm density $N \in \{2, 4, 8\}$ (doubling resources) under nominal conditions. The objective is to quantify how quickly the swarm can locate a target as agent count increases.

2) Scenario B: Fault Tolerance (Resilience): We fix the swarm size at $N = 8$ and inject stochastic faults:

- **Single Fault:** One agent triggers a `BAD_BATTERY` fault at $t = 8s$, forcing task release.
- **Double Fault:** Two agents fail in a sequence ($t = 8s, t = 15s$), reducing the workforce by 25%.

[1]Source code and documentation: github.com/luizgiacomossi/Intelligent-Drone-Swarm/tree/paper. Video of the simulator: figshare.com/s/1b36e110c99e376c9f67

TABLE I
KEY PERFORMANCE INDICATORS (KPIS) DEFINITION

Metric	Definition & Significance
1. Mission Efficiency	
Success Rate (η)	Percentage of trials where the target is identified and verified via consensus.
Duration (T_{total})	Time elapsed from mission start (t_0) until target verification.
Distance (D_{total})	The scalar sum of trajectories for all agents, serving as a direct proxy for total fleet energy consumption.
2. Reliability & Resilience	
Reallocation Latency ($L_{realloc}$)	Time elapsed between an agent's failure and the market-based reassignment of its active tasks.
Recovery Count (C_{rec})	The absolute number of tasks transferred from faulty agents to healthy neighbors to heal coverage gaps.
3. Economic Dynamics	
Bid Density	Average number of valid bids per auction, indicating network connectivity and swarm availability.
Gini Index (G)	Measures workload inequality (0 = balance, 1 = monopoly); used to quantify the "Cost of Resilience" during load shedding.

E. Quantitative Results

The aggregated data from $K = 100$ trials is presented in Table II and visualized in Fig. 4.

TABLE II
COMPREHENSIVE PERFORMANCE ANALYSIS (MEAN $\pm$ STD. DEV.)

Scenario	Succ. (η)	Time (T_{total})	Dist. (D_{total})	Recov. (C_{rec})	Lat. ($L_{realloc}$)	Bid Dens.	Gini (G)
N=2 (Baseline)	97%	721 ± 419	$\mathbf{346} \pm 187$	-	-	1.9 ± 0.3	0.02 ± 0.04
N=4 (Nominal)	**100%**	414 ± 238	400 ± 195	-	-	3.7 ± 1.1	0.04 ± 0.06
N=8 (Saturated)	99%	$\mathbf{223} \pm 190$	420 ± 196	-	-	7.3 ± 2.3	0.10 ± 0.15
Fault (1)	97%	307 ± 308	449 ± 240	6.5 ± 10.1	$\mathbf{13.1} \pm 21.2$	7.0 ± 2.7	0.17 ± 0.13
Fault (2)	93%	382 ± 394	453 ± 228	8.0 ± 12.5	17.7 ± 26.6	7.4 ± 2.1	0.27 ± 0.13

1) Scalability and Efficiency: Increasing the swarm size improves the time-to-find. As seen in Table II, the mean search duration drops from $721s$ ($N = 2$) to $223s$ ($N = 8$). Interestingly, the *Total Distance* metric increases slightly with swarm size ($346m$ to $420m$), reflecting the overhead of coordinating a larger fleet; however, this energy cost is justified by the $3.2\times$ speedup in mission completion.

2) Fault Tolerance Data: In Scenario B, the system kept high success rate (97.0%/93.0%), with failures limited to motion deadlocks (local minima) rather than sensor errors. The *Reallocation Latency* was measured at $13.1s$ and $17.7s$, respectively. Crucially, the *Recovery Count* ($C_{rec} \approx 6.5$) shows that if an agent fails, its sectors are immediately re-auctioned and picked up by neighbors, preventing "blind spots" where the target might be.

VI. DISCUSSION

The results supports the core hypothesis of IRDS: a decentralized and cost-minimizing market can serve as a robust proxy for complex centralized replanning in safety-critical environments. In this section, we analyze the emergent behaviors of the swarm and the trade-offs inherent in our proposed architecture.

A. Operational Resilience and Latency

A key metric for safety-critical SAR is the system's reaction speed to failure. As seen in Tab.II, the average Reallocation Latency ($L_{realloc}$) of roughly $13 - 18s$ represents approximately 4.3–4.6% of the total mission duration

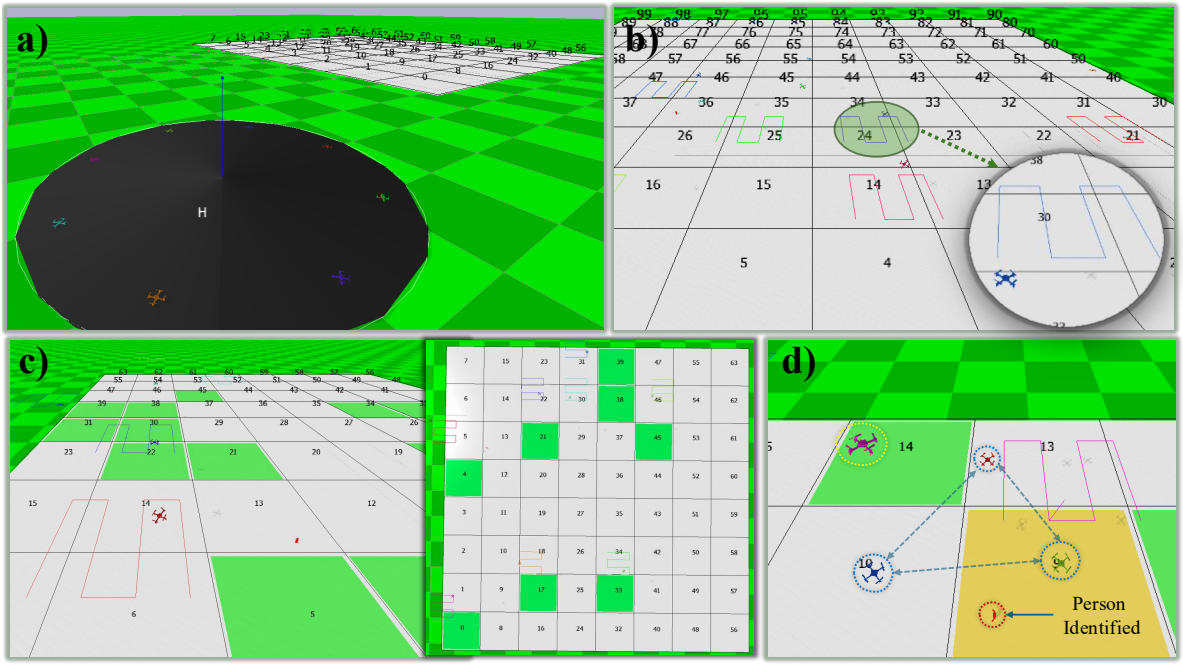

Fig. 3. Operational Sequence of the IRDS Architecture. **(a)** Swarm initialization and market bidding at the helipad. **(b)** Execution of local Boustrophedon coverage paths within assigned sectors. **(c)** Global state visualization where green cells denote cleared areas, validating the distributed memory. **(d)** Execution of the Geometric Consensus Protocol for mission: neighboring agents converge to verify a detected target (Person Identified), forming the required triangulation formation.

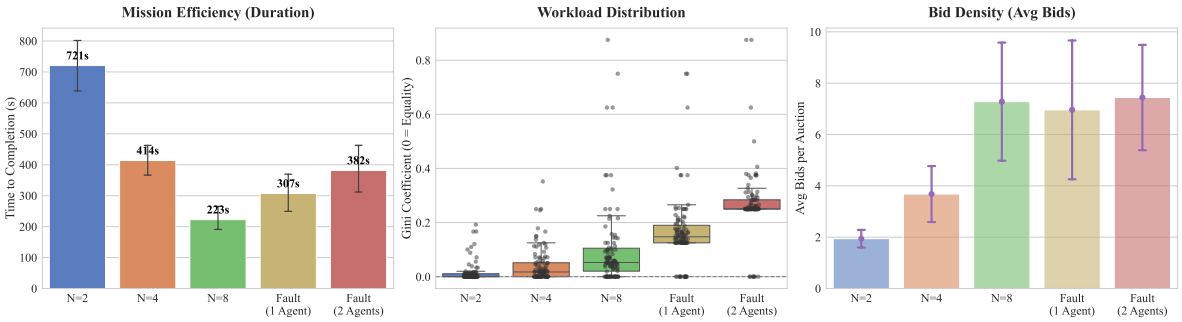

Fig. 4. **Exp. Results. (Left) Mission Efficiency:** Scalability is observed from $N = 2$ to $N = 8$. Error bars indicate the 95% confidence interval, showing higher consistency at higher swarm densities. Fault scenarios show temporal degradation compared to nominal $N = 8$ case, confirming the cost of agent loss. **(Center) Workload Distribution (Gini):** Inequality rises during faults as surviving agents absorb extra tasks. **(Right) Bid Density:** Bids per auction saturate at $N = 8$, indicating high market availability, where nearly all agents actively participate in the allocation process.

(307s and 382s, respectively), and is small relative to the mission's operational timescale. This speed is a consequence of the *Liquidation Protocol* defined in Sec. IV. In traditional leader-election algorithms (e.g., Raft), a node failure triggers a timeout and a voting phase, consuming valuable time. In IRDS, failure is treated instantaneously as a market event: the "goods" (sectors) are released, and the "buyers" (neighbors) immediately acquire them because their proximity makes them the lowest bidders.

B. Emergent Economic Dynamics

An analysis of the metrics shows how the virtual economy regulates swarm behavior without explicit control.

1) Market Availability: The *Bids per Auction* metric (Table II) indicates high system connectivity. At $N = 8$, participation saturates at ≈ 7.3 bids per auction (91% of the swarm). This "High Availability" ensures that task allocation is globally optimized rather than locally greedy, as nearly every agent is aware of every opportunity. Unlike limited-range heuristics that suffer from information silos, the auction mechanism maintains global awareness without saturating the decision-making process.

2) The Cost of Resilience (Gini Index): The Gini Coefficient highlights the unavoidable trade-off between fairness and resilience. In the baseline ($N = 8$, No Fault), the Gini index is low (0.10), indicating that the market equalizes travel distance across the fleet. However, under the "Double Fault" scenario, the Gini index rises to 0.27. This inequality represents the calculated *Cost of Resilience*: surviving agents near the failure site must undertake disproportionate work to cover the gap. Crucially, the system stabilizes at $G = 0.27$ rather than diverging to $G = 1.0$ (monopolization). This proves that the budget constraint $W_i(t)$ effectively prevents any single agent from depleting its battery to save the mission, enforcing a soft load-balancing even during crisis management.

C. Scalability vs. Energy Overhead

While the system achieves linear speedup from $N = 2$ ($721s$) to $N = 8$ ($223s$), the *Total Distance* metric reveals a hidden cost. The collective distance traveled increases from $346m$ to $420m$ as the swarm grows. This 21% increase in energy expenditure represents the overhead of coordination and initial deployment (agents traveling from the depot to distant sectors). For safety-critical SAR, this is an acceptable trade-off: the priority is minimizing time-to-rescue, even at the cost of higher energy consumption.

D. Consensus Latency and Physical Deadlocks

While the Dynamic Quorum mechanism handles agent crashes (Health Deadlocks), the current implementation does not enforce a timeout for the voting. This exposes the system to *Motion Deadlocks*, where a healthy verifier becomes trapped by obstacles or other agents and cannot reach its voting position. This phenomenon explains the high variance observed in $N = 8$ baseline ($\sigma = 190s$) and Double Fault ($\sigma = 394s$) scenarios. In crowded environments, the APF repulsion forces can create local minima that trap verifiers, causing the consensus protocol to stall. Future iterations may mitigate this by implementing a T_{max} timeout, causing the system to abort the vote if physical convergence is not achieved within a bounded window.

E. Algorithmic Complexity Analysis

An advantage of IRDS is its computational scalability compared to global optimization methods.

- **Centralized Global Optimization (Hungarian):** Solving the linear assignment problem for N agents and M tasks usually requires $\mathcal{O}(N^3)$ time. While feasible for small teams, this polynomial growth creates a computational bottleneck as the swarm size increases.
- **IRDS Greedy Reverse Auction:** Each agent calculates M bids locally with complexity $\mathcal{O}(M)$. The allocation logic effectively sorts the bids to find the lowest cost. Using a standard efficient sort (e.g., Mergesort), this operation scales as $\mathcal{O}(K \log K)$, where $K = N \times M$ is the total number of bids.

Thus, the IRDS complexity scales quasi-linearly: $\mathcal{O}(NM \log(NM))$. This lower complexity class ensures that the architecture remains viable for large-scale deployments where $\mathcal{O}(N^3)$ methods would become computationally prohibitive.

VII. Conclusion and Future Work

This work presented the *Intelligent Replanning Drone Swarm* (IRDS), a coordination architecture for safety-critical Search and Rescue. By replacing static planning with a dynamic *Reverse-Auction Market*, we transformed the problem of agent failure into a mechanism for economic opportunity, where the loss of a node automatically triggers a localized cost-minimized task reallocation.

The architecture was tested using physics-based simulations ($K = 100$ trials). Results indicate that IRDS is *Fault-Tolerant*: the swarm successfully recovers tasks from failed agents with an average reaction latency of $13.1s$ (single fault). Furthermore, the system maintained a mission success rate of 93% even under 25% agent loss, supporting its viability for fault-prone environments. Scaling the swarm from $N = 2$ to $N = 8$ yielded a $3.2\times$ speedup in time-to-completion at the cost of a modest increase in total energy expenditure due to coordination overhead. Future work includes deployment on Bitcraze Crazyflie drones to verify communication overhead under realistic RF packet loss, extending the market model to heterogeneous swarms with sensor-aware bidding, integrating Bayesian search probability models, and evaluating scalability limits with larger swarms ($N = 16, 32$).

Acknowledgment

This research was funded by the European Union's Horizon Europe research and innovation programme and the Chips Joint Undertaking under Grant Agreement No. 101194287, project NexTArc (Next Generation Open Innovations in Trustworthy Embedded AI Architectures for Smart Cities, Mobility and Logistics).

References

[1] L. Giacomossi, M. R. O. A. Maximo, N. Sundelius, P. Funk, J. F. B. Brancalion, and R. Sohlberg, "Cooperative search and rescue with drone swarm," in *International Congress and Workshop on Industrial AI and eMaintenance 2023*. Cham: Springer Nature Switzerland, 2024, pp. 381–393.

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

[3] J. A. Ricardo, L. Giacomossi, J. F. S. Trentin, J. F. B. Brancalion, M. R. O. A. Maximo, and D. A. Santos, "Cooperative threat engagement using drone swarms," *IEEE Access*, vol. 11, pp. 9529–9546, 2023.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[5] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.

[6] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multi-robot coordination," *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 758–768, 2002.

[7] A. KA and U. Subramaniam, "A systematic literature review on multi-robot task allocation," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–28, 2024.

[8] A. Amirkhani and A. H. Barshooi, "Consensus in multi-agent systems: a review," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 3897–3935, 2022.

[9] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[11] T. M. Cabreira, L. B. Brisolara, and F. J. Paulo R, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019.