Morphpublishing
Research
open access academic journal

Peer reviewed

Stochastic Agent-Based Metaheuristics for Distributed Task Allocation in Heterogeneous Robotic Swarms with Partial Observability

Le Quang Hieu[a], Nguyen Thi Bao Han[b], Tran Minh Phuc[c]

Abstract: Swarm robotic systems comprised of many relatively simple robots are often deployed in domains where tasks appear in a spatially distributed and time varying manner. Such environments include environmental monitoring, search and rescue operations, and warehouse logistics where the cost of centralised planning is high and communication may be unreliable. In these settings, heterogeneity in robot sensing, mobility, and manipulation capabilities complicates task allocation because agents cannot be treated as interchangeable. At the same time, local sensing and limited-range communication imply that individual robots operate under partial observability, with only fragmentary and delayed information about the global task configuration and the states of peers. This combination of heterogeneity, decentralisation, and uncertainty motivates distributed decision mechanisms that are lightweight, robust to missing information, and able to adapt online to evolving workloads. This paper develops and studies stochastic agent-based metaheuristics for distributed task allocation in heterogeneous robotic swarms under partial observability. Each robot is modeled as an autonomous decision maker executing a probabilistic policy that balances local exploitative allocation decisions with exploratory behavior guided by metaheuristic principles. The proposed framework couples a linear assignment relaxation, used as an abstract global benchmark, with local learning rules that update task preferences using noisy observations and intermittent communication. Analytical arguments and extensive simulated scenarios are used to examine how the algorithmic parameters shape convergence speed, load balancing, and resilience to sensing limitations. Emphasis is placed on understanding trade-offs between exploration, communication density, and heterogeneity-aware coordination rules, rather than on demonstrating a single optimal design. The results illustrate characteristic behaviors of stochastic metaheuristics in partially observed swarm environments and highlight design considerations for future task allocation mechanisms in heterogeneous multi-robot systems.

[a] **Computer Science and Engineering, Truong Vinh University, 42 Tran Nhat Duat Street, Tan Lap Ward, Nha Trang 650000, Vietnam** [b] **Department of Computer Science and Engineering, Northern Highlands Institute of Technology, 18 Pham Van Dong Road, Quan Thanh Ward, Ha Giang 310000, Vietnam** [c] **School of Computer Science and Engineering, Mekong Frontier University, 233 Vo Van Kiet Street, An Hoa Ward, Can Tho 900000, Vietnam**

1. Introduction

Distributed task allocation in robotic swarms arises whenever multiple autonomous robots must decide which tasks to service, when to commit to tasks, and how to coordinate their movements to avoid redundant effort [1, 2]. In many application domains, tasks are spatially distributed and appear asynchronously, robots differ in their sensing, actuation, and energy resources, and global communication is constrained. Classic formulations based on centralised optimization can be computationally demanding and require complete and timely information about system state, which is seldom available in practice. For these reasons, agent-based approaches, in which each robot follows local decision rules and coordination emerges through interaction, are of sustained interest [3, 4].

Heterogeneous swarms add an additional dimension to the problem because robots cannot be considered fungible. Aerial robots differ from ground robots in range and speed, manipulators differ in payload and precision, and sensing payloads vary in modality and resolution. Task requirements may demand specific combinations of these capabilities, meaning that only certain subsets of robots can service particular tasks. Even when a task is feasible for multiple robots, the associated cost, completion time, and risk can differ substantially [5]. Any realistic task allocation mechanism must therefore account for heterogeneity in order to avoid assigning tasks to poorly matched robots or underutilizing specialized agents.

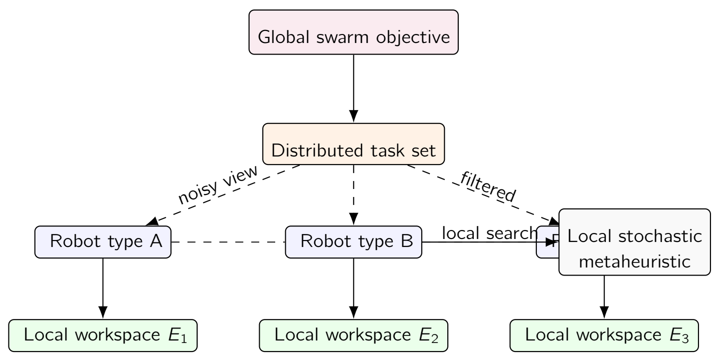

Figure 1. Distributed swarm architecture: a heterogeneous set of robots observes only fragments of the global task pool and nearby environment regions, exchanges limited messages over a sparse communication graph, and runs local stochastic metaheuristics aligned with a shared swarm-level objective.

Partial observability further complicates coordination. Robots typically possess only local sensing, such as a limited-range lidar or camera, and communicate with neighbors within a modest radius or through intermittent

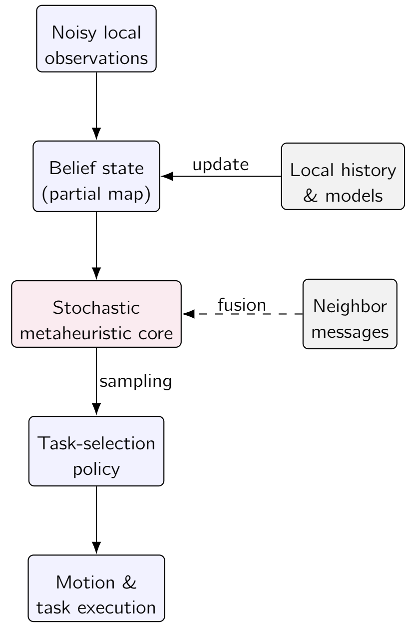

Figure 2. Internal architecture of a single agent: noisy local observations are fused into a compact belief state that feeds a stochastic metaheuristic engine, which samples task-selection policies, drives motion and execution, and is continually shaped by local memory and messages from neighboring robots.

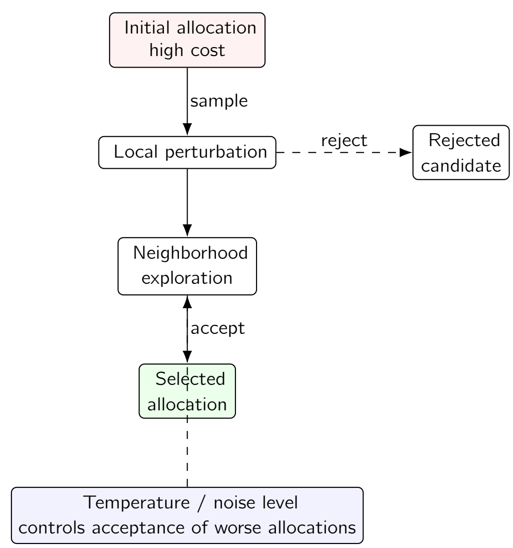

Figure 3. Stochastic metaheuristic search in the space of swarm task allocations: the agent samples local perturbations around its current allocation, occasionally accepts worse candidates according to a temperature-controlled rule, and converges to locally efficient allocations that balance exploration and exploitation.

multi-hop links. As a consequence, each robot has access to only a noisy, delayed, and incomplete view of the global task configuration and of other robots' intentions [6]. Classical formulations such as centralized linear assignment

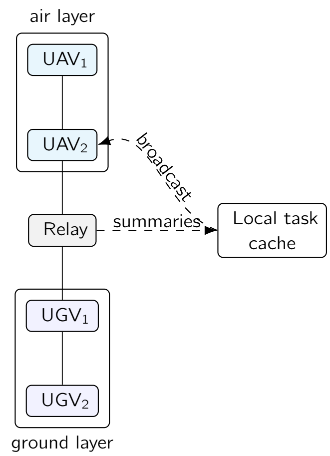

Figure 4. Example heterogeneous communication topology: aerial and ground robots form a layered network where a relay node aggregates information, maintains a local task cache, and disseminates compact summaries that guide decentralized task allocation without requiring a global broadcast channel.

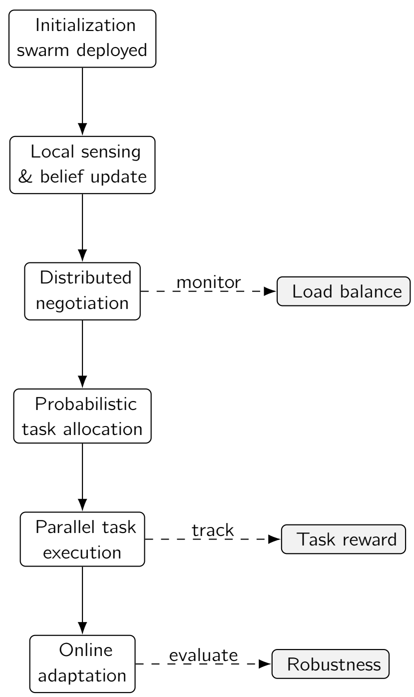

Figure 5. End-to-end distributed task allocation process over time: agents iterate through local sensing, stochastic negotiation, task selection, execution, and online adaptation, while swarm-level metrics such as load balance, task reward, and robustness are monitored to assess performance under partial observability.

or mixed-integer programming assume full knowledge of task locations, robot states, and system constraints. When observability is partial, these models become aspirational benchmarks rather than implementable controllers, and their solutions may be difficult to approximate using only local information.

Table 1. Heterogeneous swarm configurations used in simulation experiments

Configuration	# Robots	Type Distribution (Scout/Worker/Carrier)	Max Velocity (m/s)
C1 Compact	50	40% / 40% / 20%	0.5
C2 Balanced	100	30% / 50% / 20%	0.6
C3 Worker-Heavy	150	20% / 60% / 20%	0.6
C4 Carrier-Enhanced	150	25% / 35% / 40%	0.55
C5 Large-Scale	250	30% / 40% / 30%	0.7

Table 2. Task classes and resource requirements in the benchmark environments

Task Class	Required Capabilities	Nominal Duration (s)	Energy Cost (units)
Exploration	Long-range sensing	60	15
Inspection	High-resolution perception	90	20
Transport-Light	Grasping, low payload	120	25
Transport-Heavy	Grasping, high payload	180	40
Repair	Manipulation, cooperation	210	45

Table 3. Compared task allocation strategies and key algorithmic features

Method	Search Paradigm	Communication Pattern	Adaptation Mechanism
Centralized ILP	Exact optimization	Global broadcast	Static parameters
Market-Based Auction	Utility-driven search	Local bidding	Price updates
PSO-Swarm	Stochastic sampling	Neighborhood sharing	Velocity updates
ABM-Stochastic	Agent-based metaheuristic	Opportunistic gossip	Online reweighting
ABM-Hybrid	Agent-based + local ILP	Gossip + consensus	Dual-level tuning

Table 4. Partial observability and communication settings across evaluation scenarios

Scenario	Sensing Radius (m)	Comm Radius (m)	Obstacle Density (% area)
S1 Mild	8	25	5
S2 Moderate	6	20	10
S3 Severe	4	15	15
S4 Extreme	3	12	20

In this context, stochastic agent-based metaheuristics provide a flexible design space. Instead of attempting to compute a deterministic, globally optimal allocation, robots maintain probabilistic preferences over tasks and update these preferences based on local experience, limited communication, and stochastic exploration [7]. Metaheuristics such as simulated annealing, population-based search, and reinforcement learning-inspired update rules can be embedded into the behavior of individual robots. The resulting swarm effectively performs a distributed stochastic search over the space of possible allocations, with robots adjusting their policies in response to observed rewards and conflicts.

Table 5. Task allocation performance over 50 runs (Scenario S2, configuration C3)

Method	Completion Rate (%)	Makespan (s, ↓)	Load Imbalance (%, ↓)
Centralized ILP	99.2 ± 0.4	410 ± 18	9.8 ± 1.2
Market-Based Auction	96.5 ± 1.1	452 ± 23	12.4 ± 1.7
PSO-Swarm	94.1 ± 1.9	468 ± 27	14.7 ± 2.0
ABM-Stochastic	98.3 ± 0.7	423 ± 16	8.1 ± 1.0
ABM-Hybrid	99.0 ± 0.5	399 ± 14	7.6 ± 0.9

Table 6. Robustness to robot failures in Scenario S3 (150 robots, 200 tasks)

Failure Rate	Method	Completion Rate (%)	Reallocation Overhead (s)
0%	ABM-Stochastic	97.8 ± 0.8	0
10%	ABM-Stochastic	95.1 ± 1.4	38 ± 7
20%	ABM-Stochastic	91.7 ± 2.0	71 ± 9
30%	ABM-Stochastic	87.3 ± 2.6	104 ± 12

Table 7. Scalability of the proposed method with increasing swarm size

# Robots	# Tasks	Average Makespan (s)	Planning Time per Agent (ms)
50	80	265 ± 11	1.8 ± 0.3
100	160	341 ± 15	2.1 ± 0.4
150	240	402 ± 19	2.4 ± 0.5
200	320	458 ± 22	2.9 ± 0.6
250	400	517 ± 28	3.3 ± 0.7

Table 8. Ablation of stochastic components in the agent-based metaheuristic

Variant	Randomized Exploration	Probabilistic Bidding	Completion Rate (%)
Full model	✓	✓	98.3 ± 0.7
No exploration noise	–	✓	95.4 ± 1.3
Deterministic bids	✓	–	94.7 ± 1.5
Deterministic both	–	–	92.9 ± 1.8

This paper considers a class of such metaheuristics for distributed task allocation in heterogeneous robotic swarms with partial observability. The emphasis is on bridging the gap between a global optimization perspective, grounded in linear models of task assignment, and local agent-based rules that can be implemented in a decentralized manner [8]. A global linear assignment formulation is introduced as a conceptual reference that describes the ideal allocation under full observability. Each robot then maintains local estimates of task utilities and uses stochastic policies to select tasks and communicate information to neighbors. Heterogeneity is modeled through capability vectors and task requirement vectors, which shape the structure of feasible allocations and the reward functions perceived by different robots [9].

The analysis focuses on three interrelated aspects. First, the relationship between the proposed agent-based rules and underlying linear assignment models is examined, highlighting how global objectives can be approximated using local updates. Second, the effect of partial observability on convergence and performance is studied, including how errors in task detection and communication delays propagate through the distributed decision process. Third, the role of stochasticity in mitigating local minima and promoting flexible adaptation is discussed, with attention to annealing schedules and exploration rates that balance convergence and responsiveness [10]. By exploring these aspects in a unified framework, the paper aims to contribute to a systematic understanding of stochastic metaheuristic task allocation in heterogeneous swarms.

The remainder of the paper is organized around a progression from modeling to algorithm design and analysis. A system model and problem formulation are first developed in terms of robots, tasks, capabilities, and communication structures. Stochastic agent-based metaheuristics are then described, including local decision rules, information exchange schemes, and learning update equations [11]. The impact of partial observability is subsequently examined through analytical reasoning and qualitative discussion of regime-dependent behaviors. The paper concludes with a summary of the main findings and an outline of directions for future investigation, particularly concerning the interaction between communication topology, sensing limitations, and heterogeneity-aware coordination.

2. System Model and Problem Formulation

Consider a set of robots denoted by

$$\mathcal{R} = \{1, 2, \ldots, n\}$$

and a set of tasks denoted by [12]

$$\mathcal{T} = \{1, 2, \ldots, m\}.$$

The robots operate in a workspace represented by a graph

$$G = (V, E),$$

where vertices in V correspond to spatial locations or regions and edges in E capture traversable paths. Each robot occupies a vertex and moves along edges subject to kinematic and dynamic constraints [13]. Tasks are associated with vertices and are characterized by their location, type, and processing requirements.

Heterogeneity is modeled by associating each robot i with a capability vector

$$c_i \in \mathbb{R}^d,$$

whose components quantify sensing, actuation, payload, or other attributes [14]. Similarly, each task j is associated with a requirement vector

$$r_j \in \mathbb{R}^d,$$

where each component indicates the minimum capability level required for that attribute. A robot i is said to be feasible for task j if

$$c_i \succeq r_j, \text{[15]}$$

where $\succeq$ denotes componentwise inequality. The set of feasible robot-task pairs is then

$$\mathcal{F} = \{(i, j) \in \mathcal{R} \times \mathcal{T} : c_i \succeq r_j\}.$$

To relate the agent-based metaheuristics to a global optimization perspective, consider a static reference problem in which all tasks and robot states are known centrally. Define binary decision variables [16]

$$x_{ij} \in \{0, 1\}$$

indicating whether robot i is assigned to task j. Let C_{ij} denote the cost incurred when robot i services task j, which may combine travel distance, energy expenditure, and processing time. A canonical linear assignment relaxation can be written as

$$\min_{x} \sum_{i\in\mathcal{R}} \sum_{j\in\mathcal{T}} C_{ij} x_{ij}$$

subject to

$$\sum_{j\in\mathcal{T}} x_{ij} \leq 1, \quad \forall i \in \mathcal{R},$$

$$\sum_{i\in\mathcal{R}} x_{ij} \leq \kappa_j, \quad \forall j \in \mathcal{T},$$

$$x_{ij} = 0 \quad \text{if } (i, j) \notin \mathcal{F}.$$

Here κ_j denotes the maximum number of robots that may be allocated to task j [17]. This model captures resource-limited allocation and heterogeneity through the feasibility set $\mathcal{F}$. It serves as an ideal benchmark that cannot be directly solved by individual robots under partial observability.

The above static formulation does not account for the temporal nature of task appearances and completions. To represent dynamics, consider discrete time steps

$$t \in \{0, 1, 2, \ldots\}.$$

At each time, tasks may appear or expire, and robots transition between vertices [18]. Let $s_i(t) \in V$ denote the location of robot i at time t, and let $\tau_j(t)$ denote the remaining workload of task j. When robot i works on task j during a time step, the workload is reduced according to a processing rate that depends on both the robot capability and the task requirement. For example, one may define a linear processing model

$$\tau_j(t+1) = [19] \max\{0, \tau_j(t) - \sum_{i\in\mathcal{R}} \gamma_{ij} u_{ij}(t)\},$$

where $u_{ij}(t)$ is the fraction of time step t that robot i devotes to task j and γ_{ij} is an effective processing rate. Under a discrete allocation, one may restrict $u_{ij}(t)$ to be either zero or one with at most one active task per robot per step.

Partial observability is captured by introducing local observation sets [20]. Each robot i senses its environment within a radius that may depend on its sensing capabilities and the environment. Denote by $\mathcal{O}_i(t)$ the set of tasks observable by robot i at time t. This set consists of tasks located within a certain graph distance from $s_i(t)$ and may be subject to detection probabilities and false alarms. Similarly, robot i communicates with neighbors defined by a communication graph

$$G_c(t) = (\mathcal{R}, E_c(t)),$$

where $E_c(t)$ contains undirected edges representing bidirectional communication links. This communication graph may change over time as robots move [21].

The global system state includes the positions of all robots, the workloads and locations of all tasks, and possibly additional variables such as robot energy reserves. Under partial observability, robot i has access only to its local state, the tasks in $\mathcal{O}_i(t)$, and information received from neighbors. To formalize this, one can define the observation of robot i at time t as

$$o_i(t) = \big(s_i(t), [22]\mathcal{O}_i(t), \mathcal{M}_i(t)\big),$$

where $\mathcal{M}_i(t)$ denotes messages received from neighboring robots. The evolution of the system can be modeled as a stochastic process with transition probabilities influenced by robot actions and exogenous task arrivals. The decision problem faced by each robot is then a partially observable stochastic control problem.

Although a full multi-robot partially observable Markov decision process model would be extremely high dimensional, it is useful to think in those terms when designing local policies [23]. In principle, one could associate with each robot a belief over aspects of the global state and derive policies using stochastic dynamic programming. In practice, exact computations are intractable, motivating approximate methods in which robots maintain low-dimensional summaries of information, such as estimates of task utilities and congestion levels. The stochastic agent-based metaheuristics considered in this paper operate on such summaries rather than on full belief states.

To connect the global assignment perspective with local decision making, one can interpret the cost coefficients C_{ij} as encoding a notional desirability of assigning robot i to task j. Each robot seeks to estimate these desirabilities or their surrogates using local information [24]. Let

$$q_{ij}(t)$$

denote the local estimate maintained by robot i for the utility or negative cost of committing to task j at time t. Because robot i cannot observe all tasks, $q_{ij}(t)$ is defined only for tasks in a local candidate set

$$\mathcal{T}_i(t) \subseteq \mathcal{T},$$

which may consist of tasks in $\mathcal{O}_i(t)$ and tasks about which information has been received through communication. The agent-based metaheuristics determine how these utility estimates are updated and how they are translated into stochastic decisions about which tasks to service [25].

The global performance of the swarm can be evaluated through metrics such as average task completion time, fraction of tasks completed before deadlines, average distance traveled by robots, and load balancing across different robot types. Many of these performance criteria can be expressed as expectations of linear functionals of the system trajectories. For example, if T_j denotes the completion time of task j, one may consider the expected weighted sum

$$J = \mathbb{E}\left[[26]\sum_{j\in\mathcal{T}} w_j T_j\right],$$

where w_j are nonnegative weights. The agent-based metaheuristics aim to achieve reasonably low values of such global objective functions using only local decision rules operating under partial observability [27].

3. Stochastic Agent-Based Metaheuristics

In the metaheuristic framework considered here, each robot maintains a collection of local variables that summarize its knowledge and preferences regarding tasks. At each time step, a robot updates these variables based on its latest observations and information received from neighbors, then uses them to sample a task to pursue. The overall system behavior emerges from the interaction of these stochastic local decisions and the underlying task dynamics.

A central element of the metaheuristic is a set of task utility estimates $q_{ij}(t)$ that robot i associates with tasks in its local candidate set $\mathcal{T}_i(t)$. These estimates reflect the anticipated benefit of committing to task j in terms of completion reward, travel cost, and anticipated congestion [28]. The agent-based algorithm specifies how $q_{ij}(t)$ is updated and how it is transformed into a probability distribution over tasks.

A common choice for mapping utilities to probabilities is a soft-max or Boltzmann selection mechanism. For each robot i and task $j \in \mathcal{T}_i(t)$, define the probability of selecting task j at time t as

$$p_{ij}(t) = \frac{\exp\left(\beta_i(t) q_{ij}(t)\right)}{\sum_{k \in \mathcal{T}_i(t)} \exp\left([29]\beta_i(t) q_{ik}(t)\right)},$$

where $\beta_i(t)$ is an inverse temperature parameter that controls the trade-off between exploration and exploitation. When $\beta_i(t)$ is small, the probabilities are nearly uniform and the robot explores tasks broadly. As $\beta_i(t)$ increases, the robot becomes more likely to select tasks with higher estimated utility [30].

The utility estimates $q_{ij}(t)$ are updated using local reward information and possibly information from neighbors. Let $r_{ij}(t)$ denote the instantaneous reward or negative cost observed by robot i when it works on task j during time step t. This reward may depend on travel distance incurred, progress made on the task, and congestion due to simultaneous servicing by other robots. A simple stochastic approximation update rule is

$$q_{ij}(t+1) = (1 - \alpha_i(t)) q_{ij}(t) + \alpha_i(t) r_{ij}(t),$$

for the task actually selected at time t, and

$$q_{ik}(t+1) = q_{ik}(t)$$

for tasks $k \neq j$ [31]. Here $\alpha_i(t)$ is a learning rate sequence that may decrease over time. This rule implements an exponential moving average of observed rewards.

To incorporate information from neighbors, robots can periodically exchange their utility estimates or summaries thereof. Let $\mathcal{N}_i(t)$ denote the set of neighbors of robot i in the communication graph at time t. A consensus-like update step can be written as [32]

$$\tilde{q}_{ij}(t) = (1 - \lambda_i(t)) q_{ij}(t) + \lambda_i(t) \frac{\sum_{k \in \mathcal{N}_i(t)} q_{kj}(t)}{|\mathcal{N}_i(t)|},$$

for tasks j that appear in the union of candidate sets of robot i and its neighbors. The updated utility values are then [33]

$$q_{ij}(t+1) = \tilde{q}_{ij}(t)$$

for all relevant tasks. The coefficient $\lambda_i(t)$ controls the influence of neighbors' information and may depend on communication reliability or bandwidth.

Heterogeneity is integrated into the reward structure and task candidate sets. For robot i, only tasks j in $\mathcal{F}$ are considered, and the instantaneous reward is shaped by the capability vector c_i and requirement vector r_j. For example, if ρ_{ij} denotes the effective rate at which robot i can process task j, and $d_{ij}(t)$ is the travel distance from $s_i(t)$ to the location of task j, one may define a reward of the form

$$r_{ij}(t) = \eta_1 \rho_{ij} - \eta_2 d_{ij}(t) - \eta_3 z_j(t), [34]$$

where η_1, η_2, η_3 are nonnegative coefficients and $z_j(t)$ is an estimate of congestion at task j. Congestion can be estimated locally by counting how many robots announce an intention to work on task j or indirectly through observations of service delays.

The congestion term is important for avoiding over-allocation of robots to a small subset of tasks, which can lead to inefficient behavior. A simple congestion estimate for robot i could be [35]

$$z_j(t) = \sum_{k \in \mathcal{N}_i(t)} \mathbf{1}\{a_k(t) = j\},$$

where $a_k(t)$ denotes the task that neighbor k has selected at time t and $\mathbf{1}\{\cdot\}$ is the indicator function. Although this estimate is local and potentially noisy under partial observability, it provides a mechanism for robots to respond to local crowding [36].

The temperature parameter $\beta_i(t)$ can be adapted over time to implement an annealing schedule. At early stages or when the environment changes rapidly, lower values of $\beta_i(t)$ promote exploration. As time progresses or as task arrival rates stabilize, higher values shift behavior toward exploitation of high-utility tasks. A simple schedule is [37]

$$\beta_i(t) = \beta_{i,0} + \kappa_i t,$$

where $\beta_{i,0}$ is an initial inverse temperature and κ_i is a nonnegative slope. More elaborate schedules can make $\beta_i(t)$ depend on observed variability of rewards or measures of convergence such as the entropy of the task selection distribution.

In addition to utility-based selection, robots must decide when to release tasks and reassign themselves [38]. If a robot finds that a selected task has been completed by others or is no longer beneficial due to congestion or environmental changes, it should be able to abandon the task and sample a new one. This is naturally handled by the stochastic metaheuristic: at each time step, robots probabilistically reconsider their assignments based on updated utilities. The rate of reassignment can be influenced by the learning rate and temperature parameters, with faster updating promoting flexibility but potentially increasing oscillations.

To provide a more explicit connection to the global linear assignment model, one can view the stochastic agent-based mechanism as performing a distributed approximation of the assignment problem through local sampling [39]. Suppose that the utility estimates $q_{ij}(t)$ converge, in an average sense, to negative scaled versions of the costs C_{ij}. Then the soft-max selection rule approximates a randomized assignment in which lower-cost pairs are exponentially more likely to be chosen. The learning rule progressively refines these estimates based on observed travel and processing outcomes, while neighbor communication serves to propagate information about task desirability through the network.

From a metaheuristic perspective, the combination of stochastic selection, local learning, and occasional reallocation resembles features of simulated annealing and population-based search. Robots can be thought of as parallel search processes exploring the space of allocations, with communication implementing a form of information sharing analogous to crossover or migration in evolutionary methods [40]. The partial observability and heterogeneity introduce additional structure, as robots of different types explore different subsets of tasks and see different portions of the environment.

The scalability of this approach stems from the fact that each robot maintains utility estimates only for a relatively small number of tasks, namely those within its observation region or those advertised by neighbors. The computational burden per robot per time step is dominated by evaluating rewards for candidate tasks, updating the utility estimates, and sampling from the soft-max distribution [41]. All of these operations scale linearly with the size of the local candidate set. Communication requirements are similarly local, involving the exchange of utility estimates or summaries with neighbors. These properties make the metaheuristic amenable to large-scale swarms where fully centralised computation is impractical.

4. Analysis and Discussion of Partial Observability

The behavior of stochastic agent-based metaheuristics in heterogeneous swarms is substantially influenced by partial observability [42]. The local utility estimates that drive task selection are constructed from incomplete information, and the communication graph may not always connect robots that are servicing related tasks. Understanding these effects is important for interpreting performance and for choosing algorithmic parameters.

A useful conceptual model treats each robot as operating in a partially observable Markov decision problem defined on a reduced state space. Let S denote a global state space describing the positions and statuses of all robots and tasks, and let A_i denote the action space of robot i, consisting of available motion and task selection commands [43]. Robot i receives observations $o_i(t)$ that depend on the global state through an observation kernel. If robot i maintained a belief $b_i(t)$ over S, the belief update could be expressed as

$$b_i(t+1, s') = \eta_i(o_i(t+1))[44]O_i(o_i(t+1) \mid s') \sum_{s \in S} P(s' \mid s, a_i(t), a_{-i}(t)) b_i(t, s),$$

where P is the state transition kernel, O_i is the observation likelihood, $a_{-i}(t)$ denotes the joint actions of other robots, and η_i is a normalization factor. In practice, such belief updates are not computed and the utility estimates $q_{ij}(t)$ play the role of compressed information about the environment.

One can think of the local utility estimate for task j as a projection of the high-dimensional belief onto a scalar value that captures expected reward for servicing that task. The stochastic approximation update [45]

$$q_{ij}(t+1) = (1 - \alpha_i(t)) q_{ij}(t) + \alpha_i(t) r_{ij}(t)$$

can be analyzed using standard results on stochastic approximation if the reward process is ergodic and the learning rate satisfies appropriate conditions. Under such conditions, $q_{ij}(t)$ may converge in probability to a value that reflects the long-run average reward of choosing task j from the perspective of robot i. However, partial observability and the coupling between robots' decisions complicate the verification of these conditions.

Partial observability affects reward signals through two primary channels [46]. First, robots may fail to detect tasks, especially when sensing ranges are limited or occlusions occur. Second, robots may have incomplete or outdated information about the intentions of other robots, leading to misestimation of congestion and potential overcommitment to tasks. These effects introduce biases and increased variance in the observed rewards $r_{ij}(t)$, which in turn influence the convergence of the utility estimates.

To gain insight, consider a simplified regime in which tasks are static, all robots have fixed positions, and only task detection is subject to uncertainty. Suppose that task j lies within the nominal sensing range of robot i but is detected at each time step with probability δ_{ij}. When the task is not detected, it is effectively absent from the candidate set $\mathcal{T}_i(t)$, so the soft-max selection mechanism cannot choose it. Over many time steps, the effective probability that i selects task j depends on both the detection probability and the soft-max probabilities conditioned on detection [47].

If the reward associated with task j is relatively stable, the stochastic approximation process sees a subsequence of reward observations corresponding to the times when the task is detected and selected. Under mild regularity conditions, the estimated utility $q_{ij}(t)$ can still converge to a value close to the expected reward, but the convergence is slower and more variable due to the missing observations. The main consequence is that tasks with low detection probabilities may be persistently undervalued relative to tasks that are more easily observed, leading to allocation biases.

Communication among robots provides a mechanism to mitigate some of these biases. When a robot that consistently detects a task shares its utility estimates with neighbors, robots that rarely detect the task directly can still form reasonably accurate estimates [48]. The consensus-like update

$$\tilde{q}_{ij}(t) = (1 - \lambda_i(t))q_{ij}(t) + \lambda_i(t)\frac{\sum_{k\in\mathcal{N}_i(t)} q_{kj}(t)}{|\mathcal{N}_i(t)|}$$

tends to diffuse information about tasks through the communication graph [49]. The effectiveness of this diffusion depends on the graph connectivity and on the relative magnitudes of learning and consensus rates. If $\lambda_i(t)$ is too small compared to $\alpha_i(t)$, local reward observations dominate and information diffuses slowly. If $\lambda_i(t)$ is too large, robots may become overly reliant on potentially outdated neighbor information.

Heterogeneity interacts with partial observability because robots of different types may have different sensing ranges, detection probabilities, and communication capabilities [50]. Specialized robots may be more likely to detect certain tasks due to their sensing payloads, and may also be more capable of servicing those tasks due to their actuation capabilities. In such cases, it can be beneficial for the metaheuristic to emphasize the propagation of task information from specialized agents to more general agents, so that the swarm as a whole maintains awareness of task distributions even if not all robots can perceive every task directly.

From a linear modeling perspective, partial observability effectively modifies the costs and feasibility structure of the global assignment. A robot that cannot reliably detect a task behaves as if the cost of servicing that task were very high or as if the task were infeasible [51]. One can formalize this by introducing effective costs

$$\bar{C}_{ij} = \frac{C_{ij}}{\delta_{ij}},$$

with the understanding that small detection probabilities inflate effective costs. The global assignment problem with effective costs captures the long-run difficulty of coordinating robots to service tasks under sensing limitations [52]. The distributed metaheuristic can then be interpreted as approximating an assignment with respect to these effective costs, rather than the nominal costs.

Another important aspect of partial observability is the presence of delayed and noisy information about other robots' decisions. When robots communicate only intermittently, their beliefs about congestion at tasks may lag behind reality. A robot may arrive at a task expecting low congestion based on outdated neighbor messages, only to discover that several other robots have already committed to that task [53]. Such discrepancies can create transient inefficiencies, but they also inject useful stochasticity into the system by preventing robots from perfectly predicting each other's actions.

The impact of such delays on convergence can be examined using Markov chain models of the joint allocation state. Let $X(t)$ denote a random variable capturing the current assignment of robots to tasks. Under the stochastic metaheuristic, $X(t)$ evolves as a Markov chain whose transition probabilities depend on the utility estimates and on the observation and communication processes [54]. If the learning rates and temperatures change slowly, one can approximate the evolution of $X(t)$ as a quasi-static Markov chain with slowly varying parameters. Under suitable conditions, the chain may admit a stationary distribution that concentrates on allocations that are locally consistent with high utility estimates.

Partial observability perturbs this stationary distribution by altering transition probabilities through missed detections and misinformation. However, the presence of stochastic decision making and communication can preserve ergodicity, ensuring that the chain continues to explore the allocation space [55]. In this view, partial observability does not necessarily prevent convergence to useful allocation patterns, but it modifies the set of likely allocations and the timescales over which they are reached.

Another perspective connects the metaheuristic dynamics to potential games and distributed optimization. In a potential game formulation, one defines a global potential function, often related to the negative of the total cost, such that any unilateral change in a robot's task assignment produces a corresponding change in the potential equal to the robot's utility change. If the utility functions are designed carefully, the stochastic task selection rule can be interpreted as a form of logit response dynamics in a potential game [56]. Under full observability, such dynamics are known to converge in distribution to a Gibbs measure concentrated around global or local maxima of the potential. Partial observability perturbs the utility functions through estimation error, but as long as these errors are not too large or systematically biased, the dynamics may still favor allocations with relatively high potential.

In practice, the severity of partial observability and its effects on the metaheuristic depend on environmental factors and hardware characteristics [57]. Dense task environments with significant occlusion may challenge sensing, while sparse environments with long-range communication may mitigate observation gaps. The design of the metaheuristic parameters, including learning and consensus rates, temperature schedules, and reward structures, should be informed by these factors. In particular, settings with strong partial observability may benefit from slower learning rates that average out noisy observations, higher consensus rates that propagate information more aggressively, and exploration mechanisms that encourage robots to occasionally search beyond their typical sensing neighborhood.

Overall, partial observability does not invalidate the use of stochastic agent-based metaheuristics for task allocation, but it shapes their behavior in ways that must be accounted for in analysis and design [58]. By embedding partial observability into the modeling framework and interpreting its influence through linear cost modifications, stochastic approximation, and Markov chain dynamics, one can develop intuition about how different parameter regimes will perform and how to tune algorithms for specific deployment scenarios.

5. Conclusion

This paper has examined stochastic agent-based metaheuristics for distributed task allocation in heterogeneous robotic swarms operating under partial observability. A global linear assignment formulation was introduced as a conceptual reference, capturing the ideal allocation of heterogeneous robots to tasks when full information is available. Building on this, a decentralized framework was developed in which each robot maintains local utility estimates for tasks, updates these estimates using stochastic approximation based on observed rewards, and selects tasks according to a soft-max rule [59]. Heterogeneity was incorporated through capability and requirement vectors, which shape both the feasibility of allocations and the rewards perceived by different robots.

Partial observability was modeled through local sensing and communication structures that restrict each robot's view of the global state. The resulting biases and variances in reward observations influence the convergence of utility estimates and the emergent allocation patterns. To mitigate these effects, consensus-like information exchange mechanisms were considered, allowing robots to incorporate neighbor estimates into their own decision processes [60]. The combined dynamics suggest that even under limited sensing and communication, swarms can approximate solutions to effective assignment problems whose costs incorporate detection and communication limitations.

The discussion highlighted how annealing schedules, learning rates, and consensus gains interact to balance exploration and exploitation, and how stochasticity in the decision process can help avoid persistent local conflicts and facilitate adaptation to changing task distributions. The analysis also indicated that heterogeneity can be leveraged to improve observability and coverage, as specialized robots may serve as information hubs for tasks that are difficult for others to detect, thereby shaping the overall swarm behavior through local communication and utility propagation.

The presented framework and analysis are intended as a foundation for understanding and designing stochastic metaheuristic task allocation schemes in realistic swarm settings [61]. Future work may explore more detailed models of communication constraints, such as bandwidth limits and packet loss, and may consider richer reward structures that capture risk, safety margins, and temporal deadlines. It is also of interest to investigate how learning-based components, including value function approximation and policy search methods, can be integrated into the agent-based metaheuristics while preserving scalability and robustness under partial observability. Another direction is to study how environmental structure, such as clustering of tasks or time-varying obstacles, interacts with the stochastic decision dynamics to influence performance. By continuing to refine both the modeling and algorithmic aspects, it may be possible to develop task allocation mechanisms that are more systematically aligned with the constraints and opportunities inherent in heterogeneous robotic swarms [62].

References

[1] A. Garcia, C. J. P. de Lucena, F. Zambonelli, A. Omicini, and J. Castro, *Software Engineering for Large-Scale Multi-Agent Systems - Software Engineering for Large-Scale Multi-Agent Systems*. Germany: Springer Berlin Heidelberg, 4 2003.

[2] C. Nowzari, J. Cortes, and G. J. Pappas, “Event-triggered communication and control for multi-agent average consensus,” 1 2016.

[3] D. Latocha and P. Arabas, *AUTOMATION - Multi-agent System for On-Line Game Matchmaking*. Springer International Publishing, 2 2016.

[4] B. Zhao, Y. Guan, and L. Wang, “Non-fragility of multi-agent controllability,” *Science China Information Sciences*, vol. 61, no. 5, pp. 052 202–, 12 2017.

[5] D. Rodrigues, “Ciuceu: Multi-agent-based simulation of university email communities,” *International Journal of Agent Technologies and Systems*, vol. 1, no. 4, pp. 30–48, 10 2009.

[6] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, “An ant odor analysis approach to the ant colony optimization algorithm for data-aggregation in wireless sensor networks,” in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2006, pp. 1–4.

[7] M. Ragaglia, M. Prandini, and L. Bascetta, *MESAS - Multi-agent Poli-RRT**. Germany: Springer International Publishing, 10 2016.

[8] Y. Zheng, X. Xu, T. Shi, D. Fang, and P. An, “A multi-agent approach to design a wlan system,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 13, no. 11, pp. 46–64, 11 2017.

[9] L. Wenjun, W. Tianjiang, and L. Fang, “Multi-agent group programming based on co-evolution,” *The Computer Journal*, vol. 52, no. 8, pp. 910–921, 1 2009.

[10] M. D. Oliveira and M. Purvis, *Aspects of Openness in Multi-Agent Systems*. IGI Global, 1 2011.

[11] S. Kumar, P. Shah, D. Hakkani-Tur, and L. Heck, “Federated control with hierarchical multi-agent deep reinforcement learning,” 1 2017.

[12] T. Kristensen, Bech, and M. Dyngeland, “Towards a multi-agent e-learning platform,” *Journal of Computer Engineering and Informatics*, vol. 1, no. 2, pp. 64–81, 4 2013.

[13] M. Bouzahzah and R. Maamri, "An approach for fault tolerance in multi-agent systems using learning agents," *International Journal of Intelligent Information Technologies*, vol. 11, no. 3, pp. 30–44, 7 2015.

[14] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han, "Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos," in *Proceedings of the 17th ACM international conference on Multimedia*, 2009, pp. 721–724.

[15] J. Wang, Q. Zhang, and H. Li, *Delay Consensus of Second-Order Nonlinear Leader-Following Multi-Agent Systems*. Germany: Springer Berlin Heidelberg, 11 2015, pp. 615–623.

[16] N. Bessonov, I. Demin, P. Kurbatova, L. Pujo-Menjouet, and V. Volpert, *Multi-Agent Systems and Blood Cell Formation*. InTech, 4 2011.

[17] J. Zhang, "Multi-agent-based equipment data acquisition," 4 2017.

[18] Y.-H. Zhu, L.-Y. Li, J.-X. Wang, and D. Wang, "Research on quality grade classification method of daily-use porcelain based on multi-agent technology," *ITM Web of Conferences*, vol. 12, pp. 03 014–, 9 2017.

[19] W. Chen, L. Gao, X. Xu, and J. Cui, "Distributed observer-based consensus protocol for descriptor multi-agent systems," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 229, no. 10, pp. 927–938, 9 2015.

[20] M. S. Devi and M. Arora, "Design of multi agent system for resource allocation and monitoring," *International Journal of Agent Technologies and Systems*, vol. 3, no. 1, pp. 1–10, 1 2011.

[21] Y. Zhao, G. Wen, and G. Chen, *Distributed Finite-Time Cooperative Control of Multi-agent Systems*. Germany: Springer Berlin Heidelberg, 8 2015, pp. 163–206.

[22] J. Wang, "Network tutoring strategy model based on multi-agent plan," in *SPIE Proceedings*, vol. 8335. SPIE, 1 2012, pp. 55–61.

[23] R. Chandrasekar, V. Vijaykumar, and T. Srinivasan, "Probabilistic ant based clustering for distributed databases," in *2006 3rd International IEEE Conference Intelligent Systems*. IEEE, 2006, pp. 538–545.

[24] B. Banerjee, S. Loscalzo, and D. Thompson, "Detection of plan deviation in multi-agent systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 3 2016.

[25] V. Julián, C. Carrascosa, M. Rebollo, J. Soler, and V. Botti, "Ccia - simba: An approach for real-time multi-agent systems," *Lecture Notes in Computer Science*, pp. 282–293, 10 2002.

[26] T. Miyamoto, M. Sakamoto, and S. Kumagai, "On reachability analysis of multi agent nets," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, no. 10, pp. 2257–2260, 10 2007.

[27] B. Hettige, A. S. Karunananda, and G. Rzevski, "A multi-agent solution for managing complexity in english to sinhala machine translation," *International Journal of Design & Nature and Ecodynamics*, vol. 11, no. 2, pp. 88–96, 4 2016.

[28] W. Yu, H. Wang, H. Hong, and G. Wen, "Distributed cooperative anti-disturbance control of multi-agent systems: an overview," *Science China Information Sciences*, vol. 60, no. 11, pp. 110 202–, 9 2017.

[29] J. Zhang, Z. Li, L. Zhuo, X. Liu, and Y. Yang, "A compressed-domain image filtering and re-ranking approach for multi-agent image retrieval," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 03, pp. 1 552 001–, 4 2015.

[30] K. R. Kumar, S. Anand, and M. Sydulu, "Comparison of multi agent based pso approach for optimal power flows with security constraints," *Applied Mechanics and Materials*, vol. 110-116, pp. 5206–5210, 10 2011.

[31] S. A. DeLoach, "Canadian conference on ai - modeling organizational rules in the multi-agent systems engineering methodology," *Lecture Notes in Computer Science*, pp. 1–15, 5 2002.

[32] R. Chandrasekar and T. Srinivasan, "An improved probabilistic ant based clustering for distributed databases," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI*, 2007, pp. 2701–2706.

[33] M. Sierhuis, W. J. Clancey, and R. van Hoof, *Multi-Agent Programming, Languages, Tools and Applications - Brahms An Agent-Oriented Language for Work Practice Simulation and Multi-Agent Systems Development*. Springer US, 5 2009.

[34] S. Kurihara, "Www (companion volume) - the multi agent based information diffusion model for false rumordiffusion analysis," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 4 2014, pp. 1319–1320.

[35] L. Yu and G. Juanqiong, *Multi-Agent Synergy in the Behavior—Intensive CA Model of Road Intersection*. Springer Berlin Heidelberg, 4 2015, pp. 483–487.

[36] R. R. Penner, "Multi-agent societies for collaborative interaction," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 40, no. 15, pp. 762–766, 10 1996.

[37] G.-S. Han, Z.-H. Guan, J. Li, D.-X. He, and D.-F. Zheng, "Multi-tracking of second-order multi-agent systems using impulsive control," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1771–1781, 1 2016.

[38] Z. Yang, Q. Zhang, and Z. Chen, "Flocking of multi-agents with nonlinear inner-coupling functions," *Nonlinear Dynamics*, vol. 60, no. 3, pp. 255–264, 9 2009.

[39] F. Yan and D. Xie, "Containment control of multi-agent systems with time delay:," *Transactions of the Institute of Measurement and Control*, vol. 36, no. 2, pp. 196–205, 7 2013.

[40] A. Helleboogh, G. Vizzari, A. M. Uhrmacher, and F. Michel, "Modeling dynamic environments in multi-agent simulation," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 87–116, 7 2006.

[41] T. Srinivasan, V. Vijaykumar, and R. Chandrasekar, "An auction based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*. IEEE, 2006, pp. 5–pp.

[42] Q. Wang, H. Gao, F. E. Alsaadi, and T. Hayat, "An overview of consensus problems in constrained multi-agent coordination," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 275–284, 3 2014.

[43] D. Borri, *Frames, Multi-Agents and Good Behaviours in Planning Rationales*. Springer Netherlands, 8 2010, pp. 235–256.

[44] M. M. Öztürk, I. Cil, and A. Zengin, "Development of a multi-agent framework for software quality," *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1–10, 2 2015.

[45] S. Salamatian, A. Beirami, A. Cohen, and M. Médard, "Centralized vs decentralized multi-agent guesswork," 1 2017.

[46] K.-C. Jim and C. L. Giles, "Agents - how communication can improve the performance of multi-agent systems," in *Proceedings of the fifth international conference on Autonomous agents*. ACM, 5 2001, pp. 584–591.

[47] J. Melesko, E. Kurilovas, and I. Krikun, “Multi-agent system for computer science and engineering education,” *Lietuvos matematikos rinkinys*, vol. 58, pp. 57–61–57–61, 12 2017.

[48] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” 1 2017.

[49] S. Thrun, “Maple: Multi-agent planning, learning, and execution,” 2 2004.

[50] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, “An obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs,” in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, 2006, pp. 1–6.

[51] S. Didi and G. Nitschke, *EvoApplications (2) - Multi-agent Behavior-Based Policy Transfer*. Germany: Springer International Publishing, 4 2016, pp. 181–197.

[52] Y. Su and J. Huang, “Cooperative robust output regulation of a class of heterogeneous linear uncertain multi-agent systems†,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2819–2839, 6 2013.

[53] X. Mao, F. Hou, and W. Wu, *Multi-Agent System Approach for Modeling and Supporting Software Crowdsourcing*. Springer Berlin Heidelberg, 5 2015, pp. 73–89.

[54] H. Zhao, S. Xu, and D. Yuan, “Consensus of data-sampled multi-agent systems with markovian switching topologies†,” *Asian Journal of Control*, vol. 14, no. 5, pp. 1366–1373, 8 2011.

[55] A. Bidram, V. Nasirian, A. Davoudi, and F. L. Lewis, *Introduction to Multi-agent Cooperative Control*. Springer International Publishing, 2 2017, pp. 45–65.

[56] X. Wang, W. Ni, and Z. Ma, “Distributed event-triggered output regulation of multi-agent systems,” *International Journal of Control*, vol. 88, no. 3, pp. 640–652, 11 2014.

[57] M. Popovici and C. Dobre, “Wims - a game-theoretic approach to cooperation in multi-agent systems,” in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*. ACM, 6 2012, pp. 54–4.

[58] F. Chen, Y. Chen, and R. Feng, “A probability based model for multi-agent system optimization,” in *SPIE Proceedings*, vol. 7749. SPIE, 7 2010, pp. 327–330.

[59] R. Chandrasekar, R. Suresh, and S. Ponnambalam, “Evaluating an obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs,” in *2006 International Conference on Advanced Computing and Communications*. IEEE, 2006, pp. 628–629.

[60] Y. Shang, “Consensus recovery from intentional attacks in directed nonlinear multi-agent systems,” *ijnsns*, vol. 14, no. 6, pp. 355–361, 8 2013.

[61] W. Qin, Z. Liu, and Z. Chen, “Impulsive observer-based consensus control for multi-agent systems with time delay,” *International Journal of Control*, vol. 88, no. 9, pp. 1789–1804, 2 2015.

[62] N. Prudence, W. Jeberson, and B. Mazumdar, “Multi-agent negotiation strategies for task allocation process in e-commerce system,” *International Journal of Engineering Research and*, vol. 5, no. 04, 4 2016.