# A Blockchain Framework for Equitable and Secure Task Allocation in Robot Swarms

Hanqing Zhao, Alexandre Pacheco, Giovanni Beltrame, *Senior Member, IEEE*, Xue Liu, *Fellow, IEEE*, Marco Dorigo, *Fellow, IEEE*, and Gregory Dudek, *Senior Member, IEEE*

***Abstract*—Recent studies demonstrate the potential of blockchain to enable robots in a swarm to achieve secure consensus about the environment, particularly when robots are homogeneous and perform identical tasks. Typically, robots receive rewards for their contributions to consensus achievement, but no studies have yet targeted heterogeneous swarms, in which the robots have distinct physical capabilities suited to different tasks. We present a novel framework that leverages domain knowledge to decompose the swarm mission into a hierarchy of tasks within smart contracts. This allows the robots to reach a consensus about both the environment and the action plan, allocating tasks among robots with diverse capabilities to improve their performance while maintaining security against faults and malicious behaviors. We refer to this concept as *equitable and secure* task allocation. Validated in Simultaneous Localization and Mapping missions, our approach not only achieves equitable task allocation among robots with varying capabilities, improving mapping accuracy and efficiency, but also shows resilience against malicious attacks.**

***Index Terms*—Swarm robotics, failure detection and recovery, reinforcement learning, robust/adaptive control.**

## I. Introduction

In NATURE, creatures demonstrate remarkable effectiveness in collective missions like foraging and nest building through self-organization [1]. The key to the success of these missions is the ability to allocate different tasks among individuals with varying capabilities, while maintaining resilience to disabled members or attacks from natural enemies [2]. Inspired by these natural behaviors, swarm robotics relies on task allocation for effective mission completion [3], [4]. Robots in the swarm aim to reach consensus on the external environment and the status of other robots. They use this consensus to allocate tasks [5]. Traditional task allocation protocols in swarm robotics, however, assume that robots cooperate consistently and are typically tailored to specific tasks [6], [7]. Consequently, these protocols lack resilience against behaviors caused by faults or malicious tampering (i.e., Byzantine behaviors), unlike their natural counterparts.

The emergence of blockchain and smart contract technologies provides robot swarms with an infrastructure to achieve consensus while defending against Byzantine behaviors [8], [9]. This advancement has evolved from earlier task-specific efforts where robots agree on color ratios of tiles on the floor [10], [11], to a generic blockchain framework suitable for achieving consensus in arbitrary space [12]. These methods are grounded in the context of the Byzantine Generals Problem (BGP), a thought experiment on achieving decentralized consensus [13]. The BGP demonstrates that decentralized consensus becomes impossible, if more than one-third of participants coordinate malicious attacks [14]. Blockchain approaches often use crypto assets as *participation credentials* to control the influence of robots exhibiting Byzantine behaviors (i.e., Byzantine robots), and assume Byzantine robots control no more than one-third of the total assets. Moreover, crypto assets can also serve as a *behavior indicator*, with their changing balances reflecting the nature and severity of robots' Byzantine behaviors [12], [15], [16].

Little attention has been paid, however, to how evolving asset balances can incentivize behaviors that enhance mission efficiency (e.g., encourage robots to undertake tasks suitable for their physical capabilities). For example, while previous work rewards accurate direction estimates in navigation tasks [17], it overlooks role differentiation based on robots' capabilities, such as acting as stationary beacons or guiding peers. This can lead to robots with accurate odometry accumulating assets and bearing most of the workload, while others are sidelined. By enabling every robot to perform tasks that fit its particular strengths, and rewarding each one for its unique contributions, both the efficiency and the long-term resilience of the swarm would be enhanced [6].

A potential solution to this challenge is to integrate domain knowledge [18] regarding different tasks in the swarm mission, into the process that assesses robots' behaviors and redistributes crypto assets. A robot mission often involves hierarchically structured tasks that vary in timescales and levels of abstraction [19]. In swarm robotics, this hierarchical nature is reflected as individual robots perform lower-level tasks that collectively contribute to a high-level mission [20]. For example, a foraging mission contains a search task and a transport (retrieve) task [21], where robots with distinct actuators can collaborate to find and retrieve objects [22]. As illustrated in

Received 29 March 2025; accepted 18 August 2025. Date of publication 4 September 2025; date of current version 12 September 2025. This article was recommended for publication by Associate Editor A. Dutta and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. The work of Hanqing Zhao and Gregory Dudek was supported by NSERC Canadian Robotics Network. The work of Marco Dorigo was supported by Belgian F.R.S.-FNRS. This work was supported in part by Brussels International and in part by Québec MRIF through Bilateral Cooperation Québec–Brussels-Capital Region. *(Corresponding author: Hanqing Zhao.)*

Hanqing Zhao is with the School of Computer Science, McGill University, Montréal, QC H3A 2A7, Canada, and also with the MIST Lab, École Polytechnique de Montréal, Montréal, QC H3T 0A3, Canada (e-mail: hq.zhao@mail.mcgill.ca).

Alexandre Pacheco and Marco Dorigo are with the IRIDIA, Université libre de Bruxelles, B-1050 Brussels, Belgium.

Giovanni Beltrame is with the MIST Lab, École Polytechnique de Montréal, Montréal, QC H3T 1J4, Canada.

Xue Liu and Gregory Dudek are with the School of Computer Science, McGill University, Montréal, QC H3A 2A7, Canada.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3606349, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3606349

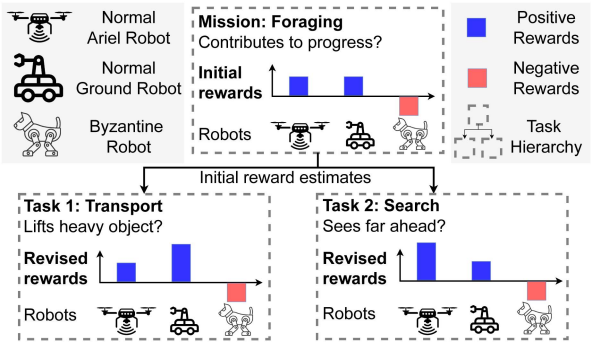


Fig. 1. Domain knowledge of task hierarchy in a foraging mission guides equitable reward estimation for robots engaged in different tasks.

Fig. 1, robots are rewarded according to their contribution within the task hierarchy. Each robot receives an initial reward for participating in the foraging mission—for example, by accurately reporting object locations or efficiently transporting items. The initial reward is then modulated based on task-specific performance: aerial robots receive higher rewards for effective search behavior, while ground robots are better rewarded for successful transport actions. In contrast, a Byzantine robot is assigned a negative initial reward and receives no additional compensation, regardless of the task it chooses to perform.

We further explore the role of crypto asset rewards as behavior incentives, guiding robots to choose tasks that maximize their collective utility. This concept hinges on computational rationality [23] – a foundation for the behavior of artificial intelligence agents. Namely, each non-Byzantine robot is assumed to select actions to maximize its crypto asset reward, within its limited physical capability and information access [24]. We propose a blockchain-deployed smart contract as a meta-controller for the robot swarm. It incorporates task hierarchies, models the environment, evaluates robot performance, and allocates rewards. This approach steers robots toward tasks aligned with their capabilities and mitigates the influence of Byzantine behaviors, ensuring an "equitable and secure" task allocation. We validate our framework in simulated swarm robot Simultaneous Localization and Mapping (SLAM) missions, where robots with noisy odometry engage more in free-space exploration, and those with accurate odometry prefer landmark location estimation. Our framework enhances landmark mapping accuracy and maintains resilience against Byzantine behaviors.

## II. Related Work

*a) Blockchain and smart contract for robot swarms:* Recent success of Decentralized Autonomous Organizations (DAOs) suggests that a framework based on Blockchain [25] and smart contract [26] that involves voting and reward mechanisms, can be used as a Byzantine-tolerant consensus protocol in decentralized and open participation networks [27]. Using crypto assets as participation credentials and behavioral incentives, DAOs can directly influence each participant in decision-making and indirectly regulate their actions, in networks where participants exhibit human rationality [28]. Strobel et al. [10] introduced blockchain smart contracts for achieving secure consensus in robot swarms. Pacheco et al. [11] then demonstrated the application of blockchain in real robots, while Zhao et al. [12] proposed a generic framework to achieve consensus in arbitrary spaces. Pacheco et al. [29] also showed that using smart contracts to assign tasks to robots can improve mission efficiency. These studies used crypto assets to moderate robots' influence on consensus, without exploring the use of crypto assets as behavioral incentives. Our work further studies the use of crypto assets to incentivize robot behavior.

*b) Learning a hierarchy of behaviors:* Behavior learning refers to a range of techniques that enable robots to mimic the actions of living organisms. A behavior can be broken down into a hierarchy of sub-behaviors [30], and learning a behavior typically involves learning its sub-behaviors [31]. There are two main streams of approaches to learning a hierarchy of behaviors: first, classical approaches involve multiple controllers, each representing a sub-behavior. For example, Lotfi et al. [32] combine a spiral search planner with a reactive controller to allow an underwater vehicle to track a diver. Second, advances in machine learning have led to another stream of approaches, where a reward signal that represents multiple behaviors is used to train a single controller that exhibits multiple behaviors in a hierarchy [33]. For instance, Manderson et al. [34] trained an underwater vehicle controller with a reward signal that integrates dive-to-goal and obstacle avoidance behaviors. Zhao et al. [35] developed an off-road vehicle controller, by combining obstacle avoidance and smooth driving behaviors in the reward signal used for training. Our work can be viewed as a combination of these two approaches, where each robot in the swarm learns its own controller from a reward signal that represents multiple behaviors. Advanced behaviors emerge from the collaboration among robots, essentially the interaction of multiple controllers, allowing them to accomplish complex missions.

## III. The Blockchain Framework

We consider a generic consensus achievement process in which robots in a swarm $\mathbb{S}$ are required to perform various tasks to achieve consensus. This is an advancement of a generic framework for consensus achievement [12], in which every robot performs the same task. The robots aim to agree on a *consensus set* $G$ whose elements are from an observation space $\Omega$. This process has three stages:

1) **Proposition**: Robots submit $reports$ to the smart contract. Each report contains a deposit of a fixed portion of the robot's assets and information resulting from the task that the robot performed. Reports that are significantly different from previous ones initiate a $proposal$.
2) **Verification**: Robots perform tasks by referring to the information available on the smart contract (e.g., proposals and consensus set). After task completion, robots submit $reports$ that contain information on whether existing proposals should be part of the consensus set.
3) **Confirmation**: The smart contract confirms/rejects the inclusion of proposals in the consensus set and redistributes crypto assets among robots, according to their contributions in the Proposition and Verification stages.

Three customizable components are involved in these stages:

1) **Clustering algorithm**: It categorizes reports into clusters, the centers of these clusters forming proposals.
2) **Weighted voting game (WVG)**: Conducted within a proposal, its outcomes decide if a proposal's information becomes part of the consensus set.

3) **Reward function**: It redistributes assets based on robots' contribution to the consensus achievement.

In this letter, we extend the framework by incorporating domain knowledge of the *task hierarchy* in the current mission, comprising $T$ different tasks. Upon completing a task, the robot submits a *report* through a structured blockchain transaction to the smart contract, formatted as follows:

*Definition 1:* A *report* is a 4-tuple $\boldsymbol{r} = (\boldsymbol{o}, t, w, x)$, consisting of an observation $\boldsymbol{o} \in \Omega$, a task $t \in [0, T]$, a crypto asset deposit $w$ and the robot's identity $x$ on the blockchain.

The deposit amount $\boldsymbol{r}.w$ of a report is initially a fixed portion of the reporting robot's crypto assets at the time of submission and may later be adjusted by the smart contract. Reports are stored in a tree structure in the smart contract:

*Definition 2:* A *task tree* $\mathbb{T}$ is a set of nodes $\{N_i = (\boldsymbol{L}, N^p, \boldsymbol{N}^c, A^c, R^w),\ i \in [0, T]\}$ that represents the task hierarchy, each task-specific node is a 5-tuple that consists of a report set $N_i.\boldsymbol{L}$, a parent node $N_i.N^p$ and a set of child nodes $N_i.\boldsymbol{N}^c$, a clustering algorithm $N_i.A^c$ and a reward function $N_i.R^w$. Each node $N_i$ aggregates reports from all its child nodes, satisfying $N_i.\boldsymbol{L} = \bigcup_{N_j \in N.\boldsymbol{N}^c} N_j.\boldsymbol{L}$.

The root node $N_0$ represents the collective mission. A leaf node $N_l$ satisfies $N_l.\boldsymbol{N}^c = \emptyset$, representing a low-level task that each robot can choose to perform. For any node $N_i \in \mathbb{T}$, the task-specific clustering algorithm $N_i.A^c$ divides the task report set $N_i.\boldsymbol{L}$ into a set of distinct clusters $\boldsymbol{\mathcal{C}}_i$, satisfying:

$$\bigcup_{C_j \in \boldsymbol{\mathcal{C}}_i} C_j = N_i.\boldsymbol{L} \wedge \forall (C_j, C_k) \in \boldsymbol{\mathcal{C}}_i^2 : C_j \cap C_k = \emptyset \tag{1}$$

at the root node, the clustering algorithm $N_0.A^c$ divides all reports in the mission into a set of clusters $\boldsymbol{\mathcal{C}}_0$, the center of each cluster forms a *proposal*. A root node cluster $C_{0,k} \in \boldsymbol{\mathcal{C}}_0$, $k \in [1, |\boldsymbol{\mathcal{C}}_0|]$ is *sufficiently verified* if it meets the assets super-majority condition: $\sum_{\boldsymbol{r} \in C_{0,k}} \boldsymbol{r}.w \geq \frac{2}{3} q_r T$, where $T$ is the total supply of crypto assets in circulation and $q_r$ is a relative asset quota (in our experiments $q_r = \frac{1}{4}$), which defines the portion of crypto assets a robot must deposit whenever it submits a report, $\lfloor \frac{1}{q_r} \rfloor$ limits the number of open proposals with positive deposits in the smart contract.

Once a root node cluster $C_{0,k}$ is sufficiently verified, a WVG [36] is conducted over all reports in this cluster. The WVG segregates the reports into two distinct coalitions: $\{C_A, C_B\}$ ensuring $C_A \cup C_B = C_{0,k}$ and $C_A \cap C_B = \emptyset$ based on the *voting rule* of the WVG. The information from reports in the winning coalition $C_W \in \{C_A, C_B\}$ is then confirmed and included in the *consensus set*.

*Definition 3:* The *consensus set* $G$ is a set of observations $G \in pow(\Omega)$, that have been verified and confirmed. It is agreed upon by all robots in the swarm and is utilized for decision-making across all nodes in the task tree.

The reward for each robot is assessed at every node along a branch of the task tree, down to the leaf node, which corresponds to the task it performs. At each node $N_i$ representing a specific task $i$, the clustering result $\boldsymbol{\mathcal{C}}_i$ from this node's clustering algorithm $N_i.A^c$ is used as an input for the task-specific reward function $N_i.R^w$, together with the reward from its parent node. This enables a nuanced reward assessment: rewarding robots both for their overall mission contribution and the quality of their task execution.

*Definition 4:* A *task-specific reward function* $N_i.R^w(\boldsymbol{r}, \boldsymbol{\mathcal{C}}_i, N_k.R^w(\boldsymbol{r}), G) \to \mathbb{R}$, of a node $N_i$ on the task tree, takes a report $\boldsymbol{r} \in N_i.\boldsymbol{L}$, the set $\boldsymbol{\mathcal{C}}_i$ of all clusters generated by clustering algorithm $N_i.A^c$, the reward from the parent node's reward function $N_k.R^w(\boldsymbol{r})$ where $N_k = N_i.N^p$ and the consensus set $G$ as input. It determines the crypto reward for the robot based on its performance in task $i$ and all preceding tasks in the hierarchy. The final reward is credited by the leaf node's reward function to the robot's wallet.

Unlike prior work [12], we relax the requirement that clusters must be sufficiently verified before being used as input to the reward function. Instead, we allow robots to earn rewards for creating and revising proposals during the Proposition and Verification stages. We also implement the crypto assets as on-chain tokens [37], this supports complex reward functions, such as negative rewards, and inflation strategy to counter non-responsive attacks.

## IV. Concrete Solution for a SLAM Mission

### A. The Task Tree Hierarchy

We consider a collaborative SLAM mission where robots in a swarm want to explore an area and reach a consensus on a map $G \in pow(\mathbb{R}^2)$ that contains 2D coordinates of various landmarks, even in the presence of Byzantine robots attempting to disrupt the mission (i.e., malicious robots). The mission has two objectives: **1)** To accurately map the landmarks; **2)** To find and map the landmarks as quickly as possible. We divide this mission into two tasks: **1) Landmark localization:** estimate the coordinates of landmarks; and **2) Free space exploration:** explore and locate open areas that are far from landmarks. Unlike free space exploration, the landmark localization task requires higher accuracy in the robots' odometry estimates.

### B. The Reward Functions

As introduced in Section III, the reward for each report is determined through a sequence of reward functions from the root to a leaf node of the task tree. Specifically, a report for the landmark localization task: $\boldsymbol{r} : \boldsymbol{r}.t = 1$, includes a 2D coordinate $\boldsymbol{r}.\boldsymbol{o} \in \mathbb{R}^2$, representing the robot's estimate of the landmark's location. The *mission reward* assigned by the root node's reward function $m_0(\boldsymbol{r}) = N_0.R^w(\boldsymbol{r}, \boldsymbol{\mathcal{C}}_0, \emptyset, G)$, measures the report's contribution to the SLAM mission:

$$m_0(\boldsymbol{r} | \boldsymbol{r}.t = 1) = \begin{cases} \boldsymbol{r}.w + \frac{\sum_{\boldsymbol{r}' \in C_{0,\boldsymbol{r}} \setminus C_W} \boldsymbol{r}'.w}{|C_W|}, & \text{if } \boldsymbol{r} \in C_W \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $C_{0,\boldsymbol{r}} \in \boldsymbol{\mathcal{C}}_0$ is the cluster produced by the root node's clustering algorithm that contains report $\boldsymbol{r}$; and $C_W$ denotes the winning coalition in the WVG on $C_{0,\boldsymbol{r}}$. The WVG splits $C_{0,\boldsymbol{r}}$ into two coalitions by task type: $C_L = \{\boldsymbol{r}' \in C_{0,\boldsymbol{r}} | \boldsymbol{r}'.t = 1\}$ for landmark localization reports, and $C_E = C_{0,\boldsymbol{r}} \setminus C_L$ for free space exploration reports. The WVG uses the following assets superiority voting rule to select the winning coalition: $C_W \in \{C_L, C_E\} : \sum_{\boldsymbol{r} \in C_W} \boldsymbol{r}.w \geq \frac{1}{2} \sum_{\boldsymbol{r} \in C_{0,\boldsymbol{r}}} \boldsymbol{r}.w$.

If $C_{0,\boldsymbol{r}}$ is not sufficiently verified, $C_W$ is an empty set. Once $C_{0,\boldsymbol{r}}$ is sufficiently verified, the mission reward redistributes the deposits from incorrect reports in the losing coalition $C_{0,\boldsymbol{r}} \setminus C_W$, evenly among the robots that submitted reports to the winning coalition $C_W$. A free space exploration report, $\boldsymbol{r} : \boldsymbol{r}.t = 2$ contains a coordinate that the reporter robot believes to be far from any landmarks, helping mark free space to accelerate new landmark discovery and voting to exclude incorrect landmarks

from consensus. The mission reward for such reports follows this structure:

$$m_0(\boldsymbol{r}|\boldsymbol{r}.t=2)=$$
$$\begin{cases}\boldsymbol{r}.w+\frac{1}{|C_W|}\sum_{\boldsymbol{r}'\in C_{0,\boldsymbol{r}}\setminus C_W}\boldsymbol{r}'.w, & \text{if } \boldsymbol{r}\in C_W,\\ \boldsymbol{r}.w, & \text{if } (\forall \boldsymbol{r}'\in C_{0,\boldsymbol{r}}:\boldsymbol{r}'.t=2)\wedge \boldsymbol{r}.w\neq 0,\\ -q_r AllAssets(\boldsymbol{r}.x), & \text{if } C_W\neq\emptyset\wedge\boldsymbol{r}\in C_{0,\boldsymbol{r}}\setminus C_W\wedge\boldsymbol{r}.w=0,\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

$AllAssets(\boldsymbol{r}.x)$ is the sum of the robot's wallet balance and all report deposits submitted by robot $\boldsymbol{r}.x$ in $\boldsymbol{\mathcal{C}}_0$.

In (3), when a free space exploration report is part of the winning coalition $C_W$ of a sufficiently verified cluster, the reporter robot receives its deposit back and shares the deposits from incorrect landmark reports. If the report belongs to a cluster without landmark reports, the deposit is refunded immediately, and the deposit amount is set to zero, though the report remains in the smart contract for reference. If a previously refunded free space report later becomes part of a losing coalition, the reporter robot loses assets for submitting incorrect free space reports near new landmarks.

For landmark localization tasks, the clustering algorithm at node $N_1$ clusters landmark location reports, into a set of distinct clusters $\boldsymbol{\mathcal{C}}_1$. The landmark localization task reward function $m_1(\boldsymbol{r})=N_1.R^w(\boldsymbol{r},\boldsymbol{\mathcal{C}}_1,m_0(\boldsymbol{r}),G)$ adjusts the mission reward $m_0(\boldsymbol{r})$ based on the accuracy of the reported landmark location, as defined in (4):

$$m_1(\boldsymbol{r})=$$
$$\begin{cases}\frac{1-d(\boldsymbol{r}.\boldsymbol{o},ce(C_{1,\boldsymbol{r}}))}{\sum_{\boldsymbol{r}'.\boldsymbol{o}\in C_{1,\boldsymbol{r}}}1-d(\boldsymbol{r}'.\boldsymbol{o},ce(C_{1,\boldsymbol{r}}))}I_L+m_0(\boldsymbol{r}), & \text{if } m_0(\boldsymbol{r})>0,\\ m_0(\boldsymbol{r}), & \text{otherwise.}\end{cases} \tag{4}$$

where $C_{1,\boldsymbol{r}}\in\boldsymbol{\mathcal{C}}_1$ denotes the cluster containing report $r$ and $I_L$ is an inflation constant, set as $I_L=\frac{1}{15}T_0$. The term $T_0$ represents the initial amount of assets in circulation within the swarm. The function $d(\cdot,\cdot)$ represents the normalized Euclidean distance in the observation space $\Omega$, and $ce(\cdot)$ is the center of a cluster. In our experiments, $ce(C_{1,\boldsymbol{r}})$ is the weighted average of all observations within cluster $C_{1,\boldsymbol{r}}$, where the weight equals the deposit in each report:

$$ce(C_{1,\boldsymbol{r}})=\frac{\sum_{\boldsymbol{r}'\in C_{1,\boldsymbol{r}}}\boldsymbol{r}'.\boldsymbol{o}\,\boldsymbol{r}'.w}{\sum_{\boldsymbol{r}'\in C_{1,\boldsymbol{r}}}\boldsymbol{r}'.w}\,. \tag{5}$$

Consider a free space report $\boldsymbol{r}.t=2$ that has received its mission reward $m_0(\boldsymbol{r})$. It will then be evaluated by the reward function of task node $N_2$, as shown in (6):

$$m_2(\boldsymbol{r})=$$
$$\begin{cases}\max(\sigma_G^2(\boldsymbol{r}.\boldsymbol{o}|O_F\setminus\boldsymbol{r}.\boldsymbol{o})\\ -\sigma_G^2(\boldsymbol{r}.\boldsymbol{o}|O_F),0)I_E+m_0(\boldsymbol{r}), & \text{if } m_0(\boldsymbol{r})>0,\\ m_0(\boldsymbol{r}), & \text{otherwise.}\end{cases} \tag{6}$$

in which $I_E$ is an inflation constant set to $I_E=\frac{3}{2|\mathbb{S}|}I_L$, with $|\mathbb{S}|$ being the number of robots in the swarm. The set $O_F=\{\boldsymbol{r}.\boldsymbol{o},\forall\boldsymbol{r}\in N_2.\boldsymbol{L}\}\bigcup G$ consists of all free space locations at node $N_2$ and confirmed landmarks from the consensus set $G$. The term $\sigma_G(\boldsymbol{r}.\boldsymbol{o}|O_F)$ refers to the variance of $\boldsymbol{r}.\boldsymbol{o}$ in the predictive distribution of a Gaussian process regression [38],

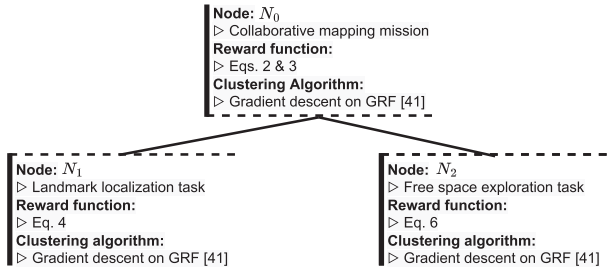


Fig. 2. The task tree for the collaborative SLAM mission.

calculated based on observations in $O_F$. We refer to a Gaussian process within $\Omega=\mathbb{R}^2$ as a Gaussian Random Field (GRF). Each free space report is rewarded according to its reduction in spatial uncertainty (the decrease of local variance in the GRF), encouraging robots to explore new areas. The variance $\sigma_G^2$ is calculated as follows:

$$\sigma_G^2(\boldsymbol{r}.\boldsymbol{o}|O_F)=Cov(\boldsymbol{r}.\boldsymbol{o},\boldsymbol{r}.\boldsymbol{o})-\boldsymbol{k}^T\boldsymbol{C}^{-1}\boldsymbol{k} \tag{7}$$

where $Cov(\cdot,\cdot)$ represents the Radial Basis covariance kernel function [39] used in the GRF:

$$Cov(\boldsymbol{o}_1,\boldsymbol{o}_2)=\exp\left(\frac{-||\boldsymbol{o}_1-\boldsymbol{o}_2||^2}{2l}\right) \tag{8}$$

where $\boldsymbol{k}=[Cov(\boldsymbol{r}.\boldsymbol{o},\boldsymbol{o}'),\boldsymbol{o}'\in O_F]$ is a covariance vector between the newly reported location and previously observed coordinates; $\boldsymbol{C}$ is a covariance matrix; $l$ is a distance scaling factor. Clustering algorithms at each node independently cluster reports using a gradient descent on the GRF. Unlike prior work [12], our method doesn't require predefined cluster numbers. Fig. 2 summarizes the task tree structure.

We implement as detailed in Fig. 3: in Subfigure A, when a robot submits a report $\boldsymbol{r}$, it is first stored in the leaf node that corresponds to the report's task $N_{\boldsymbol{r}.t}$. The report is then copied upward through all ancestor nodes, reaching the root node $N_0$ of which the report set $N_0.\boldsymbol{L}$ includes reports from all of its child nodes. In subfigures B, C & D, at the root node, reports are clustered by a clustering algorithm $N_0.A^c$, sufficiently verified clusters then undergo a WVG, the center locations of the winning coalition is added to the consensus set. The root node calculates mission rewards for all reports based on clustering and WVG outcomes ( (2) & (3)). Finally, the leaf nodes evaluate task rewards for each robot, with the mission rewards as input; this step updates the deposits of unverified reports and robots' wallet balance ( (4) & (6)).

## V. Experimental Studies

### A. Introduction to the Experiment Settings

To evaluate our framework, we simulate collaborative SLAM missions, where robots jointly explore and map multiple landmark locations in the ARGoS simulator [40]. Experiments are run on an Intel NUC 12 (Core i7-1260P, 64 GB RAM, Ubuntu 20.04). Each robot uses a local Extended Kalman Filter SLAM (EKF-SLAM) estimator that tracks the state $\boldsymbol{x}^{EKF}$ contains its own position, orientation, and locations of seen landmarks in the global reference frame:

$$\boldsymbol{x}_{t+1}^{EKF}=f_{EKF}(\boldsymbol{x}_t^{EKF},\boldsymbol{u}_t,\boldsymbol{o}_t^l,\boldsymbol{o}_t^p)\,. \tag{9}$$

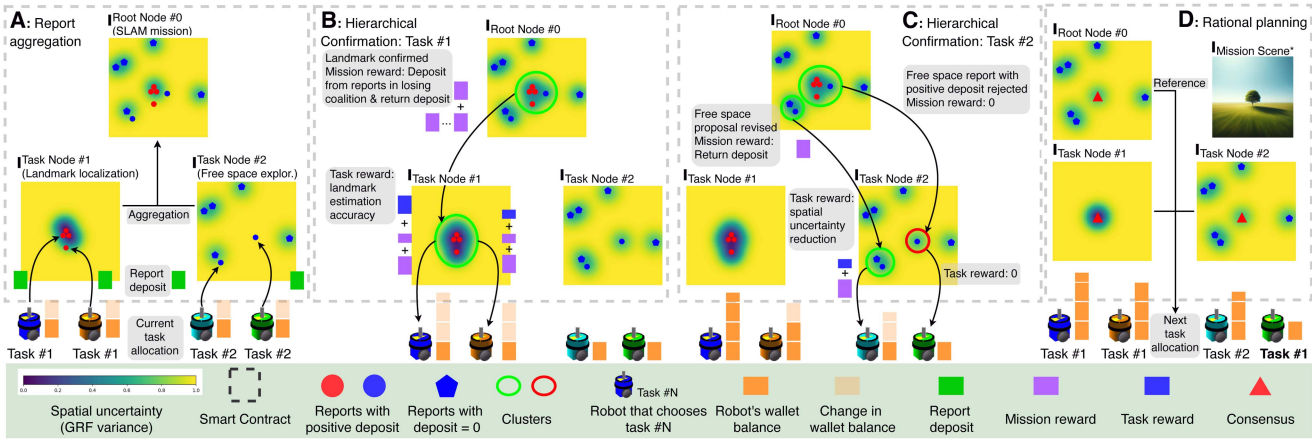


Fig. 3. Illustration of the smart contract meta-controller: Reports submitted by robots to the two task leaf nodes are aggregated in the root node. A clustering algorithm based on gradient descent in the variance field of a GRF categorizes these reports into clusters. The rewards for each report are then determined in a top-down way along the task tree's branches. Each robot that submits a report earns a reward that is credited to its crypto asset balance, coinciding with the generation of new proposals and consensus set. Finally, robots choose their next tasks based on the received rewards.

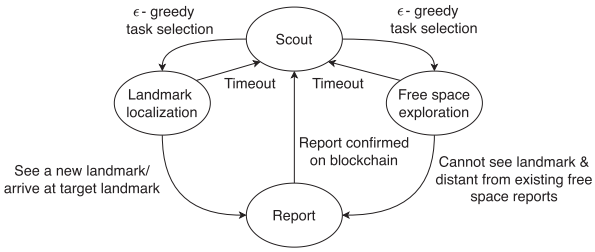


Fig. 4. The finite state machine controller of a robot in the experiment.

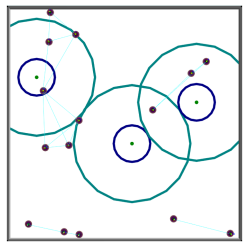

Fig. 5. Small arena for the collaborative SLAM task simulation.

Here, $\boldsymbol{u}_t$ denotes the control input at time step $t$, including the robot's linear and angular velocities, while $\boldsymbol{o}_t^l$ represents the relative positions to the robot of visible landmarks from a noisy virtual *landmark sensor*. Moreover, given our focus on the robot's behavior guided by the smart contract, we mitigate cumulative errors in the robot's local EKF module [41] by integrating the robot's self-position readings $\boldsymbol{o}_t^p$ from a noisy virtual *position sensor* into the EKF.

Each robot operates with a two-arm bandit model, where each arm corresponds to a task. A $Q$-learning estimator estimates the $Q$-values for both tasks [42]. Using an $\epsilon$-greedy policy, robots choose tasks based on the $Q$-values. In our experiment, we set $\epsilon = 0.3$. Precisely, when a robot receives a reward $m(\boldsymbol{r}_a)$ associated with a previously submitted report $\boldsymbol{r}_a$ for task $a$, it updates its $Q$-values estimates accordingly:

$$Q(a) \leftarrow Q(a) + \frac{1}{num_t(a)}(m(\boldsymbol{r}_a) - Q(a)) \qquad (10)$$

where $num_t(a)$ is the number of times the robot has received a reward for reporting on task $a$ up to time step $t$. Each robot runs a four-state Finite State Machine (FSM) controller (Fig. 4): **1) Scout:** The robot performs random walks, monitors new proposals from the smart contract, and updates the $Q$-value based on changes in its crypto assets. It then selects its next task using an $\epsilon$-greedy policy. **2) Landmark Localization:** The robot navigates to proposed landmarks using odometry estimates, or searches for new landmarks via a stochastic gradient ascent walk within a GRF built from free-space reports (i.e., navigation GRF), similar to existing strategies that use GRFs to model spatial uncertainty and support planning [43], [44]. Upon discovering a landmark, it submits its estimated location via a blockchain transaction. **3) Free Space Exploration:** The robot performs a stochastic gradient ascent walk within its navigation GRF, submitting free-space reports with its current location estimates when sufficiently far from known landmarks. **4) Report:** The robot performs a random walk until its previous transaction is included in or discarded from the blockchain.

Each robot operates a blockchain virtual machine with the Proof-of-Authority protocol [45], on which a smart contract is deployed that contains a task tree as detailed in Fig. 2.

### B. Experiments in Small Arena

We first study a simulated square-shaped arena, with three randomly placed landmarks and a swarm of robots starting at random positions. The landmarks, marked as green circles in Fig. 5, are surrounded by cyan circles denoting their *visible ranges* and smaller blue circles indicating their *scrutable ranges*. Within the visible range, a robot can perceive the landmark's relative location. As it moves closer into the scrutable range, it may obtain more accurate estimates of the landmark's position, depending on its sensing capabilities.

Each robot uses the EKF-SLAM model for odometry estimation (9) and is affected by white noise from uniform distributions in all its sensors. The swarm comprises two types of robots:

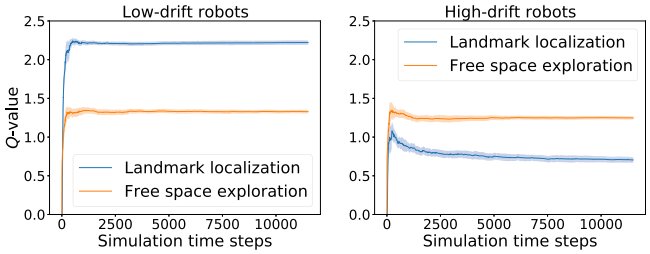


Fig. 6. $Q$-value estimates for the two tasks of two different types of robots. In a swarm of 15 robots over 20 repetitions in a 3 landmark small arena, the shaded area represents 90% confidence interval.

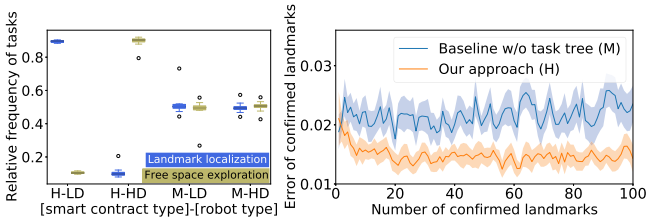


Fig. 7. Left: Relative frequency of task selection for a swarm of 15 robots over 20 repetitions. Right: Consensus error of the landmark locations in Euclidean distance. The box plots are labeled according to the format [smart contract type]-[robot type], where: [smart contract type] $\in$ {H: our approach, M: baseline approach} and [robot type] $\in$ {LD: low-drift, HD: high-drift}. The time-series data is smoothed by a Savitzky-Golay filter with a window-length of 3, the shaded area represents 90% confidence interval.

*high-drift* and *low-drift*, with different sensor noise levels. Low-drift robots experience 90% less noise in position sensors and 50% less noise noise in landmark sensors compared to high-drift robots in the visible range. Additionally, their landmark sensor noise is reduced by another 90% when within the scrutable range of a landmark.

In our setup, computationally rational robots strive to position themselves within a landmark's scrutable range based on self-position estimates before reporting its location. Moreover, in this small arena, confirmed landmarks are promptly removed from the consensus, allowing the robots to continue exploring unseen areas. In a swarm of 10 low-drift and 5 high-drift robots. As shown in Fig. 6, low-drift robots learned higher $Q$-values for the landmark localization task, due to their more accurate reports, earning greater rewards under (4). Moreover, both types of robots received similar rewards for the less demanding free space exploration task.

Different $Q$-values influence the robots' task selection, as shown in the left part of Fig. 7. Unlike the baseline smart contract proposed in [12], which rewards robots solely based on mission performance (Eq. (2)), our approach promotes equitable task allocation: low-drift robots select the landmark localization task in 89.5% of cases, while high-drift robots choose the free-space exploration task 89.6% of cases. As illustrated in the right part of Fig. 7, this targeted role assignment reduces the average landmark localization error — measured after the $10^{th}$ confirmed landmark — by 32.7%.

To study the role of crypto assets as participation credentials to mitigate Byzantine behaviors, we consider a swarm of 10 low-drift robots, 4 high-drift robots, and 1 malicious robot. The malicious robot is a low-drift robot that consistently submits

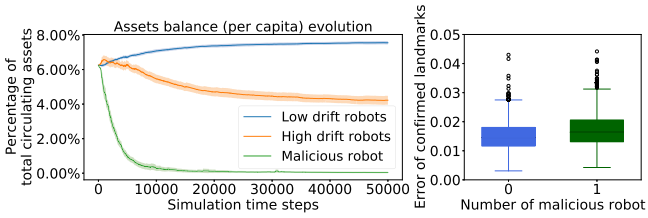


Fig. 8. In the small arena with three landmarks: Left: Evolution of crypto asset balance in a swarm with 1 malicious robot over 20 repetitions, the shaded area representing the 90% confidence interval. Right: Consensus error of 2000 confirmed landmarks locations in 20 repetitions in Euclidean distance, of two swarms – with and without the malicious robot.

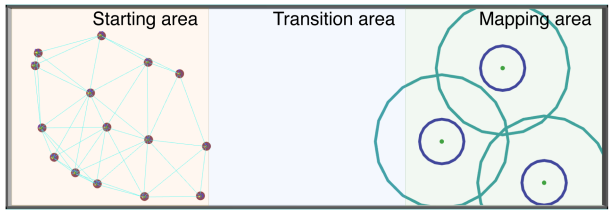


Fig. 9. Large arena for the collaborative SLAM task simulation, where 15 robots are randomly distributed in the leftmost starting area.

incorrect reports (e.g., misclassifying landmarks as free space or vice versa): it could potentially introduce errors into the consensus set and disrupt task allocation. The left part of Fig. 8 shows that the mission reward supersedes the task reward, quickly depleting the malicious robot's assets, thus reducing its influence. Moreover, as shown on the right side of Fig. 8, despite the presence of the malicious robot, the consensus error in landmark locations remains similar to a swarm without it, indicating that our framework minimizes the malicious robot impact and maintains secure consensus and task allocation via asset redistribution.

### C. Experiments in Large Arena

We evaluate our framework in a larger arena (Fig. 9) with three equal-sized areas: robots start randomly in the leftmost starting area, navigate through a transition area in the middle, and map three randomly placed landmarks in the rightmost area. Once a landmark location is added to the consensus set, all robots freeze their location in their local EKF-SLAM state. Non-malicious robots inherit $Q$-values from the earlier experiment, preferring the task that suits their abilities.

Robots attempt to efficiently find new landmarks via a stochastic gradient ascent walk with the local navigation GRF, as detailed in section V-A. Fig. 10 shows that unlike the GRFs for clustering in smart contracts, the navigation GRFs of individual robots feature a larger distance scaling factor $l$ in the covariance function (Eq. (8)).

To quantify performance, we use two performance measures: **1) Time to landmarks' discovery**: The time step when all three landmarks have their first reports stored in the smart contract; **2) Time to landmarks' confirmation**: The time step when all three landmarks have been verified and added to the consensus set. With the local navigation GRF, the average time required for landmark discovery and confirmation is reduced by 31.3% and 23.2%, respectively, saving a significant amount of time

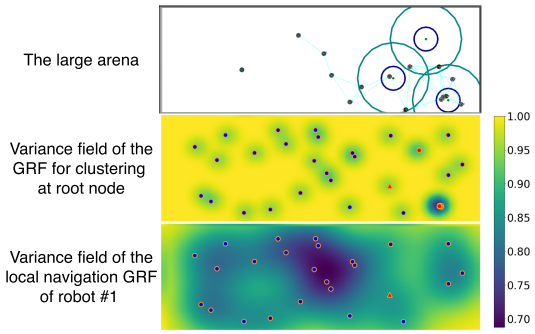


Fig. 10. Top: large arena with 15 robots and 3 landmarks; Middle: variance field of the GRF for clustering at the root node in the smart contract; Bottom: variance field of the local navigation GRF of a specific robot. Red and blue points represent landmark localization and free space exploration reports, respectively; red triangles are confirmed landmark locations.

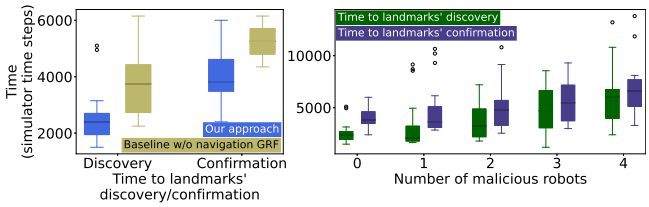


Fig. 11. Left: Time to landmark discovery and confirmation with our approach and the baseline approach without navigation GRF, in a swarm of 15 non-malicious robots. Right: Evolution of time to landmark discovery and confirmation of our approach with increasing number of malicious robots.

compared to random walk (Fig. 11, left). However, its reliance on unconfirmed free space reports makes the swarm vulnerable to malicious robots misreporting free space as landmarks, delaying both landmarks' discovery and confirmation. Since the GRF calculation uses unconfirmed free space reports, malicious robots misreporting free space as landmarks can delay landmarks' discovery and confirmation (Fig. 11, right). Requiring robots to verify all free space proposals could mitigate these attacks but would drastically reduce efficiency due to the high volume of reports awaiting confirmation.

This vulnerability to malicious attacks on mission efficiency, highlights the efficiency-security trade-off — a crucial factor in consensus achievement [46]. The task tree hierarchy enables fine-tuning of this trade-off through reward design. In our experiment, we design free space task rewards to prioritize efficiency in map exploration by discouraging verification of free space proposals, at the cost of some security in free space localization which affects mission efficiency (6). Conversely, we emphasize security in landmark localization by only rewarding accurate reports that contribute to the confirmation of a landmark (2) and penalizing inaccurate reports (4). This design encourages robots with low odometry drift to verify landmark proposals while penalizing malicious robots for false landmark reports.

To study resilience against malicious behavior, we build a baseline without blockchain that mirrors traditional decentralized SLAMs, where robots merge maps from peers and use local outlier detection to maintain mapping accuracy [47], precisely, robots use the weighted-mean-subsequence reduced (W-MSR) protocol [48] to update landmark beliefs by discarding the five beliefs that are furthest from their own; they perform both

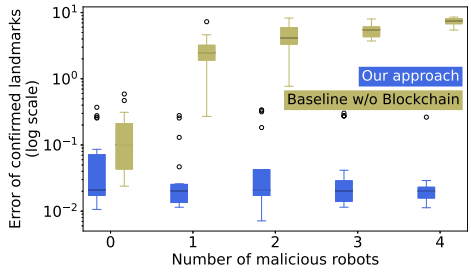


Fig. 12. Evolution of error in confirmed landmark locations in a swarm of 15 robots, comprising 10 low-drift robots, and 0 to 4 malicious robots (5 to 1 high-drift robots). Comparison between our approach and a baseline without blockchain, conducted in the large arena across 20 repetitions.

landmark localization and free space exploration tasks with equal probability. Fig. 12 shows that our approach achieves lower landmark location errors than the baseline in the absence of malicious robots thanks to equitable task allocation. As the number of malicious robots increases, our method maintains low error, showing resilience against attacks on mapping accuracy.

## VI. Conclusion

We presented a blockchain and smart contract-based framework for Byzantine-tolerant consensus in multi-robot systems, using crypto assets as incentives for computationally rational robots to achieve equitable task allocation. By embedding mission-specific domain knowledge into a task tree within the smart contract, our framework controls the information in the consensus and hierarchically assesses rewards to guide robots toward tasks suited to their capabilities.

This framework is validated in collaborative SLAM missions involving robots with different sensing capabilities. Results show that our approach increases each robot's utility to improve the overall mission quality. Further experiments in complex environments with malicious robots confirm our approach's resilience to Byzantine behaviors, while the task tree hierarchy enables a nuanced efficiency–security trade-off adjustment, opening avenues for future investigation.

In real-world swarm robotics missions such as SLAM [47] or persistent surveillance [49], robots must reach consensus on various types of information (e.g., shared maps or energy levels), despite differing connectivity conditions and energy constraints. The time and energy required to maintain real-time consensus under such limitations also depends on the underlying blockchain protocol. In this letter, we adopt Proof-of-Authority as a representative example; however, exploring alternative consensus mechanisms better suited to the specific constraints of different missions remains an important direction for future research.

## References

[1] D. Sumpter, "The principles of collective animal behaviour," *Philos. Trans. Roy. Soc. B: Biol. Sci.*, vol. 361, no. 1465, pp. 5–22, 2006.
[2] E. Robinson, O. Feinerman, and N. Franks, "Flexible task allocation and the organization of work in ants," *Proc. Roy. Soc. B: Biol. Sci.*, vol. 276, no. 1677, pp. 4373–4380, 2009.
[3] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Sci. Robot.*, vol. 5, no. 49, 2020, Art. no. abe4385.

[4] G. Dudek and M. Jenkin, "Robot collectives," in *Computational Principles of Mobile Robotics*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2024, ch. 11, pp. 301–315.

[5] R. Liu et al., "Trust-aware behavior reflection for robot swarm self-healing," in *Proc. Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 122–130.

[6] E. Debie, K. Kasmarik, and M. Garratt, "Swarm robotics: A survey from a multi-tasking perspective," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–38, 2023.

[7] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Auton. Robots*, vol. 3, pp. 375–397, 1996.

[8] A. Reina, "Robot teams stay safe with blockchains," *Nature Mach. Intell.*, vol. 2, pp. 240–241, 2020.

[9] E. C. Ferrer, T. Hardjono, A. Pentland, and M. Dorigo, "Secure and secret cooperation in robot swarms," *Sci. Robot.*, vol. 6, no. 56, 2021, Art. no. eabf1538.

[10] V. Strobel, E. C. Ferrer, and M. Dorigo, "Blockchain technology secures robot swarms: A comparison of consensus protocols and their resilience to Byzantine robots," *Front. Robot. AI*, vol. 7, 2020, Art. no. 54.

[11] A. Pacheco, V. Strobel, and M. Dorigo, "A blockchain-controlled physical robot swarm communicating via an ad-hoc network," in *Proc. Int. Conf. Swarm Intell.*, 2020, pp. 3–15.

[12] H. Zhao et al., "A generic framework for byzantine-tolerant consensus achievement in robot swarms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 8839–8846.

[13] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[14] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, 1980.

[15] V. Strobel, A. Pacheco, and M. Dorigo, "Robot swarms neutralize harmful byzantine robots using a blockchain-based token economy," *Sci. Robot.*, vol. 8, no. 79, 2023, Art. no. eabm4636.

[16] E. C. Ferrer, E. Jiménez, J. Lopez-Presa, and J. Martıń-Rueda, "Following leaders in Byzantine multirobot systems by using blockchain technology," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1101–1117, Apr. 2022.

[17] L. Van Calck, A. Pacheco, V. Strobel, M. Dorigo, and A. Reina, "A blockchain-based information market to incentivise cooperation in swarms of self-interested robots," *Nature Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 20417.

[18] S. Anand, D. Bell, and J. Hughes, "The role of domain knowledge in data mining," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 1995, pp. 37–43.

[19] R. Arkin, *Behavior-Based Robotics*. Cambridge, MA, USA: MIT Press, 1998.

[20] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present and future," *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.

[21] T. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Trans. Auton. Adaptive Syst.*, vol. 1, no. 1, pp. 4–25, 2006.

[22] M. Dorigo et al., "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robot. Automat. Mag.*, vol. 20, no. 4, pp. 60–71, Dec. 2013.

[23] R. Lewis, A. Howes, and S. Singh, "Computational rationality: Linking mechanism and behavior through bounded utility maximization," *Topics Cogn. Sci.*, vol. 6, no. 2, pp. 279–311, 2014.

[24] H. A. Simon, "A behavioral model of rational choice," *Quart. J. Econ.*, vol. 69, no. 1, pp. 99–118, 1955.

[25] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[26] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Found., Tech. Rep. c74b55f, 2024. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[27] Y. El Faqir, J. Arroyo, and S. Hassan, "An overview of decentralized autonomous organizations on the blockchain," in *Proc. Int. Symp. Open Collaboration*, 2020, pp. 1–8.

[28] W. Mitchell, "The rationality of economic activity," *J. Political Economy*, vol. 18, no. 3, pp. 197–216, 1910.

[29] A. Pacheco, V. Strobel, A. Reina, and M. Dorigo, "Real-time coordination of a foraging robot swarm using blockchain smart contracts," in *Proc. Int. Conf. Swarm Intell.*, Springer, 2022, pp. 196–208.

[30] F. Ratnieks and C. Anderson, "Task partitioning in insect societies," *Insectes Sociaux*, vol. 46, pp. 95–108, 1999.

[31] F. Kirchner, "Q-learning of complex behaviours on a six-legged walking machine," *Robot. Auton. Syst.*, vol. 25, no. 3/4, pp. 253–262, 1998.

[32] F. Lotfi, K. Virji, and G. Dudek, "Robust scuba diver tracking and recovery in open water using YOLOv7, SORT, and spiral search," in *Proc. Conf. Robots Vis.*, 2023, pp. 233–240.

[33] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artif. Intell.*, vol. 299, 2021, Art. no. 103535.

[34] T. Manderson et al., "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles," in *Proc. Robot.: Sci. Syst.*, Corvalis, Oregon, USA, 2020.

[35] H. Zhao, T. Manderson, H. Zhang, X. Liu, and G. Dudek, "Behaviour learning with adaptive motif discovery and interacting multiple model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 10788–10794.

[36] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, *Handbook of Computational Social Choice*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[37] M. Shirole, M. Darisi, and S. Bhirud, "Cryptocurrency token: An overview," in *Proc. Int. Conf. Blockchain Technol.*, Springer, 2019, pp. 133–140.

[38] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[39] J.-P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods Comput. Biol.*, vol. 47, pp. 35–70, 2004.

[40] C. Pinciroli et al., "ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 5027–5034.

[41] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 3562–3568.

[42] M. Duff, "Q-learning for bandit problems," in *Machine Learning Proceedings*. New York, NY, USA: Elsevier, 1995, pp. 209–217.

[43] M. Ali, H. Jardali, N. Roy, and L. Liu, "Autonomous navigation, mapping and exploration with Gaussian processes," in *Proc. Robot.: Sci. Syst.*, Daegu, Republic of Korea, 2023, Art. no. 104.

[44] S. Caccamo, Y. Bekiroglu, C. Ek, and D. Kragic, "Active exploration using Gaussian random fields and Gaussian process implicit surfaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 582–589.

[45] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. Ital. Conf. Cybersecurity*, 2018, Art. no. 06. [Online]. Available: https://ceur-ws.org/Vol-2058/paper-06.pdf

[46] G. A. Rebello, G. Camilo, L. Guimaraes, L. A. de Souza, G. Thomaz, and O. C. Duarte, "A security and performance analysis of proof-based consensus protocols," *Ann. Telecommun.*, vol. 77, no. 7, pp. 517–537, 2021.

[47] P.-Y. Lajoie and G. Beltrame, "Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 475–482, Jan. 2024.

[48] H. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[49] H. Fouad and G. Beltrame, "Energy autonomy for robot systems with constrained resources," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3675–3693, Dec. 2022.