Swarm Robotics: A Survey from a Multi-Tasking Perspective

Author:
Debie, E; Kasmarik, K; Garratt, M

Publication details:
ACM Computing Surveys
v. 56
Chapter No. 2
pp. 1 - 38
0360-0300 (ISSN); 1557-7341 (ISSN)

Publication Date:
2024-02-29

Publisher DOI:
https://doi.org/10.1145/3611652

[1]

Swarm robotics: A Survey from a Multi-tasking Perspective

ESSAM DEBIE[†], KATHRYN KASMARIK[*], and MATT GARRATT[*], University of New South Wales, Canberra, Australia

The behaviour of social insects such as bees and ants has influenced the development of swarm robots. To enable robots to cooperate together, swarm robotics employs principles such as communication, coordination, and collaboration. Collaboration among multiple robots can lead to a faster task completion time compared to the utilisation of a single, complex robot. One of the key aspects of swarm robotics is that control is distributed uniformly across the robots in the swarm, which boosts the system's resilience and fault-tolerance. Through the use of the robots' embodied sensors and actuators, this distributed control often facilitates the emergence of collective behaviours through the interaction of the robots with one another and with the environment. The purpose of this survey is to examine the reasons behind the lack of utilisation of swarm robots in multi-tasking applications, which will be accomplished by studying previous research works in the field. We examine the literature from the perspective of multi-tasking: we pay particular attention to concepts that contribute to the progress of swarm robotics for multi-tasking applications. In order to do this, we first examine the different studies in multi-tasking swarm robotics, covering platforms, multi-tasking scenarios, sub-task allocation methodologies, as well as performance metrics. We then highlight several swarm robotics-related disciplines that have significant effect on the development of swarm robotics for multi-tasking problems. We propose two taxonomies: the first categorises works based on the characteristics of the scenarios being handled, while the second taxonomy categorises works based on the swarming strategies utilised to achieve multi-tasking capabilities. We finish with a discussion of swarm robots' existing limitations for real-world multitasking applications, as well as recommendations for future research directions.

CCS Concepts: • **Computing methodologies → Cooperation and coordination**.

Additional Key Words and Phrases: swarm robotics, multi-tasking, reinforcement learning

ACM Reference Format:
Essam Debie, Kathryn Kasmarik, and Matt Garratt. 2023. Swarm robotics: A Survey from a Multi-tasking Perspective. 1, 1 (November 2023), 38 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A swarm of robots is a collection of mobile robots that work together to accomplish a shared objective[130]. Each individual robot has restricted capabilities in terms of processing and sensor power, as well as communication and mobility. Swarm robotics is an alternative approach to using a single, expensive, and complicated robot to complete tasks that are too difficult or dangerous for humans to perform alone. A swarm can be thought of as a distributed system in which robots

[*]Corresponding author.

Authors' address: Essam Debie, e.debie@unsw.edu.au; Kathryn Kasmarik, k.kasmarik@adfa.edu.au; Matt Garratt, m.garratt@adfa.edu.au, University of New South Wales, Canberra, Canberra, ACT, Australia, 2600.

XXXX-XXXX/2023/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

act as autonomous nodes and communicate via messages. Because a robotic distributed system employs an asynchronous timing model, each robot may perform tasks in any sequence and at any arbitrary pace [104, 130]. To achieve effective performance, the robotic swarm must be robust, resilient, and scalable in order to perform a wide range of tasks [28].

The social organisation of social insects, particularly their division of work, is a key factor in their ecological success. It is commonly acknowledged that splitting responsibilities among colony members leads to individuals specialising in specific roles. This makes the colony more efficient either because robots learn task-specific skills through practise, or due to spatial fidelity, in which individuals become increasingly spatially localised to fulfil specific tasks reducing the time and energy required to travel between different locations [21]. Task allocation patterns have been shown to be influenced by a number of factors, allowing them to be more efficient (i.e., allow for more tasks to be completed for the same amount of energy) while also being more flexible [23].

The difficulty of swarming tasks varies based on factors like complexity, robot numbers, and types. Single robots can handle small-scale tasks (mapping, pick and place, navigation) [61]. Complex tasks need multiple robots collaborating, ranging from loosely to tightly coordinated tasks [61]. Examples of weakly coordinated tasks include exploration, mapping, waste removal, tracking, and monitoring. Tightly coordinated tasks include soccer playing, item transportation, and massive structure building. Swarm cooperation significantly impacts complex tasks, surpassing traditional approaches with limited costly robots [73, 135]. Swarm agents collectively achieve successful outcomes, enabling applications like disaster management, environmental monitoring, and radar jamming [15, 53, 75]. Swarm robotics has progressed with diverse implementation components. Multitasking challenges are addressed using centralised workload distribution, but it faces issues with information accuracy and breakdown resilience [84, 100]. Recently, decentralised swarm systems with communication and bargaining gain attention [78].

1.1 Motivation

Real-life applications require a flexible and adaptable swarming system that is capable of performing multiple tasks simultaneously (solving complex problems through task partitioning and task allocation). Potential applications include agriculture (harvesting, targeted weed spraying), search and rescue (SAR), transportation and delivery of packages with different autonomous vehicles (ships/self-driving trucks,drones) used at different stages, environmental cleanup (e.g. some robots to find the contaminated regions, others to do the repair), air-crash investigation and many others.

1.2 Scope of This Survey

This survey includes three important areas related to robotic swarms for multi-tasking problems: (1) Nature of multi-tasking scenarios; (2) Approaches for sub-task allocation/selection; (3) and performance metrics.

1.3 Main Contribution

This paper aims to summarise the research papers presented in the literature that study swarming robotics in multi-tasking environments. The key contributions of this survey are described as follows:

(1) Discussion of the multi-tasking paradigm, covering its components, and requirements.
(2) Review of previous research work conducted on swarm robotics in multi-tasking scenarios, covering their approaches, design choices, advantages and limitations.

(3) Discussion of the benchmark problems commonly used in swarm robotic research for applications with multi-tasking requirements. We explain and highlight the differences among them, and their suitability for testing different aspects of multi-tasking work.
(4) Presentation of the current research challenges and possible future directions for research in swarm robotics for multi-tasking applications.

The remainder of this paper is organised as follows. In Section 2, existing surveys related to swarm robotics and multi-tasking problems are discussed. In Section 3 we introduce the systematic review methodology followed in this paper. Section 4 describes existing benchmark multi-tasking scenarios for swarm robotics. Following this, Section 5 presents our view of the stages required to achieve multi-tasking, including task decomposition and task allocation/selection. Different approaches to these stages are discussed. Section 6 describes the performance metrics that are available in the literature for evaluating swarm performance in multi-tasking contexts. Finally, discussions of current challenges and the future work directions and paper's conclusion are provided in Sections 7.

2 RELATED WORK

The task allocation problem in swarm robotics has been a significant research topic due to its complex and dynamic nature. To maximise the performance of a swarm system, tasks need to be distributed efficiently among robots. Consequently, researchers have introduced various optimisation approaches to solve this problem, which we classify into four broad categories: stochastic methods, bio-inspired algorithms, learning-based methods, and hybrid methods.

Stochastic Methods. Stochastic methods rely on probability and randomness to solve the task allocation problem. One popular approach is the Monte Carlo method, which uses random sampling to obtain numerical results [89]. This method, although computationally expensive, is capable of providing near-optimal solutions even in dynamic and uncertain environments. The Stochastic Hill Climbing (SHC) algorithm is another stochastic method that uses random exploration and probability to climb towards the optimal solution [18]. Stochastic methods have shown promising results in task allocation. They typically follow a decentralised approach, where each robot makes individual decisions based on locally available information. However, the computational complexity of these methods can be quite high, making them less suitable for large-scale swarm systems due to the computational and communication demands.

Bio-Inspired Algorithms. Bio-inspired algorithms mimic natural processes to solve complex problems. Notably, the Particle Swarm Optimisation (PSO) algorithm simulates the social behaviour of bird flocks or fish schools to search for optimal solutions [83]. Another commonly used bio-inspired approach is the Ant Colony Optimisation (ACO) algorithm, which emulates the pheromone trails left by ants to find the shortest path to food sources [43]. Bio-inspired algorithms are efficient and scalable, making them suitable for task allocation in swarm robotics. These algorithms typically adopt a distributed architecture, where information is implicitly exchanged among the swarm via environmental modifications. The computational complexity of these methods tends to be lower than that of stochastic methods, enabling better scalability. Nevertheless, their performance in dynamic environments may drop due to their reliance on fixed parameters.

Learning-Based Methods. Learning-based methods use machine learning algorithms to learn and adapt to the task allocation problem over time. Reinforcement Learning (RL), for instance, allows robots to learn optimal task allocation strategies by interacting with the environment and receiving feedback in terms of rewards or penalties [154]. Meanwhile, Deep Learning (DL) can help swarm systems learn complex task allocation strategies through layers of artificial neural networks [110]. Learning-based methods show high adaptability to dynamic environments and complex tasks, yet they require a large amount of data and extensive training time. These methods can operate in both centralised and decentralised settings. In a centralised setting, a single agent (or a subset of agents) learns the task allocation strategy based on global information, which can be computationally demanding and requires substantial communication bandwidth. In a decentralised setting, each agent learns individually based on local information, reducing the computational and communication requirements but potentially leading to sub-optimal solutions.

Hybrid Methods. Recently, researchers have proposed hybrid methods that combine two or more of the aforementioned approaches to overcome their individual limitations and boost their strengths. For example, a PSO-ACO hybrid method could enhance the global search ability of PSO with the local search ability of ACO [96]. Similarly, integrating RL into stochastic methods could reduce the computational requirements of the latter while maintaining their robustness [162]. Hybrid methods show promise in improving the performance and adaptability of task allocation in swarm robotics, opening new avenues for future research. However, these methods often involve complex information flow management strategies to ensure efficient integration and scalability. The computational complexity of these methods depends on the specific combination of approaches used.

In general, the choice of optimisation approach and the system architecture (centralised, decentralised, distributed) depends on various factors, including the number of robots, the complexity of tasks, and the available communication bandwidth. Each method brings its own trade-off between computational complexity and optimisation performance, as well as unique challenges and opportunities related to managing information flow. Understanding these nuances will be crucial in the development of efficient task allocation strategies in swarm robotics.

Various methodologies have evolved in swarm robotics for task allocation, driving practical applications and field advancements. Surveys and taxonomies have been conducted to understand the field, but most focus on specific aspects, not addressing multi-tasking applications and design requirements. Table 1 compares our survey with others, which mainly concentrate on group architecture, engineering, or environment. For example, Dudek et al. [49] proposed a taxonomy for classifying multi-agent robotic systems based on various communication and processing factors. Cooperation in robots was presented as a hierarchical overview by Cao et al.[29]. There are five categories: group architecture, resource conflicts, cooperative origins, and geometric difficulties. In addition, group architecture is separated into the following categories: centralisation / decentralisation, differentiation (which implies homogenous or heterogeneous robot groups), communication structure, and dimension modelling of other agents. Other agents' intents, beliefs, actions, capabilities, and states are all modelled in order to improve cooperation amongst robots in this dimension. In their taxonomy, Iocchi et al. [72] offered a taxonomy of multi-robot systems and addressed certain multi-robot system investigations. They used levels to convey their taxonomy in a hierarchical format. The first level is collaboration, which is separated into aware and unaware categories. The degree to which the aware category is divided into three types: not coordinated, weakly coordinated, and strongly coordinated. At the organisational level, the strongly-coordinated category is separated into three categories: strongly-centralised, weakly-centralised, and dispersed. The application domains of multi-robot systems were also described in a separate section. Gazi and Fidan [59] overviewed multi-agent systems with a focus on dynamics and control. Brambilla et al. [24] reviewed swarm robotics for real-world applications. Chung et al. [35] investigated integrating aerial and ground robots. Connor et al. [37] reviewed underwater swarm robotics. Bayındır [16] surveyed swarm tasks, while Schranz et al. [133] focused on swarm behaviour. Majid et al. [105]

Table 1. Previous surveys on swarm robotics.

Survey Ref	Main contribution
Dudek et al. [49]	Taxonomy based on swarm size, communication range, topology, bandwidth, swarm reconfigurability and swarm processing capability
Cao et al. [29]	Taxonomy of cooperative robotics based on five dimensions group architecture, resource conflicts, cooperative origins, learning, and geometric issues
Iocchi et al. [72]	Hierarchical taxonomy of robotic swarm with focus on cooperation and coordination
Gazi and Fidan [59]	System dynamics and control
Brambilla et al. [24]	Analysis of the swarm robotics literature from the perspective of swarm engineering, proposing taxonomies to classify works dealing with design and analysis methods
Bayındır [16]	Taxonomy of swarm tasks
Chung et al. [35]	Survey of theoretical tools developed and applied to aerial swarms with focus on trajectory generation, task allocation, adversarial control, distributed sensing, monitoring, and mapping
Nedjah and Junior [113]	Reviews existing research in swarm robotics (SR) and categorises it into groups dealing with SR design and tasks required in SR.
Connor et al. [37]	Underwater swarm robotics
Schranz et al. [133]	Taxonomy of different swarm behaviour and classification of robotic applications from the scientific and industry fields
Majid et al. [105]	Technical review of swarm robotics tasks, which are divided into low-level and high-level tasks
Cheraghi et al. [34]	Review of the history and pioneers of swarm robotics, its current presence in simulators, and real-life applications, and visions and ideas for the future of swarm robotics.
Dias et al. [42]	Overview of current activities in Swarm Robotics and examination of the literature to establish a connection between swarm robotic system and real-world implementations.
Dorigo et al. [44]	Summarises the main lessons learned and open problems still to be solved. The main avenues of research in the future are also presented, with application demands
Albiero et al. [7]	Systematic literature review of 32 papers that reported research on swarm robots in agriculture

provided a technical review of swarm robotics tasks, hardware, software, challenges, and future directions. Table 1 summarises previous swarm robotics survey studies.

3 REVIEW METHODOLOGY

The systematic review was performed using Scopus and Google Scholar databases and only includes studies that are related to application of swarming to multi-tasking problems. The search terms used to collect the studies included in the survey are as follows: ((multi AND tasking) OR (division AND labor)) AND swarm* AND robotic*). The electronic databases were filtered to improve the

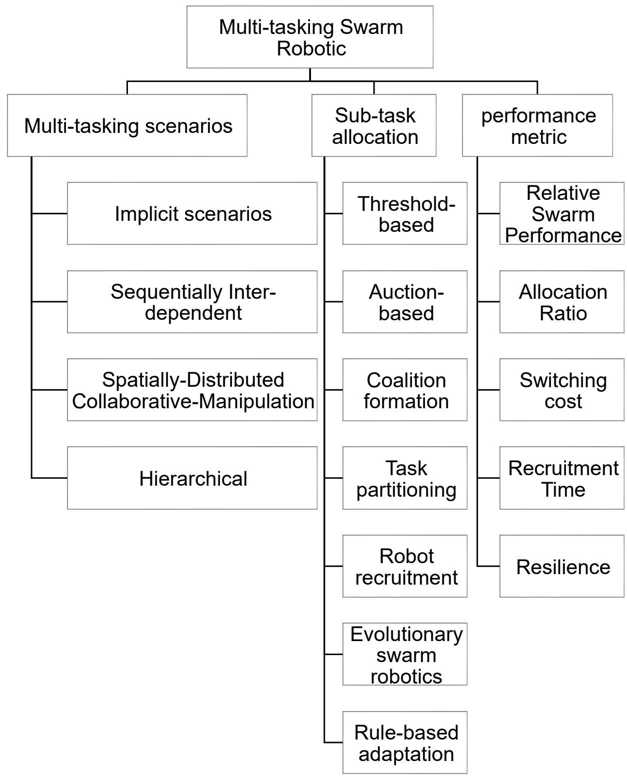

Fig. 1. Taxonomies proposed in this survey.

quality of the literature review, which was then enhanced further. Only papers written in English were considered for inclusion in the search. We scanned journals, book chapters, and conference proceedings. Finally, we assessed the suitability of the gathered papers in order to decide whether or not the articles fit the inclusion requirements. We checked the abstracts of the selected research to make sure the material was accurate. A study was eliminated if it did not meet our search criteria or if it was published more than once by the same author(s). Using this final stage, we were able to compile a list of articles for inclusion in our survey.

In this survey we introduce three taxonomies for multi-tasking scenarios, multi-tasking stages and performance metrics. See Figure 1 for a visual overview of the structure of the survey. In Section 4, we analyse application scenarios with multi-tasking requirements. In Section 5, we analyse methods to design and analyse swarm robotics systems for multi-tasking applications. In Section 6 we examine performance metrics for multi-tasking swarms.

4 MULTI-TASKING SCENARIOS FOR SWARMS

The aim of this section is to comprehend the existing research at the junction of swarm robotics and multi-tasking. Multi-tasking swarming is defined as the ability of a swarm of robots to perform multiple tasks simultaneously or in a coordinated manner. It necessitates the coordination and cooperation of multiple agents in the swarm to attain various objectives, such as exploration, search and rescue, environmental monitoring, or surveillance. The achievement of multi-tasking swarming often requires the use of advanced algorithms and communication protocols to ensure the effective and efficient collaboration of the agents within the swarm. The primary objective is to optimise the use of resources and reduce the time required to complete multiple tasks. Two categories of such work are identified: implicit multi-tasking and explicit multi-tasking scenarios that are either sequentially or spatially dependent. These are further discussed in the following subsections.

4.1 Implicit Multi-tasking Scenarios

Implicit multi-tasking refers to scenarios where a swarm performs multiple tasks simultaneously or in a coordinated way without explicit instruction and/or without explicit coordination or communication among robots in the swarm. In other words, the swarm is able to perform multiple tasks based on its programming and without specific commands. Implicit multi-tasking occurs in many seemingly single task scenarios. For example, if swarm members have a choice to respond to the given task or remain idle.

Examples of scenarios that can be considered as implicit multi-tasking include but not limited to: collective foraging, navigation and sensing, and environmental monitoring.

4.1.1 Remains Idle as Secondary Task. In many scenarios, a robot may appear to be performing a single task, but there may be an implicit secondary task that is not immediately apparent. For example, in a swarm of robots, each robot may have a choice to either respond to the given task or remain idle. Even though the explicit task is to respond to the task, there is an implicit secondary task of deciding whether or not to respond. This is an example of implicit multi-tasking, where the robot is performing multiple tasks simultaneously, even though only one task is explicitly stated.

Existing literature has noted that acknowledging the implied secondary task of remaining idle is important for optimising energy expenditure [16, 122]. In the example above, if the robots always respond to the task, they may consume more energy than necessary, as some tasks may not require a response from all robots. By considering the implicit secondary task of remaining idle, the robots can optimise their energy expenditure by only responding to tasks that require a response. This is an example of task prioritisation, where the robots prioritise the tasks based on their importance and urgency. Efficient energy expenditure is important in robotics, as it directly affects the robot's ability to perform its tasks and its overall lifespan. By acknowledging the implicit secondary tasks and prioritising the tasks accordingly, robots can optimise their energy expenditure and perform their tasks more efficiently. This is especially important in scenarios where the robots have limited energy resources, such as in remote environments or in scenarios where the robots are required to operate for extended periods of time without recharging.

Let us define the following variables and parameters:

- N_a: the number of agents (robots) in the swarm
- N_t: the number of explicit tasks to be executed in the environment
- e: time in discrete intervals
- c_e: the cost per time interval of each robot performing Task 2 (explicit task)
- p_e: the profit per time interval of each robot performing Task 2
- q_e: the profit per time interval of each robot performing Task 1 (remaining idle)

The swarm of robots must decide whether to perform Task 1 or Task 2 at each time interval e, subject to the following constraints:

- Each robot can perform only one task at a time, i.e., it cannot perform both Task 1 and Task 2 simultaneously.
- At any given time, the total number of robots performing Task 2 cannot exceed a certain limit M (i.e., the maximum number of robots that can search and bring items back at the same time).

- The objective is to maximise the total profit while minimising the total cost, over a certain time horizon E. Thus, we want to find the optimal allocation of robots to Tasks 1 and 2 at each time interval e.

Let $x_{n,e}$ be a binary variable that indicates whether robot n is performing Task 1 (if $x_{n,e} = 0$) or Task 2 (if $x_{n,e} = 1$) at time interval e. Then, the optimisation problem can be formulated as:

$$\begin{aligned} \text{Maximize} \quad & \sum_{e=1}^{E} \left(\sum_{n=1}^{N_a} p_e x_{n,e} - \sum_{n=1}^{N_a} c_e x_{n,e} + \sum_{n=1}^{N_a} q_e (1 - x_{n,e}) \right) \\ \text{Subject to} \quad & \sum_{n=1}^{N_a} x_{n,e} \leq N_t, \forall e \\ & x_{n,e} \in 0, 1, \forall n, e \end{aligned} \tag{1}$$

The objective function represents the total profit minus the total cost over the time horizon E. The first term in the objective function represents the profit earned by all robots performing Task 2 at time e, the second term represents the cost incurred by all robots performing Task 2 at time e, and the third term represents the profit earned by all robots performing Task 1 at time e. The constraint ensures that the total number of robots performing Task 2 at any given time e does not exceed N_t. The decision variables are binary, indicating whether each robot is performing Task 1 or Task 2 at each time interval e.

Collective Multi-Foraging. One example of multi-tasking scenarios that incorporate the choice of remaining idle is multi-foraging. Foraging is the most common testbed application for swarm robotics systems. It is a collaborative task inspired by ant colony behaviour. In an artificial swarm robotics system, a certain location is designed to be the "nest", and the aim of the swarm is to find scattered objects and bring them back to the nest [16]. Multi-foraging is a variation of the standard foraging where various items of different types must be transported to their respective assigned nests. Despite its simplicity, foraging may be viewed as an abstraction with numerous analogies to more complex scenarios including waste cleanup, demining, and search and rescue. It is widely used as a test-bed for collective exploration, transportation, and decision-making. Additionally, it is also used to examine the influence of interference in swarm robotics systems [24]. From a multi-tasking perspective, the foraging task can be partitioned into two sub-tasks: 1) remaining idle; and 2) searching the environment and bringing items back to the nest. Several studies are conducted to study the dynamics of swarm energy arising from foraging activity [122]. For example, in comparison to food sources, items retrieved by the robots to the nest contribute energy to the swarm, whereas search effort results in energy loss. Each robot's control algorithms should decide when the robot goes foraging and when it sits about doing nothing. This helps maximise net energy efficiency and improves foraging performance [16, 122].

4.1.2 Simultaneous Execution of Sub-tasks. In many swarm robotics scenarios, robots are required to perform multiple sub-tasks simultaneously to achieve their objectives efficiently (i.e. navigation and sensing, and obstacle avoidance). Such sub-tasks can be performed concurrently and implicitly, without explicit communication or coordination among the robots. The concept of simultaneous execution of sub-tasks is crucial for efficient swarm behaviour, as it allows each robot to allocate resources dynamically and optimally among different sub-tasks, based on its local information and objectives. Examples include navigation and sensing [13], and environmental monitoring [81].

Let t_1 and t_2 denote two sub-tasks. For each robot r_n, let $s_1(r_n)$ and $s_2(r_n)$ represent the accuracy of task t_1 and t_2 respectively. Let $c_1(r_n)$ and $c_2(r_n)$ denote the costs associated with performing tasks t_1 and t_2, respectively. Let y_n denote the allocation of resources (time/effort) that robot r_n dedicates to task t_1 and z_n denote the allocation of resources that robot r_n dedicates to task t_2, where $0 \leq y_n + z_n \leq 1$.

Then, the objective is to maximise the accuracy of both tasks while minimising the overall cost, which can be formulated as follows:

$$\max \sum_{n=1}^{N_a} \omega_1 \cdot (s_1(r_n) - c_1(r_n)) + \omega_2 \cdot (s_2(r_n) - c_2(r_n)) \tag{2}$$

where the weights ω_1 and ω_2 represent the relative importance of the two tasks.

Navigation and Sensing. In many navigation scenarios, robots need to simultaneously build a map of the environment while estimating their own position within that map. Simultaneous localisation and mapping (SLAM) [13] is a classic example of a multi-tasking scenario in robotics. In SLAM, robots use sensor measurements, such as laser range finders or cameras, to simultaneously build a map of the environment and estimate their own position within that map. SLAM requires the robot to perform two tasks simultaneously: mapping the environment and self-localisation. However, robots are not explicitly given a task to perform, but rather need to implicitly allocate their resources to both mapping and localisation to achieve the objective while ensuring that the tasks are not in conflict with each other.

Environment Monitoring. Environment monitoring refers to the detection and analysis of the surrounding environment to identify potential hazards and/or obstacles. Environment monitoring is often performed simultaneously with other tasks by robots in order to ensure safe and efficient transportation of objects in various environments. For example, a robot may have to detect obstacles in its environment and manipulate objects around them. Another example is a robot detecting and avoiding obstacles on the floor while also picking up and moving objects that are located near the obstacles. A robot may have to avoid obstacles and plan a path to its goal simultaneously. For example, a robot designed for exploration in unknown environments may have to avoid obstacles while simultaneously planning its path to cover as much area as possible [81]. In this scenario, the robot is implicitly coordinating between the different tasks related to object transport and environment monitoring. This coordination is achieved through the robot's control system, which is designed to handle multiple tasks and prioritise them based on their importance and urgency. The robot's control system may use sensor data to detect obstacles, hazards, or other objects in the environment, and adjust its path or manipulation strategy accordingly, while also taking into account its other objectives and constraints.

4.2 Sequentially Inter-dependent Tasks

Social insects, which have been seen to be capable of decomposing huge tasks into smaller ones, are natural examples of organisms that use interdependent sub-tasks [127]. Sequential interdependency is the most prevalent type of interdependency: Sub-tasks must be completed sequentially to complete the larger task. The term highlights the fact that the sub-tasks are not only performed in a specific order, but also have inter-dependencies that require careful coordination and management.

The leaf-cutter ant, which collects leaves by cutting them from trees and returning them back to the nest after they have already been chopped, is an example of an interdependent sub-task [57]. Because the cutting of the leaves is required before they can be transported, there is a need for sequential dependence. The swarm faces a task allocation dilemma in order to optimise performance: how should swarm members divide up the various tasks? Self-organisation is used by social insects to deal with challenges like this one of task allocation (see [127] for examples of such strategies).

Consider for example, a swarm of robots tasked with cleaning a contaminated area. The task involves the following sub-tasks that are inter-dependent:

(1) Deploy the swarm of robots in the contaminated area
(2) Identify the location and extent of the contamination using sensors on the robots
(3) Determine the type of contaminant and the appropriate cleaning method
(4) Dispatch specialised robots to apply the cleaning method to the contaminated area
(5) Monitor the cleaned area to ensure that the contaminant has been completely removed
(6) Decontaminate the specialised robots before returning them to the swarm

In this scenario, the sub-tasks are inter-dependent on one another, as the identification of the contaminant and the appropriate cleaning method must be completed before the specialised robots can be dispatched to apply the cleaning method. Additionally, the monitoring of the cleaned area is dependent on the successful application of the cleaning method, and the decontamination of the specialised robots is necessary to prevent the contamination from spreading to other areas. The successful completion of each sub-task is dependent on the successful completion of the previous sub-task, making this an example of sequential inter-dependent tasks in swarm robotics.

Sequential inter-dependence in artificial swarm systems can be formulated as follows: suppose that $T = t_1, t_2, \ldots, t_{N_t}$ be a set of N_t sub-tasks that are sequentially inter-dependent, such that t_m must be completed before t_{m+1} can start, for $m = 1, \ldots, N_t - 1$. Let $R = r_1, r_2, \ldots, r_{N_a}$ be a set of N_a robots available to work on the sub-tasks. Each sub-task t_m requires a certain number of robots, denoted by q_m, where $q_m > 0$. Let x_{mn} be a binary decision variable that indicates whether robot r_n is assigned to sub-task t_m. That is, $x_{mn} = 1$ if robot r_n is assigned to sub-task t_m, and $x_{mn} = 0$ otherwise. Then, we have the following constraints:

- Each sub-task t_m must be completed before t_{m+1} can start, for $m = 1, \ldots, N_t - 1$. This can be expressed as:

$$\sum_{n=1}^{N_a} x_{mn} \geq q_m, \quad \text{for } m = 1, \ldots, N_t - 1 \tag{3}$$

- Each robot can only work on one sub-task at a time. This can be expressed as:

$$\sum_{m=1}^{N_a} x_{mn} \leq 1, \quad \text{for } n = 1, \ldots, N_a \tag{4}$$

- If a robot is assigned to a subsequent sub-task (t_m with $m > 1$) and not the first sub-task (t_1), it must remain idle until all previous sub-tasks are completed.

The objective function can be to minimise the time/energy/cost to complete all sub-tasks. This can be a linear or nonlinear function depending on the specifics of the problem. Equation 5 shows an example objective function that minimises the total cost of completing all sub-tasks, where the cost of each sub-task is multiplied by a binary decision variable that determines which robot is assigned to it.

$$\text{minimize} \quad \sum_{m=1}^{N_t} \sum_{n=1}^{N_a} x_{mn} \cdot c_m \tag{5}$$

where c_m is the cost associated with completing sub-task m.

Constrained foraging. The foraging task can be partitioned into two sequential sub-tasks: robots searching for items in the environment and robots transporting items to the nest. Because items must be harvested before they can be stored, the two sub-tasks have a sequential interdependence: robots engaged in the storing sub-task rely on robots engaged in the preceding harvesting sub-task.

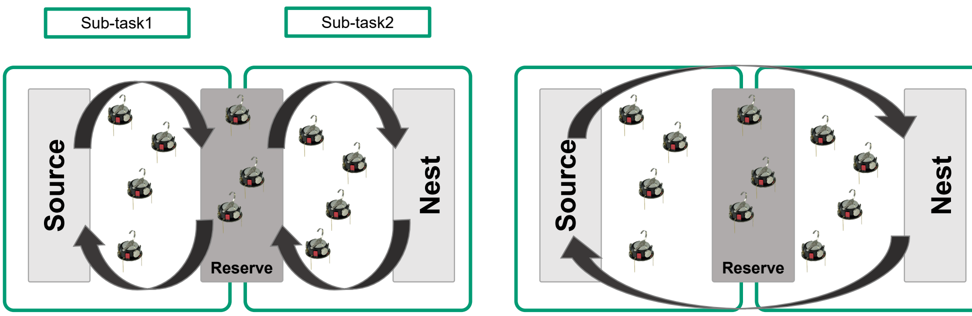

Fig. 2. Tackling foraging task with two groups of robots (left) and a single group of robots (right).

There are two approaches to this scenario [93]. One method is to make the robots collaborate as a single entity. When a robot approaches the task interface in this method, it immediately shifts to the other sub-task. With the second technique, we may solve the foraging problem by segmenting the swarm into two groups of robots, with each group responsible for completing one of the two sub-tasks in turn. An important decision is required by robots upon arriving to the task interface: they can either continue working on the current sub-task or move onto a new one. If they elect to continue working on the same sub-task, they must transfer the transported object to a robot executing the other sub-task, see Figure 2. Mechanisms of robot cooperation can make it easier for a group of robots to perform each of these sub-tasks. Additionally, cooperation is advantageous in reducing the negative impacts of robot interference and therefore increasing the system's scalability.

The application of sequentially interdependent multi-tasking systems to complex foraging problems has been studied in the literature. For example, [132] consider a foraging task with heterogeneous swarm members which specialise in specific task (**foragers**: go to the nectar source, load their crop with nectar, and then fly back to the entrance of the nest **storers**: wait for returning foragers. The returning foragers' harvest loads are transported to the storage facility by these workers. They leave their nectar load there, and then go back to the entrance and start again, **unemployed** bees travel indiscriminately throughout the hive, whilst **larvae** are confined to patches in the central nest region and do not move at all. When nectar supplies are low, they release a chemical hunger signal, and **nurses**: navigate (uphill) via the chemical stimulation provided by hungry larvae swarms. As soon as they come into contact with a patch that contains a hungry larva, they begin feeding the larva until the nurse is nearly depleted. In this way, an efficient division of labour scheme has been developed in the honeybee colony to address foraging efficiently. During Brutschyet's study of foraging [26], the author separated the task into two sub-tasks, harvesting and storing, which are linked in a sequential manner. A harvesting robot carries an item from the source zone to the task interface area, where it awaits the completion of another sub-task by another robot. It is subsequently transferred to that robot, which moves it to its destination within the nest.

4.3 Spatially-Distributed Collaborative-Manipulation Tasks

Collaborative manipulation is a term used to describe operations performed by swarm robots, in which objects in the environment are manipulated by groups of robots collaborating together. To put it simply, swarm systems enable agents to work together to do collective tasks more quickly and effectively than each agent could do individually. Additionally, in certain cases the task at hand cannot be completed by a single individual and requires the collaboration of numerous individuals

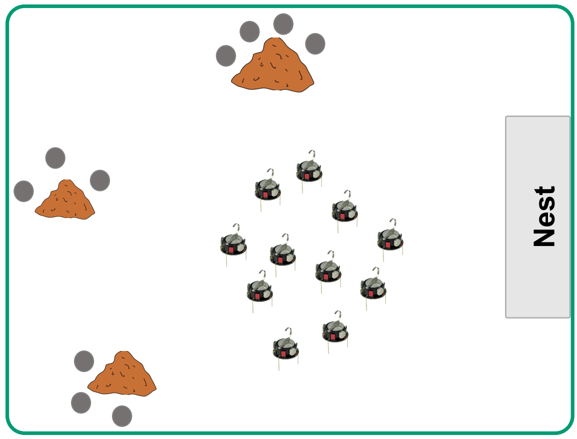

Fig. 3. Illustration of collaborative manipulation task. Each member of the swarm must decide which task to perform given task minimum requirement.

to be completed. The recovery of large food items by groups of ants is a classic example borrowed from insect societies: depending on the size of the object, this activity may need the participation of a large number of ants, all of whom must collaborate in order to execute the work effectively (i.e. transport the food item to the nest).

Suppose there is a swarm of robots $R = \left\{r_1, r_2, ..., r_{N_a}\right\}$, and a set of a spatially distributed tasks $T = \left\{t_1, t_2, ..., t_{N_t}\right\}$, to be carried out collectively by the robots. Each robot r_n has its own capability (ability to perform a given task) and capacity (i.e. enough energy to complete the task), and each task t_m has its own requirements in terms of the minimum workload (number of agents) and positions. Task t_m is deemed completed if the total work capacities of the agents assigned to it reach its minimum requirement. The goal is therefore to distribute robots as evenly as feasible in relation to the requirements of different tasks in order to fulfil all tasks in a timely way while minimising total energy consumption (i.e. avoid collisions and congestion). Other limits in the environment, such as reaching task positions within a specific time limit, may need to be met. Such a deadline may be critical to the success of urgent tasks, such as military applications or a search and rescue mission. This spatially-distributed collaborative-manipulation scenario is briefly illustrated in Figure 3.

The problem can be formulated as follows: Let x_{mn} be a binary decision variable which takes a value of 1 if robot r_n is assigned to task t_m, and 0 otherwise. Let c_n and p_n denote the capability and capacity of robot r_n, respectively. Let w_m denote the minimum workload (number of agents) required for task t_m to be completed, and let p_{mn} denote the position required for robot r_n to perform task t_m. The objective can be formulated as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{n=1}^{N_a} p_n x_{mn} \\
\text{subject to} \quad & \sum_{n=1}^{N_a} x_{mn} \geq w_m \quad \forall m = 1, \dots, N_t, \\
& \sum_{m=1}^{N_t} c_n x_{mn} \leq p_n \quad \forall n = 1, \dots, N_a, and \\
& x_{mn} \in 0, 1 \quad \forall n = 1, \dots, N_a, m = 1, \dots, N_t
\end{aligned}
\tag{6}
$$

The objective function minimises the total energy consumption, which is the sum of the positions (distances) of the assigned robots. The first constraint ensures that each task is completed by assigning at least the minimum required number of agents to it. The second constraint ensures that each robot is assigned tasks that it has the capability and capacity to perform. The decision variable x_{mn} indicates whether robot r_n is assigned to task t_m or not.

Robotic swarm decision-making challenges in this scenario include [74, 103, 157]:

(1) Organising agents into task-specific groups and evenly aligning and dividing their work capabilities and capacities in proportion to task requirements.
(2) Choosing work positions that fulfil the bare minimum of criteria for each task and maximise the working capacities of the agents by reducing travel expenses while avoiding collisions with other agents or obstacles and utilising positions with higher work efficiency.

Some applications of multi-tasking systems have been studied extensively in the literature. For example, dynamic task allocation in a dual patrolling system is proposed by [62, 63] to solve the issue of specialisation among the individual agents. To differentiate themselves from one other, the individual robots have developed distinct communication signals. The suggested system evolves groups of equal individuals into separate or specialised groups, with each individual robot distinguishing itself and identifying its task based on the signals that they can transmit. Some robots, for example, increase the power of their communication signals, while others decrease the power of their signals. As a consequence, based on the power level of the output communicative signal, the robot team is divided into two specialised groups.

Swarm Distribution Guidance. In the swarm distribution guidance, a swarm of robots must be divided into different subgroups or partitions, such that each partition satisfies certain requirements or constraints, such as a desired population percentage or swarm density. Each partition corresponds to a different bin, area, or task that the swarm must complete [5, 6]. Figure 4 shows an example of swarm distribution into certain bins. The problem is challenging because the robots must be allocated to the partitions in a way that minimises the overall cost, such as the total distance travelled or the time required, while still satisfying the constraints. In addition, the problem can be further complicated by the presence of obstacles, limited sensing and communication capabilities, and the need to coordinate the movements of multiple robots. Solutions to the swarm distribution problem have many practical applications, such as in warehouse automation, where robots are assigned to different areas to pick and transport items, or in environmental monitoring, where robots are allocated to different regions to collect data on different environmental variables [6, 76].

Collective Search. In the case of collective search, there is a trade-off between properly searching and covering as much area as possible. Collective search problems are now a hot topic in robotics, with researchers focusing their efforts on finding solutions through the use of distributed algorithms

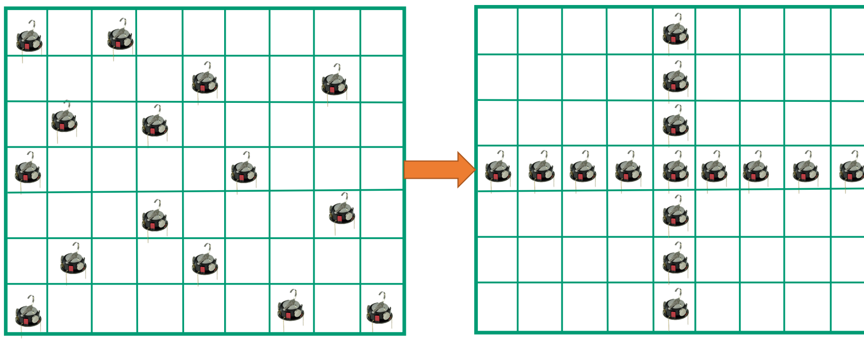

Fig. 4. Illustration of swarm distribution guidance problem (distributing homogeneous robots into bins with a specified population distribution for each bin).

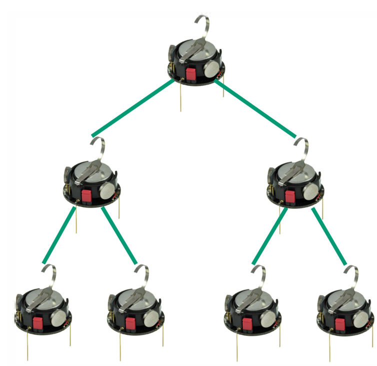

Fig. 5. Illustration of hierarchical topology of responsibility scenario. Each layer of the swarm system is responsible for a certain task.

applied to this problem. The goal is to devise methods for robots to execute search and rescue operations without relying on centralised control. When dealing with unstructured environments, such as post-disaster and dangerous settings, where direct communication is limited, the difficulty of coordinating a team of robots for exploration can be extremely difficult. Applications of collective search has been explored in the literature such as [159] who develop task allocation method inspired by division of labour in a wasp swarm to provide collective searching and retrieval of heterogeneous targets. The method is based on the response threshold model and the goal is to split the robots into sub-groups specialised in searching for certain type of targets.

Heterogeneous Sub-tasks. There are situations in which it is necessary to do several distinct types of tasks at the same time. Each of those tasks has its own requirements [20]. Given a swarm of robots with different capabilities, then a challenge arises as to how to assign different robots to different types of tasks so that the final goal is achieved. The authors of [20] study a distributed task allocation problem allocating unmanned aerial vehicles (UAVs) with different capabilities to one of three targeting tasks: detection, attack and verification.

4.4 Hierarchical Self-Organisation

Traditionally, most swarms operate in a "flat mode" in which all of the robots are regarded to be on the same hierarchical level regardless of whether they interact with each other. A hierarchical swarm topology, on the other hand, is based on the concept of having many hierarchical layers inside a single functioning swarm [9, 86, 94].

Much research in sociology, management science, and psychology has examined hierarchical self-organisation, which focuses not only on how to build an organisation but also on how to develop connections between individuals and improve the organisation's efficiency [94, 121]. Individuals in a hierarchical organisation have strong connections and well defined subordinate relationships, allowing the entire system to successfully manage complex tasks with appropriate planning.

Using a hierarchical swarm topology, the swarm is divided into levels, with each level having a superior and subordinates at progressively lower levels. This hierarchical responsibility scenario is briefly illustrated in Figure 5. This is how a complex task is decomposed into smaller tasks. The root node is the original task, the internal nodes are intermediate sub-tasks that can be broken down further, and the leaf nodes are the simplest sub-tasks that should be solved directly. Hierarchical organisation of swarms is particularly important in scenarios with problems that are deemed too complex or not efficient enough to be solved with a "flat" swarm design. Examples include area coverage, aggregation problems, and group control.

Hierarchical Self-Organising Swarm (HSOS) can be formulated as follows: Let us consider a swarm of N_a robots that can operate in two modes: flat mode and hierarchical mode. In the flat mode, all robots are considered to be on the same hierarchical level, while in the hierarchical mode, there are multiple hierarchical layers inside a single functioning swarm.

The hierarchical swarm topology is based on the concept of having many hierarchical levels, with each level having a superior and subordinates at progressively lower levels. Let us assume that the swarm is divided into L levels. Each level l has a set of robots R_l such that $\bigcup_{l=1}^{L} R_l = 1, 2, \ldots, N_a$. The root node of the hierarchy represents supervisory level , while the internal nodes represent intermediate supervisory/execution levels. The leaf nodes represent execution levels (robots that perform given sub-tasks) .

Let T be the set of all tasks that the swarm needs to accomplish. Each task $t \in T$ has a set of sub-tasks B_t that need to be performed to complete the task. Let L_t be the level of the hierarchy at which the task t is assigned. Then, we can represent the hierarchical swarm topology as a directed acyclic graph (DAG) $G = (V, E)$, where V is the set of nodes in the graph, and E is the set of directed edges. The nodes in the graph represent the sub-tasks, and the edges represent the dependencies between the sub-tasks. Let us assume that each robot r in the swarm has a set of skills S_r that it can perform. Each sub-task $b \in B_t$ requires a set of skills S_b to be performed. A robot r can perform a sub-task b if $S_b \subseteq S_r$. To accomplish a main task t, the swarm needs to find a set of robots that can perform all the sub-tasks $b \in B_t$. The objective is to minimise the time required to accomplish all the tasks. Let x_{br} be a binary decision variable that indicates whether robot r performs sub-task b. The objective can be formulated as shown in Equation 7:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{t \in T} \sum_{b \in B_t} \sum_{r \in R_{L_t}} x_{tbr} \\
\text{Subject to} \quad & \sum_{r \in R_{L_t}} x_{br} \geq 1 \quad \forall t \in T, b \in B_t \\
& \sum_{b \in B_r} x_{br} \leq 1 \quad \forall r \in R
\end{aligned}
\tag{7}
$$

The objective function minimises the total time required to accomplish all the sub-tasks. The first constraint ensures that at least one robot is assigned to each sub-task. The second constraint ensures that each robot is assigned to at most one sub-task. The third constraint ensures that the decision variables are binary.

Area Coverage. The area coverage problem in swarm robotics refers to the challenge of deploying a team of robots to cover a given area, such as a field, building, or outdoor environment. The goal of the problem is to achieve complete coverage of the area by the robots, either to gather information about the environment or to perform a task [38]. Hierarchical topology may be seen in action when a set of robots attempts to fully uncover an unknown or known territory through the process of area coverage. The robot residing at the top level can determine the size of the region to be covered and assign portions to lower levels, which can subsequently be redistributed until the entire area has been covered and returned back up the hierarchy to the top level [8].

Robot Aggregation. Another example of hierarchical responsibilities can be seen in aggregation problems where randomly placed robots in an environment are aggregated by another set of robots with higher rank or with different capabilities. For instance, in the context of unmanned aerial and ground vehicles (UAVs and UGVs), a hierarchical topology can be used to aggregate UGVs in a given area through the use of UAVs. [114, 139].

Group Control. The motion planning of a group of robots using an external entity is defined as group control [147]. This approach has many real-world applications in different domains including crowd control [98]; oil spill cleanups [107]; aerospace protection [58]; disaster relief and rescue operations [137]; wildlife protection [123]; and the handling of microorganisms [117].

The problem is complex because the behaviour of each agent affects the behaviour of the entire group, and the group's behaviour is influenced by a wide range of factors, including the environment, the actions of other agents, and the overall goal of the system. The goal of group control is to develop a set of control strategies and algorithms that enable the agents to work together in a coordinated and efficient manner. This involves breaking down the overall problem into smaller sub-tasks that can be more easily managed by the individual agents or sub-groups of agents, and developing communication and decision-making protocols that allow the agents to exchange information and coordinate their behaviour.

5 MULTI-TASKING STAGES

Multi-tasking allows social insects to complete difficult tasks by decomposing them into smaller sub-tasks that are carried out by distinct subgroups of individuals [12, 19, 46, 127]. Although multi-tasking has significant benefits for adaptation in social insects, the evolution of the multi-tasking process is not quite clear. The reason for this is that the ability to multitask requires the co-occurrence of numerous complex mechanisms, including decomposition of complex tasks into simpler sub-tasks, allocation of appropriate number of individuals to each sub-task, the ability of individuals to effectively execute each of the sub-tasks, and coordination of the execution of these sub-tasks [45, 54]. The requirement for a flexible division of labour in order to adapt to changing environmental conditions complicates this problem [45, 54]. Figure 6 summarises the stages required for multi-tasking.

5.1 Task Decomposition

Task decomposition is the process through which complex tasks are broken down into smaller and manageable sub-tasks [77]. Task decomposition is observed in social insects (i.e. beetles) in tasks such as transportation of material [127], foraging [134], hunting [131], nest excavation [11],

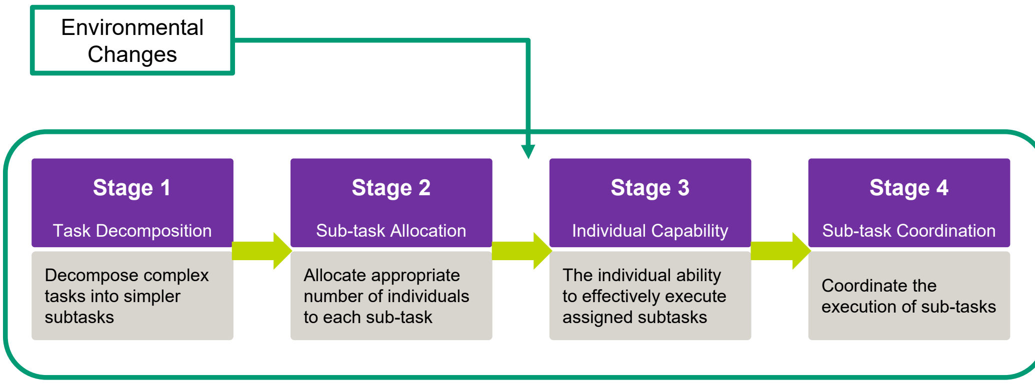

Fig. 6. Stages required for multi-tasking.

and rubbish disposal [66]. In swarm robotics, robots capable of autonomously decomposing tasks would be immensely adaptive, since they could tailor their task execution to unique settings and purposes. However, most swarming models in the literature assume that task decomposition is made by a human expert prior to deploying a swarm in given environments. Very few studies attempt to study the autonomous task decomposition by swarm robots.

Pini et al. [125, 126] propose an approach for autonomous task decomposition in foraging tasks based on a cost function. This cost function is a mapping between the quantity of work done by a robot and the total cost of executing the whole task.

Duarte et al. [47] propose a semi-autonomous approach for task decomposition where controllers are evolved for robots to solve a certain task. Should evolution fail to identify a viable controller for a given situation, the task is manually split into two or more smaller sub-tasks, with each sub-task evolving its own separate sub-controller. Next, an extra controller is developed that allows the user to pick the particular sub-controller to be active at any given time.

5.2 Task Allocation and Selection

The purpose of task allocation in a swarm robotic system is to have each individual robot choose appropriate tasks or actions at each point in time based on its local sensations, ensuring that system-level goal is completed [51, 61, 108, 161]. Only local communication is possible, and knowledge of the environment is also quite limited. There are two important aspects of task allocation: (1) How do individuals obtain information? (2) How are decisions based on such information made?

5.2.1 Threshold-based Model. The response threshold model (RTM) has been used to address task allocation in decentralised systems for several problems. A probabilistic approach is used in the majority of RTM-based systems [30, 32, 79, 82, 150]. It is common practise to estimate this probability as a function of the agent's task threshold (i.e. the current task stimulus value, and other variables). Probabilistic decision-making is used in both systems where all agents have the same threshold for a certain task and systems where different agents have different thresholds for the identical task. Using probabilistic rather than deterministic approach in these systems stem from historical as well as practical reasons. As a result of the non-determinism inherent in natural systems, this method has its historical roots in the division of labour model seen in insect communities [21]. Practically, the spectrum of swarm agents' behaviours is expanded via a probabilistic decision-making process. Deterministic decision-making states that if all agents are exposed to the identical stimuli, they will react the same way if they all have the same thresholds. Occasionally, because of probabilistic decision-making, some agents fail to act when their task threshold indicates that they ought to. Increased diversity in agent behaviour expands the swarm's ability to respond to different situations. As a result, a swarm is more likely to be able to cover all of the tasks that require attendance.

Fixed Response Threshold Model The fixed response threshold model (FRTM) is one of the most prominent theories for explaining decentralised and flexible task allocation in social insects. FRTM is a straightforward mathematical model that uses fixed response thresholds to regulate task allocation in insect societies. Suppose s represents the intensity of a stimulus linked with a specific task; s might be any quantitative signal detected by swarm members. A robot's inclination to respond to the stimulus s and accomplish the accompanying task is determined by a response threshold θ, which is an internal variable represented in units of stimulus intensity. The variable θ is set in such a way that $s \ll \theta$ and $s \gg \theta$ have lower and higher response probabilities, respectively. Equation 8 represents one family of response functions $R_\theta(s)$ that may be parameterised with thresholds that satisfy this criterion (It is worth mentioning that other forms of response functions have been used in the literature such as exponential response functions [118]). Figure 8 shows several response curves, where $n > 1$ determines the steepness of the threshold, for different values of θ. For $s \ll \theta$, the likelihood of engaging task performance is close to 0, while for $s \gg \theta$, it is close to 1. This likelihood is exactly 0.5 for $s = \theta$. As a result, those with lower θ values are more likely to respond to lesser levels of stimuli. The response curve becomes steeper as the value of n increases. If $n \to \infty$, then equation 7 illustrates how a sigmoid function may be used to represent the response function. For a given s and n , if $\theta_1 < \theta_2$, then $R_{\theta_1}(s) > R_{\theta_2}(s)$. As a result, when presented with the identical task-related stimuli, The task is more likely to be completed by individuals who have a low θ value compared to those who have a high θ value. According to previous research in the field, high values of θ can reduce interference among robots and thus improve system performance, whereas low values of θ can improve adaptation in response to rapid changes in the environment [32].

$$R_\theta(s) = \frac{s^n}{s^n + \theta} \tag{8}$$

$$lim_{n\to\infty}(R_\theta(s)) = lim_{n\to\infty}\left(\frac{s^n}{s^n+\theta}\right) = \begin{cases} 0 & s \le 0 \\ 0 & s > 0 \end{cases} \tag{9}$$

It has been demonstrated that FRTM is capable of adapting quickly to changing environmental circumstances[155]. Despite its simplicity, FRTM has shown similar behaviours to the mechanisms that regulate labour in insect colonies such as worker specialisation [91, 155]. However, FRTM has the following three assumptions:

(1) Agents (swarm members) are capable of assessing task demands in response to specific cues (stimuli) that drive task performance (the model does not address the nature and interpretation of these stimuli).
(2) Each agent is likely to experience all stimuli at some point within a given time period and can in theory respond to these stimuli.
(3) Response thresholds are not time-dependent (i.e. remain constant over time).

Adaptive Response Threshold Model Several approaches are proposed in the literature to alleviate the limitations of FRTM. The Adaptive Response Threshold Model (ARTM) [30–32] is an extension of the classical FRTM proposed to alleviate the limitations of FRTM where the response threshold θ is adjusted according to the external stimulus instead of having it fixed.

Rather than keeping a task stimulus fixed, Pang et al [119] followed another approach to adapt task stimulus according to environmental conditions such as number of other agents performing

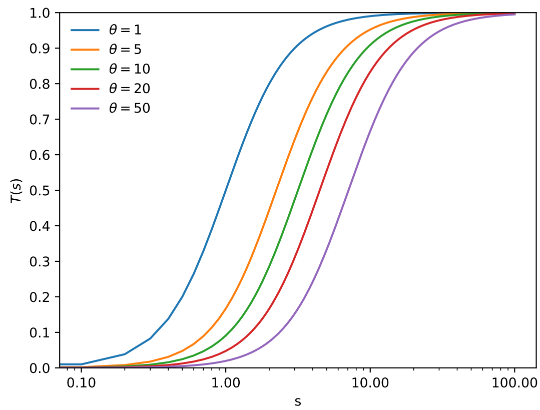

Fig. 7. Semi-logarithmic plot of threshold response curves ($n = 2$) with different thresholds ($\theta = 1, 5, 10, 20, 50$)

the task, and the number of obstacles avoided in the previous foraging task. The proposed approach referred to as the Dynamical Response Threshold Model (DRTM) is governed by two factors: 1) internal response threshold; and 2) external stimulus. The external stimulus $S(e)$ is defined as shown in Equation 10.

$$S(e) = D(e_0) - D(e) \tag{10}$$

where $D(e)$ denotes the task demand at time e and $D(e_0)$ is the optimal task demand to be maintained. Stimulus $S(e)$ measures the gap between the existing task demand and the optimal task demand. When current task demand decreases, the stimulus $S(e)$ increases correspondingly and in order to achieve the desired task demands, the engaging probability of the agent increases. The response threshold is calculated dynamically according to Equation 11.

$$\theta(t) = \alpha(N_a M_a) + \beta m f(t) \tag{11}$$

where N_a corresponds to the total number of agents in the environment and M_a is the number of agents not currently engaged with the task. Parameter m is the historical number of obstacle avoidances in the last round of doing the task which may reflect the traffic condition. $f(t)$ is a time function that represents the time interval between successive task engagement rounds. α and β are determined by trial and error. The probability of an agent a_i to engage in a given task t_j is defined as shown in Equation 12.

$$p_{ij} = \begin{cases} 0 & \text{If } S^t \leq 0 \\ \frac{(S^t)^n}{(S^t)^n + \theta} & \text{Otherwise} \end{cases} \tag{12}$$

Reinforced Threshold Various works [39, 156] propose application of some reinforcement learning strategies to θ, thereby changing the sensitivity of the robot in real-time. The robot has a reinforcement learning (RL) controller and its learning is triggered using a tunable response threshold.

Given a set of N_t tasks $T = \{t_1, .., t_{N_t}\}$, and a set of N_a robots $A = \{a_1, .., a_{N_a}\}$, each robot a_i selects a task t_j to perform at each time instance e with a probability $p_{ij}(e)$ such that $\sum_{i=1}^{N_a} p_{ij}(e) = 1$

Initially, these probabilities are initialised equally as shown in Equation 13:

$$p_{ij}\left(0\right)=\frac{1}{N_t} \tag{13}$$

In the following step, a learning process is initiated, in which each robot changes its selection probabilities in accordance with a reinforcement learning update rule. In [39], an update rule is proposed based on learning automata where the probability of robot a_i selecting a task t_j to perform at time instance $e+1$ is defined as shown in Equation 14.

$$p_{ij}\left(e+1\right)=p_{ij}\left(e\right)+\beta R_{ij}\left(e\right)\left[1-p_{ij}\left(e\right)\right] \tag{14}$$

where β is the learning rate and R_{ij} is a binary reward signal received from the environment calculated as shown in Equation 15. $R_{ij}=1$ when the number of robots W_j performing task t_j is less than or equal to the minimum number of robots required to complete the task L_j, otherwise the reward signal receives a reward $R_{ij}=0$

$$R_{ij}=\begin{cases} 1 & \text{If } \frac{W_j}{L_j}\leq 1 \\ 0 & \text{Otherwise} \end{cases} \tag{15}$$

The action of the automata is evaluated by the environment at each time step e. The action is considered "favorable" if the environment yields a response value of 1 for the action (task selection), and "unfavorable" if the environment yields a response value of 0. Such a reinforcement learning approach has been shown to be more effective than the traditional fixed threshold models. However, it is more prone to errors as the environmental noise increases [39].

5.2.2 Rule-based Task Selection. In contrast with the threshold-based approaches, the robots in rule based task selection approach decides to perform certain tasks based on adaptation rules that use information sensed locally from the environment. Those rules can be pre-specified or evolved through a learning process. Liu et al. [101] proposes a rule-based task selection approach to optimise net efficiency of a swarm. A foraging scenario is employed, in which a number of food-items are randomly scattered across the arena. A swarm of robots searches for and retrieves food items, which they then return to the nest. In addition, each food item collected supplies an amount of energy to the swarm, but the process of foraging itself consumes a certain amount of energy at the same time. To stay alive, the swarm needs to find as much food as possible over time. In order for the swarm to get the most energy from its food, the robots have to switch between foraging and resting. This is because there is limited amount of food in the environment. Adaptation criteria are utilised that are based on three sorts of stimuli: internal cues (successful food retrieval), external cues (collisions with team members while seeking for food), and social cues (team-mate success in food retrieval).

5.2.3 Auction-based Task Allocation. The agents in auction-based algorithms employ a negotiation mechanism to bid on tasks in an auction. They utilise their local information of the environment to calculate how many resources (bid) they should provide to the task. The agents bid based on the utility or cost they calculate, with the goal of achieving the highest utility or the lowest cost for the task assigned. The auctioneer may be a central agent, or the auction may be completed in a distributed manner by system agents. The auction may take several rounds to finish and can be done for one or more tasks simultaneously. Based on the utility functions of the agents, a global objective function is optimised [41, 120, 143, 145].

The auction based task allocation is highly efficient in terms of reaching optimal or near optimal solutions. They are also scalable thanks to their moderate computational cost and communication burden. Additionally, auction based approaches are flexible since they can accommodate newly added/removed tasks from the auction procedure [85, 143].

5.2.4 Coalition Formation. In scenarios with multiple tasks that are spatially distributed, the agents have to partition themselves into task-specific coalitions (or groups). A coalition can be defined as a group of agents that have resolved to collaborate together to accomplish a common task. Each coalition focuses on a single task at a time, while each agent may be a member of several coalitions at any given time [73, 138]. This latter assumption has the potential to improve an agent's capacity to utilise its resources for task completion. In this manner, task allocation is viewed as a problem of assigning groups of agents to tasks. The key challenge becomes the division of the agents into subgroups.

A capability vector B^C represents the sum of each member's individual contributions to the coalition C. A value V may be derived for each coalition C, which is the combined utility (can also be considered as cost) that the members of C can attain through collaboration to complete a certain task. The capabilities that coalition members contribute, the sequence in which tasks are completed, and the number of coalition members all have an impact on the coalition utility value V. In the event of overlapping coalitions, a strategy is needed by which the agents determine how to allocate their capabilities among the coalitions they engage in.

Given a set of N_t tasks $T = \left\{t_1, .., t_{N_t}\right\}$, and a set of N_a agents $A = \left\{a_1, .., a_{N_a}\right\}$ with their capabilities, the problem can be formulated as assigning tasks $t_i \in T$ to coalitions of agents $C_i \subseteq N$ such that $\sum V_i$ is maximal.

Coalition formation refers to techniques that enable robots to form task-specific coalitions (groups), assign a task to each group, and choose work positions in such a way that each task's minimum requirement is met while also optimising the agents' work capacities through reduced travel costs and utilisation of more efficient works positions [74, 138].

Several approaches are proposed in the literature to solve the coalition formation problem. Shehory et al [138] formulated the task coalition problem as set partitioning and set covering problems and used combinatorial and optimisation algorithms to solve the problem. On the other hand, [74**?**] propose a game-theoretic approach to divide the agents into disjoint task-specific coalitions in a way that balances the agents' capabilities in proportion to the task's minimum workload requirements.

[102] study multi-task cooperation problem for unmanned aerial vehicle (UAV) swarms with energy consumption constraints. In this study, certain tasks may need the cooperation of several UAVs with specific capabilities. The authors follow a coalition game theory approach to address the coalition formation and task selection problems.

5.2.5 Task Partitioning. Task partitioning is an advanced form of division of labour that is prevalent in insect societies in which two sets of tasks are executed sequentially by different individuals [54]. Task allocation and task partitioning are closely related topics that examine robot collaboration. In contrast to task allocation which examines robot dynamics while working on distinct sub-tasks, task partitioning focuses on how the task itself is divided into sub-tasks. When applied to a complex task, task partitioning benefits the robotic swarm on both the individual and collective levels. The partitioned sub-tasks are easier for the robots to complete on an individual level. Decomposing a given task into simpler sub-tasks lessens the burden of management at the collective level, allowing task allocation techniques to be used to reduce switching costs and increase the swarm's net efficiency [148]. When complex tasks have characteristics that may be leveraged to reduce switching costs while simultaneously increasing the net efficiency of the group, task partitioning is the preferable method of operation. In this case, there is a better likelihood of securing an appropriate combination of task specialists if the behavioural representations for each sub-task are available as ready-made building blocks. Traditionally, task partitioning is performed by a human designer with domain expertise [48, 125, 126]. While some progress has been made in automatic task partitioning, the findings are limited to specific tasks, and a generic task partitioning guideline is still lacking [126, 148].

5.2.6 Robot Recruitment. Robot recruitment is a type of task allocation that aims to design a low-cost coordination mechanism capable of forming groups of robots at certain locations in response to events in the environment (i.e. the discovery of targets or food) [141]. When a robot identifies a target, but lacks the resources to deal with it, it can operate as a strong attractor for other robots to join a coalition and work collaboratively to deal with it. The identification of a target can occur at any point in time throughout the exploration of the environment, hence the recruiting process occurs in real time and in different locations of the environment.

One approach for robot recruitment is a greedy approach, in which an identified target is immediately assigned to the robots regardless of future events [90]. Each robot makes its own judgements, which may lead to it withdrawing from recruitment requests. For example, when on the route to a given target or receiving another request, it is possible to detect another target or receive another request, and therefore alter decisions following a selfish decision making process.

Advanced and more adaptive approaches are proposed in the literature that are inspired by social insects. These approaches can be generally categorised into two groups based on the recruitment request mechanism used: 1) Non-propagating, 2) propagating approaches.

In the non-propagating approaches, recruiters announce the detected targets and only local robots are informed about the targets and no propagation of information is done. For robot recruitment, three bio-inspired meta heuristics have been proposed in the literature. These are as follows: 1) Artificial Bee Colony (ABC), based on honey bees [80]; 2) the Firefly-based Team Strategy (FTS) [153], an algorithm derived from swarms of fireflies; 3) Particle Swarm Optimisation (PSO), which is modelled from schools of fish and flocks of birds. Despite the bio-inspired approaches (bio-heuristics) working well, there are big costs to picking the right algorithm and setting the right parameters for each new type of swarming task [36]. Moreover, non-propagating approaches allow for only one robot to be recruited at a time.

The propagating approaches rely on an ad-hoc routing architecture for multi robots communication, in which information about the detected targets can propagate across the network of robots. A popular recruitment approach under this category is the Wave algorithm [140]. The Wave algorithm is a type of distributed algorithm in which a node known as the initiator transmits messages to its neighbours, which then send messages to their neighbours, and so on. An initiator is a process that starts its local algorithm on its own when an internal trigger reaches a certain condition. A non-initiator begins its local algorithm when a message arrives. The computation of this distributed method is referred to as a wave, which is a succession of occurrences, each of which is a message sent or received. A decide event is a form of internal event that is distinct from other types of internal events. A wave algorithm sends and receives a specified number of messages before making a decision. A centralised wave algorithm is one in which there is a single initiator and all other nodes act as non-initiators. A wave algorithm must satisfy all three of the following requirements to be effective: 1) each computation cycle is finite; 2) each computation cycle comprises at least one decide event 3) In each computing cycle, every decide event is preceded by an event in each process [140, 142]. One or more arbitrary leaders initiate robot recruiting where each wave should produce a group of robots. Propagating recruitment can also be implemented using nature-inspired approaches, for example, in [40] a strategy is proposed using pheromones to attract other robots to a mine location for collaborative task completion. The robot that detects the mine becomes the coordinator and deposits a specific type of pheromone that guides other robots to the mine cell. The pheromone evaporates following the same rules as previously explained. When a robot detects this pheromone, it uses a formula to choose the next cell to move to.

5.2.7 Swarm Shepherding. Studies of animal behaviour reveal how swarm-like behaviour may be maintained by a diverse group of social animals [128]. The reaction of a sheep flock to a herding dog is a famous illustration of this. Swarm control mechanisms such as shepherding were inspired by sheepdogs and sheep.

Shepherding is considered a promising technique in swarm robotics for multi-tasking because it allows the swarm to perform multiple tasks simultaneously while maintaining coordination and cohesion. It involves using a specialised robot, called a shepherd, to direct and guide a group of other robots, called a swarm. In a shepherding scenario, the shepherd robot acts as a task manager and assigns specific tasks to individual members of the swarm. For example, some robots might be assigned to explore a new area, while others might be tasked with gathering resources or building structures. The shepherd coordinates the swarm's activities, ensuring that each robot is working on the appropriate task and that the overall mission is being accomplished.

Shepherding can also be used to reconfigure the swarm dynamically in response to changes in the environment or task requirements. For example, if a new resource becomes available, the shepherd can reassign robots to collect it. Or, if a robot becomes damaged or malfunctions, the shepherd can adjust the swarm's behaviour to compensate for the loss.

Overall, shepherding is a promising solution for multi-tasking in swarm robotics because it provides a way to coordinate the activities of a large group of robots, allowing them to work together efficiently and effectively.

A particular example where shepherding approach was investigated is swarm navigation. In this particular scenario, the shepherding problem is defined as leading a swarm of agents from their beginning location to their desired position. As a result, the shepherd is primarily responsible for the high-level path planning and work distribution of the swarm, whereas the swarm members are mainly concerned with single agent dynamics [99]. For direction and control applications, shepherding offers a clear design of both artificial sheep and sheepdog robots that are simple to construct. Sheep agents are repulsive robots that react to fundamental forces. A prior knowledge of the goal or the course that the sheep robots will travel is not required for the robots. The sheepdog robot (or robots) determine their position by following a series of predetermined rules, and they then push the sheep to respond and move to the desired locations [52].

5.2.8 Evolutionary Swarm Robotics. Evolutionary Swarm robotics (ESR) is a branch of robotics that builds autonomous robots using evolutionary computation [48, 160]. Artificial evolution is commonly employed for behavioural control [111], but it may also be utilised to develop robot formations [33, 67]. An advantage of ESR techniques is that they are able to automate control system design without relying on significant and time-consuming manual specification of desired behaviour [55]. An evolutionary process follows a fitness gradient [56]: A population of candidate solutions is evaluated, and the ones that score higher on fitness become the parents of the following generation. Depending on the procedure utilised, the next generation is fed mutant forms (small variants) of its parents, as well as mixtures of two or more parents.

ESR is a promising approach to multi-tasking scenarios, and there are several models in the literature in which it is applied to solve task allocation/selection problem in swarm robotics. For example, in [97], the authors propose an enhanced algorithm for multi-robot task allocation and path planning (MRTA-PP) in complex environments. The MRTA problem is transformed into the multiple travelling salesman problem (MTSP), and an improved A* algorithm (IA*) is used to solve the distance matrix consisting of the distances between task points. An improved genetic algorithm is proposed to solve the MTSP based on the distance matrix. Finally, the overall route of each robot is planned using the solution obtained by the IA*.

In [152], the authors propose the use of swarm intelligence optimisation algorithms for target searching tasks in swarm robotics, specifically focusing on the challenges presented by multiple target scenarios. The paper introduces the Brain Storm Optimisation (BSO) algorithm, which has shown promise in solving multi-modal optimisation problems, and proposes a BSO-based collaborative searching framework for swarm robotics called Robotic BSO. Simulation results demonstrate the potential effectiveness of this approach for multi-target searching problems in swarm robotics.

In [151], the authors discuss the evolution capability of the swarm robots' strategy and present a heterogeneous-homogeneous swarm co-evolution method called TORCH to improve it. The method uses a swarm co-evolution mechanism and a behaviour expression tree to expand the strategy search space. TORCH makes swarm robots' strategies evolve under local information conditions, making them more adaptable to distributed task scenarios. The paper reports extensive experiments to verify TORCH, showing its superiority in terms of evolutionary efficiency improvement and strategy performance enhancement compared to three other methods.

In [106], the authors discuss the use of automated guided vehicles (AGVs) in Industry 4.0 and investigate an optimisation model and a solution using a decentralised multi-agent approach for scheduling problems that involves conflicting products and AGV capacity constraints in flexible production systems. This study showed that the decentralised approach using ESR is capable of providing efficient and effective solutions to complex multi-tasking problems

5.3 Research Platforms

This section presents research platforms summarised in Table 2 designed for research purposes and were used to study the application of swarm algorithms, in general and multi-tasking capabilities in particular, to robotic swarms.

NVIDIA Isaac Sim.]colorredNVIDIA Isaac Sim [116] is a robotics simulation platform developed by NVIDIA for testing and validating autonomous robot systems. It is part of the NVIDIA Isaac platform, which also includes hardware and software components for building and deploying autonomous robots. Isaac Sim enables the simulation of various robotic platforms in a virtual environment, including ground and aerial vehicles, manipulators, and sensors. The platform provides a high-fidelity physics engine and realistic sensor models, allowing developers to test and validate their algorithms in a simulated environment before deploying them on physical robots. The platform is built on top of the Unity game engine and supports a range of programming languages, including Python and C++. It also includes integration with various popular robotics frameworks, such as Robot Operating System (ROS) and the Isaac Software Development Kit (SDK), allowing developers to seamlessly transfer their algorithms from simulation to the physical robot.

Gazebo. Gazebo [88] is a general-purpose open source robotics simulation platform developed and maintained by the Open-Source Robotics Foundation (OSRF). It offers graphical user interfaces for visualising the scenario as well as interfaces to four different physics engines for rigid-body dynamics modelling. Gazebo offers an interface to the Robot Operating System (ROS) [2], which has become a defacto standard in robotics research.

Webots. Webots [109] is a robotics simulation platform that lets users design, build, and test mobile robots in a rapid prototyping environment. Control programmes can be transferred to a number of commercially available real mobile robots using the software libraries.

Table 2. Research simulation platforms for (multi-tasking) swarm robotics.

Application/ product name	**Description**	**Web link**
Gazebo	General-purpose 3D simulator	http://gazebosim.org/
Webots	Mobile robotics simulation software	http://www.cyberbotics.com/
Evorobot	A simulator developed for running evolutionary robotics experiments	http://laral.istc.cnr.it/evorobot/simulator.html
ARGoS	Discrete-time physics based simulation	https://www.argos-sim.info/
USARSim/ROS	ROS-enabled robot simulation platform	https://sourceforge.net/projects/usarsim
V-Rep	A virtual robot experimentation platform	http://www.coppeliarobotics.com
Box2D	Open-source Two-dimensional physics simulator engine	http://box2d.org
Jasmine III	Open-Source Micro-Robot platform	http://www.swarmrobot.org/
NetLogo	A multi-agent modelling and simulation environment	http://ccl.northwestern.edu/netlogo/
Airsim	High fidelity physical and visual simulator	https://github.com/Microsoft/AirSim/
SwarmLab	Matlab based simulation for swarming with focus on drones	https://github.com/lis-epfl/swarmlab/
NVIDIA Isaac Sim	Robotics simulation platform developed by NVIDIA for testing and validating autonomous robot systems	https://developer.nvidia.com/isaac-sim
The Robotarium project	Hardware platform that can be remotely utilised by anyone, and is available free of charge.	https://https://www.robotarium.gatech.edu/

Evorobot. Evorobot [115] is a software package that allows users to design and run evolutionary swarm robotics. Experiments may be carried out in simulation and/or on a real robot, thanks to the Khepera robot, which was developed at the Laboratory of Microprocessors and Interfaces, Swiss Federal Institute of Technology in Lausanne [64]. The software is written in C and C++.

ARGoS. ARGoS [124] is an open source discrete-time physics based simulation framework that is publicly accessible. ARGoS can simulate large-scale swarms of robots of various kinds (i.e. e-puck) at varying degrees of detail [25]. It is possible to further customise ARGoS with the addition of new plugins.

USARSim/ROS. Unified System for Automation and Robot Simulation (USARSim) [14] is an open-source high-fidelity swarm robotic simulator that is based on the Unreal Developers Kit (UDK) [4] developed by Epic Games.

Virtual Robot Experimentation Platform. Virtual Robot Experimentation Platform (V-Rep) [129] is a simulation platform that incorporates various models of research as well as industrial robots. It also allows for the creation of customised robots and can simulate real-world physics and object interactions. V-REP has a flexible design that allows it to accommodate a wide range of functionalities that may be activated or deactivated as needed. This makes it a good balance between high-fidelity simulation capabilities and simulation completeness.

Box2D. Box2D [1] is a freely available open source two-dimensional simulator engine built in C++. The engine has been exported to a wide selection of additional programming languages including Java, C,and python. The engine is often used in games and performs rigid body simulation. It is capable of simulating bodies made out of circles, polygons, and edge forms. Bodies are connected together by joints and are subjected to the influence of forces. Friction, gravity, and restitution are example forces that are included in the engine's library of effects.

Jasmine. Jasmine [3] is a 3D platform developed for large-scale swarm robotic research using sensors including touch, proximity, distance, and colour.

NetLogo. NetLogo [149] is a programming language and integrated development environment that is available as open source software (IDE). An extensive model collection contains models from a variety of disciplines, including economics as well as biology and chemistry.

Airsim. Airsim [136] is an open-source high fidelity physical and visual simulator written in C++. Airsim offers a large library of vehicle and sensor models (i.e. GPS, gyroscope, accelerometer, magnetometer, and barometer).

SwarmLab. SwarmLab [144] is an open-source application built in Matlab that provides defined methods and metrics to assess the performance and robustness of swarm algorithms, with an emphasis on drones.

Robotarium project. The Robotarium project [60] offers an accessible swarm robotics research platform that can be remotely utilized by anyone, and is available free of charge.

6 PERFORMANCE METRICS

Swarm intelligence literature offers a diverse range of indicators and metrics for the operation of swarming systems. Collective motion, for examples, is difficult to define in theory, but indicators exist to help identify when it is occurring. These include 'group' and 'order' [54, 146] metrics. Other metrics may pertain to performance at specific tasks (e. g. see Section 6.1) or to aspects of multi-tasking such as switching cost and swarm diversity discussed later in Section 6.

6.1 Relative Swarm Performance

Relative swarm performance is calculated as the ratio of swarm performance P_{Na} relative to maximal (optimal performance) possible P_{max} as shown in Equation 16.

$$P_R = \frac{P_{N_a}}{P_{max}} \tag{16}$$

Brutschy et al. [26] used relative swarm performance to compare swarm performance in a foraging task with and without task partitioning strategy. The authors defined P_{max} to be the maximum number of objects that can be harvested by the swarm.

In multi-tasking scenarios, the relative swarm performance metric can be used to evaluate the efficiency of a swarm in accomplishing multiple tasks simultaneously. For instance, assume that a multi-tasking swarm is designed to complete three different tasks: task A, task B, and task C. The swarm's overall performance is measured in terms of the completion time for each task. To evaluate the swarm's relative performance, the completion time for each task is normalised with respect to the optimal completion time for that task. The optimal completion time for each task is obtained by assuming that the swarm is dedicated exclusively to that task. Then, the relative swarm performance metric can be calculated as the ratio of the actual completion time for all tasks to the sum of the optimal completion times for all tasks. If the ratio is less than 1, it indicates that the swarm is performing better than the optimal individual swarms for each task. Conversely, if the ratio is greater than 1, it implies that the swarm is under-performing and has room for improvement.

Overall, the relative swarm performance metric can help in identifying the efficiency of a multi-tasking swarm and provide insights into how to optimise the allocation of resources and coordination mechanisms within the swarm to improve its overall performance.

6.2 Allocation Ratio

In a multi-tasking scenario, the allocation ratio metric can be used to evaluate how effectively the available robots are allocated to different sub-tasks. For a given sub-task t_i, AR^i_{alloc} is the fraction of robots working on the t_i and calculated as $AR^i_{alloc} = \frac{N^i_a}{N_a}$. Quality of allocation is then evaluated by measuring the deviation from an optimal allocation ratio AR^{i*}_{alloc} of robots as reference. This optimal allocation ratio represents the ideal allocation of robots to sub-tasks, and it can be obtained through analytical or optimisation methods that consider the characteristics of the tasks and the capabilities of the robots. In [26], the authors use the mean absolute error MAE across all sub-tasks to evaluate this quality of allocation as shown in Equation 17.

$$MAE_{AR} = \frac{1}{N_t} \sum_{i=1}^{N_t} \left| AR^{i*}_{alloc} - AR^i_{alloc} \right| \tag{17}$$

By measuring the deviation from the optimal allocation ratio, the allocation ratio metric can provide insights into how well the available robots are being allocated to the different sub-tasks. If the deviation is small, it indicates that the robots are being allocated efficiently, and the allocation ratio metric is high. Conversely, if the deviation is large, it implies that the robots are being allocated sub-optimally, and the allocation ratio metric is low.

In summary, the allocation ratio metric can be used to evaluate the quality of the allocation of robots to different sub-tasks in a multi-tasking scenario. It can help identify areas for improvement and inform decisions about how to optimise the allocation of robots to improve the overall performance of the swarm.

6.3 Switching Cost

The number of times the robots transition between sub-tasks is referred to as the "switching cost" [25, 70, 157, 158]. In a multi-tasking scenario, the switching cost metric can be used to evaluate the efficiency of task allocation and the stability of the swarm's allocation strategy. The switching cost is an important factor to consider when designing an effective strategy for task allocation/selection.

To minimise the switching cost, an effective strategy should ensure that robots are consistently assigned to specific sub-tasks and that very few robots are required to transition between sub-tasks. The the number of switches N_s can be recorded at each time step e and investigated over time to gain insights into how well the swarm converges to a stable allocation.

For instance, in [26], the authors recorded the switching cost over a pre-specified time-window w_e and analysed the evolution of the switching cost to evaluate the speed with which the swarm reaches a stable allocation. The results can help identify the factors that contribute to instability and inefficiency in the task allocation strategy and inform decisions about how to optimise the strategy.

Moreover, in [68], the authors measured the switching cost as the number of times a robot switches to a new task and proposed an evolutionary algorithm guided by an autonomous specialisation metric based on the switching cost function to evolve autonomous specialisation in swarm members. This approach could be used to optimise the task allocation strategy by minimising the switching cost and improving the stability of the swarm's allocation strategy.

In summary, the switching cost metric can be used to evaluate the efficiency and stability of the task allocation strategy in a multi-tasking scenario. It can help identify the factors that contribute to instability and inefficiency and inform decisions about how to optimise the task allocation strategy to improve the overall performance of the swarm.

6.4 Swarm Diversity

Swarms may be comprised of identical units (sometimes called homogeneous swarms) or similar but subtly different units (sometimes called heterogeneous swarms). The use of different units can permit specialisation at a lower cost. For example, if a small number of units carry costly sensors or actuators. Specialisation can be costly if all the specialised units are lost and the swarm can no longer adapt. Brutschy et al. discuss the costs and benefits of behavioural specialisation [27].

In a multi-tasking scenario, the metric of behavioural specialisation can be used to evaluate the benefits and costs of using identical or heterogeneous units within a swarm.

In the case of a homogeneous swarm, all the units have the same capabilities, which may limit the swarm's ability to perform multiple tasks simultaneously. In contrast, heterogeneous swarms consist of units with different capabilities, which can allow for specialisation and improve the swarm's performance in multi-tasking scenarios.

The behavioural specialisation metric is used to measure the degree to which the units in the swarm are specialised to perform specific tasks. For instance, some units may be specialised in sensing or actuation, while others may be specialised in communication or navigation. The metric can be used to determine the optimal level of specialisation for the swarm, balancing the benefits of specialised units against the costs of losing them.

As discussed in [27], the costs of behavioural specialisation include the risk of losing specialised units and the reduced flexibility of the swarm when faced with new tasks or environmental conditions. Therefore, the level of specialisation should be carefully balanced to maximise the swarm's overall performance while minimising the risk of failure.

In summary, the metric of behavioural specialisation can be used to evaluate the benefits and costs of using identical or heterogeneous units in a swarm and to determine the optimal level of specialisation to achieve the best performance in a multi-tasking scenario. It provides a tool for optimising the design and performance of swarms in complex and dynamic environments.

6.5 Recruitment Time

Recruitment time is a metric that can be used to evaluate the efficiency of the recruitment process. It measures the time it takes for the recruitment process to complete, from the start of the local broadcast of recruiting messages to the receipt of the last feedback message. Silva et al. [142] measure the average recruitment time across all initiators in their simulation to evaluate the scalability of their proposed robot recruitment strategy when the swarm size increases.

In a multi-tasking scenario, recruitment time can be used to evaluate the efficiency of the recruitment process for each sub-task. A shorter recruitment time would indicate a more efficient and effective recruitment strategy, which could lead to faster completion of multiple tasks. Therefore, in a multi-tasking scenario, the recruitment time metric can be used to evaluate the efficiency and scalability of the robot recruitment strategy. Furthermore, the recruitment time metric can also be used to evaluate the scalability of the robot recruitment strategy. As the swarm size increases, the recruitment process may become slower and less efficient. By measuring the average recruitment time across all initiators, the scalability of the recruitment strategy can be evaluated and optimised to handle larger swarm sizes.

In summary, recruitment time is an important metric for evaluating the efficiency and scalability of robot recruitment strategies in multi-tasking scenarios.

6.6 Resilience

Resilience refers to the flexibility of a swarm to adapt to changing task demands or shortage in robot availability in real time [17, 65, 92].

In multi-tasking scenarios, resilience can be used as a metric to evaluate the ability of a swarm to handle changes in task demands or robot availability in real-time. The time it takes for the swarm to reach a task distribution equilibrium can be measured and used to assess the resilience of the system. For example, if a sub-task requires more robots to complete it, the swarm should be able to allocate additional robots to that sub-task in a timely manner.

One way to measure the distribution equilibrium is through the allocation equilibrium proposed in [92]. This approach can measure the adaptive response of the swarm to changing sub-task demands in real-time and can be used to adjust robot response rates for different sub-tasks. Another approach, as shown in the work of Jang et al. [**?**], is to use Hellinger Distance to measure the similarity between the current robot distribution and a desired optimal distribution as shown in Equation 18 where θ_a^j is the optimal proportion of robots required to perform task t_j and $\frac{N_a^j}{N_a}$ is the current proportion of robot distribution. This can help in identifying any imbalances in the distribution of robots and can be used to adjust the allocation of robots to different sub-tasks accordingly.

$$HD = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{N_t} \left(\sqrt{\theta_a^j} - \sqrt{\frac{N_a^j}{N_a}} \right)^2} \tag{18}$$

In summary, resilience can be an important metric in multi-tasking scenarios as it evaluates the ability of a swarm to adapt to changing task demands and maintain a balanced distribution of robots.

7 CHALLENGES, FUTURE DIRECTIONS AND CONCLUSIONS

Swarm robots have a wide range of potential applications, including exploration and surveillance as well as search and rescue, mining, intrusion tracking, cleaning, inspection, and transportation of large items. Although they have the potential to be resilient, scalable, and versatile, swarm robotics systems have not yet been put to use in real-world applications and have instead been restricted to academic investigations. The current status of the swarm robotics area is mostly concerned with acquiring desirable collective behaviours and analysing their characteristics. In this case, researchers often use simplified testbed applications to avoid difficulties that may arise in real-world applications. There are many possible reasons for the absence of swarm robotics in the real world, we envision an increasing need for: Validation and assurance, task decomposition and flexible task allocation protocols.

7.1 Validation and assurance

A swarm robotics system's validation and comparison to other systems remain difficult despite the wide variety of analytical methods proposed in the literature. This is due to the lack of testbed applications and well-defined metrics. Metrics are usually closely linked to specific scenarios and hence cannot be used for other systems or for comparisons. Metrics that can capture overall system efficacy and efficiency is a possible way to compare different systems. The average idle time across robots, total time to complete the whole task are just two examples.

7.2 Task Specialisation Trade-off

In order to optimise their efficiency and take full advantage of the increased performance, the robots in the swarm should focus their efforts on a single type of tasks. However, despite the obvious advantages of exploiting the performance boost achieved through learning, specialisation also has certain drawbacks such as inefficiency in dynamically changing environments [27, 69, 127]. The time a specialised robot spends looking for a suitable task is one example of these inefficiencies. For example, a robot that is highly specialised will focus on a particular task at a time. In contrast, an unspecialised robot may perform whatever task it is presented with. In this situation, the specialised robot may spend more time than the unspecialised robot searching for a suitable task, depending on the amount of tasks of each type available in the environment. As a result, when tasks of the sort in which they are occur infrequently, specialised robots may be less efficient than unspecialised robots. This means that in contexts where the number of tasks and their spatial distribution regularly fluctuate, specialisation may be less desirable.

7.3 Lack of Association between Tasks and Stimuli

Large portions of the work presented in the literature adopt threshold-based strategy to regulate division of labour in swarms. The following assumption underlie most of these models: Every task has a stimulus, and completing a task reduces its associated stimulus [10, 22]. Thus, a robot's perception of an internal or external (in the environment) stimulus activates the execution of the related task, diminishing the intensity of the signal lowers the likelihood of repeating the task. Consequently, the perceived stimulus may be considered as a representation of the task's demands. In certain circumstances, however, a task's performance may have no influence on the amount of the associated perceived stimulus, or there may be no connection at all between a task and a stimulus. For example, when the nature of the task is unknown in advance or might vary over time [87]. As a result, new approaches are required to drive the task allocation process so that robots can autonomously determine whether or not to continue their current task in the absence of task stimuli.

7.4 Automatic task decomposition

Automatic task decomposition: Almost all swarming models in the literature assume that task decomposition is made by a human expert prior to deploying a swarm in given environments. However, in real-world applications, it is challenging to identify and decompose complex tasks in advance. An automatic and adaptive approach is in need to achieve this stage for multi-tasking swarms to be able to adapt to complex and dynamic environments. Lessons learned from reinforcement learning domain can be utilised here to achieve this goal. Particularly, the goal-oriented reinforcement learning (GORL) approach [71, 95, 112], is an interesting and promising way for dealing with intricate reinforcement learning problems including long-term credit assignments and sparse rewards. GoRL's main premise is to create intermediate goals (sub-goals) that break down the original complex task into a series of simpler intermediate sub-tasks, each of which is driven by a goal. A similar approach can be adopted for task decomposition (or partitioning) in swarming systems for multi-tasking applications.

7.5 Closed-loop Multi-tasking Frameworks for Swarm Robots

In addition to the algorithmic challenges outlined above, the gap between simulation and reality for swarm robotics remains. For the creation and testing of autonomous behaviours, current simulation environments employ empirical/probabilistic sensor models. These sensor models may be acceptable for robots meant to operate in man-made settings free of complications (vegetation, mud, etc). However, when operating in a natural environment, these sensor models cannot reliably anticipate robot behaviour. Additionally, assumptions made by robotics experts and algorithms experts are still very different, so it is not trivial to put state of the art multi tasking algorithms on real physical systems.

There is a need for an interface between complex algorithmic design and implementation and hardware robotic programming. In real robots, factors such as sensor noise and communication bandwidth become significant issues. Modes of communication can be significantly limited in genuinely distributed systems compared to simulations where a centralised computer manages communication. High fidelity simulation models provide a possible approach to addressing the shortcomings of simplified simulation environments by capturing the complex interactions of the robots' sensors with complex environments [50].

7.6 Conclusion

The self-organised swarm behaviours utilised in swarm robotics systems are inspired by natural swarm systems such as insect colonies or bird flocks, which can adapt to a broad range of dynamic conditions. Natural-inspired robotics provide a number of advantages, including the capacity to adapt to a dynamic environment. They are designed to function in a physical environment that is always changing and necessitates the capacity to deal with unpredictable conditions and external influences. Aside from the enormous potential for applications in industries like as logistics, agriculture, and inspection, the use of swarms in environments that are inaccessible to humans, such as those that are hazardous, unclean, or difficult to reach is extremely favourable and desired. These settings might be used to better exploit and leverage the advantages of swarm characteristics, such as flexibility, resilience and scalability.

The literature on the current state of multi-tasking swarm robotics has been presented in this survey paper. The different scenarios for multi-tasking environments for robotic swarms have been presented with their multi-tasking structure and requirements have been explained. The different approaches presented in the literature that can be used to control a swarm in an environment with multiple tasks have been presented. The research issues discussed in this survey paper span scenarios with different multi-tasking requirements, and different approaches in the literature for sub-task allocation/selection, offering the readers a broad perspective on multi-tasking swarm robotics. In addition, we surveyed and presented the performance metrics suitable for multi-tasking swarming systems.

The development of closed-loop swarming designs that will give swarms of robots with high degrees of autonomy and flexibility while responding to a dynamic environment with multi-tasking needs is an important field of further research. We believe that such designs will eventually lead to increased efficiency and efficacy, as well as long-term autonomy. A framework for high-level planning and decision making (i.e. automatic task decomposition and a feedback loop about the uncertainties in the environment to alter its operations) must be in place, but in a distributed way. The function of high-level planning and decision making in conjunction with low-level swarm management and estimate systems in multi-tasking swarm systems with highly unpredictable environments should be described using the qualities of stability, convergence, and resilience. To maximise scalability and computing efficiency, distinct parts of swarm decision-making, control, and estimation should be available on multiple timescales and hierarchical levels. A closed-loop architecture capable of changing its task breakdown and task assignment/selection online is an example of such characterisation on stability. Another important area is to establish rigorous methodologies for task allocation/selection independent from task stimulus. Such methodologies can be used for improving system scalability since swarming design will no longer be tied to the task specification.

ACKNOWLEDGMENTS

This research has been funded by an Australian Research Council Discovery Project DP200100710

REFERENCES

[1] Box2d documentation. URL https://box2d.org/documentation/.

[2] ROS documentation. URL http://docs.ros.org/.

[3] Jasmine documentation. URL http://www.swarmrobot.org/Simulation.html.

[4] Udk documentation. URL https://docs.unrealengine.com/udk/Three/WebHome.html.

[5] Behçet Açikmeşe and David S Bayard. A Markov chain approach to probabilistic swarm guidance. In *2012 American Control Conference (ACC)*, pages 6300–6307. IEEE, 2012.

[6] Behçet Açıkmeşe and David S Bayard. Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Asian Journal of Control*, 17(4):1105–1124, 2015.

[7] Daniel Albiero, Angel Pontin Garcia, Claudio Kiyoshi Umezu, and Rodrigo Leme de Paulo. Swarm robots in mechanized agricultural operations: a review about challenges for research. *Computers and Electronics in Agriculture*, 193:106608, 2022.

[8] Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, and Yahya Zweiri. A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Applied Sciences*, 1:1–24, 2019.

[9] Nicholas Alunni, R Goloski, A Haggerty, and E Jones. Hierarchical swarm robotics. *project number GSF113, Graduate Thesis*, 2011.

[10] T. Alves, J. Rivière, C. Alaux, Y. Le Conte, F. Singhoff, T. Duval, and V. Rodin. An interruptible task allocation model: Application to a honey bee colony simulation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020.

[11] C Anderson and FLW Ratnieks. Task partitioning in insect societies: novel situations. *Insectes sociaux*, 47(2):198–199, 2000.

[12] Carl Anderson and Nigel R Franks. Teamwork in animals, robots, and humans. *Advances in the study of behavior*, 2003.

[13] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

[14] Stephen Balakirsky and Zeid Kootbally. Usarsim/ROS: A combined framework for robotic control and simulation. In *International Symposium on Flexible Automation*, volume 45110, pages 101–108. American Society of Mechanical Engineers, 2012.

[15] Kira Barton and Derek Kingston. Systematic surveillance for UAVs: A feedforward iterative learning control approach. In *2013 American Control Conference*, pages 5917–5922. IEEE, 2013.

[16] Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

[17] Spring Berman, Adám Halász, M Ani Hsieh, and Vijay Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE transactions on robotics*, 25(4):927–937, 2009.

[18] Spring Berman, Vijay Kumar, and Radhika Nagpal. Stochastic strategies for a swarm robotic assembly system. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1953–1958, 2009.

[19] Samuel N Beshers and Jennifer H Fewell. Models of division of labor in social insects. *Annual review of entomology*, 46(1):413–440, 2001.

[20] Di Bin, Zhou Rui, Wu Jiang, and Chen Shaodong. Distributed coordinated task allocation for heterogeneous UAVs based on capacities. In *2013 10th IEEE International Conference on Control and Automation (ICCA)*, pages 1927–1932. IEEE, 2013.

[21] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807, 1998.

[22] Eric Bonabeau, Marco Dorigo, and Guy Théraulaz. From natural to artificial swarm intelligence, 1999.

[23] Anindita Brahma, Souvik Mandal, and Raghavendra Gadagkar. Emergence of cooperation and division of labor in the primitively eusocial wasp ropalidia marginata. *Proceedings of the National Academy of Sciences*, 115(4):756–761, 2018.

[24] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[25] A. Brutschy, N.-L. Tran, N. Baiboun, M. Frison, G. Pini, A. Roli, M. Dorigo, and M. Birattari. Costs and benefits of behavioral specialization. *Robotics and Autonomous Systems*, 2012.

[26] A. Brutschy, G. Pini, C. Pinciroli, M. Birattari, and M. Dorigo. Self-organized task allocation to sequentially interdependent tasks in swarm robotics. *Autonomous Agents and Multi-Agent Systems*, 2014.

[27] Arne Brutschy, Nam-Luc Tran, Nadir Baiboun, Marco Frison, Giovanni Pini, Andrea Roli, Marco Dorigo, and Mauro Birattari. Costs and benefits of behavioral specialization. *Robotics and autonomous systems*, 60(11):1408–1420, 2012.

[28] Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Guy Theraula, and Eric Bonabeau. *Self-organization in biological systems*. Princeton university press, 2020.

[29] Y Uny Cao, Andrew B Kahng, and Alex S Fukunaga. Cooperative mobile robotics: Antecedents and directions. In *Robot colonies*, pages 7–27. Springer, 1997.

[30] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro. Task allocation for a robotic swarm based on an adaptive response threshold model. 2013.

[31] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro. Foraging optimization in swarm robotic systems based on an adaptive response threshold model. *Advanced Robotics*, 2014.

[32] E. Castello, T. Yamamoto, F.D. Libera, W. Liu, A.F.T. Winfield, Y. Nakamura, and H. Ishiguro. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence*, 2016.

[33] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1):11–23, 2014.

[34] Ahmad Reza Cheraghi, Sahdia Shahzad, and Kalman Graffi. Past, present, and future of swarm robotics. In Kohei Arai, editor, *Intelligent Systems and Applications*, pages 190–233, Cham, 2022. Springer International Publishing.

[35] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.

[36] Mario GCA Cimino, Domenico Minici, Manilo Monaco, Stefano Petrocchi, and Gigliola Vaglini. A hyper-heuristic methodology for coordinating swarms of robots in target search. *Computers & Electrical Engineering*, 95:107420, 2021.

[37] Jack Connor, Benjamin Champion, and Matthew A Joordens. Current algorithms, communication methods and designs for underwater swarm robotics: A review. *IEEE Sensors Journal*, 21(1):153–169, 2020.

[38] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.

[39] Javier de Lope, Darío Maravall, and Yadira Quiñonez. Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems. *Neurocomputing*, 163:47–55, 2015. ISSN 0925-2312. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems Progress in Intelligent Systems Mining Humanistic Data.

[40] Floriano De Rango, Nunzia Palmieri, Xin-She Yang, and Salvatore Marano. Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. *Soft Computing*, 22:4251–4266, 2018.

[41] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[42] Pollyanna G. Faria Dias, Mateus C. Silva, Geraldo P. Rocha Filho, Patrícia A. Vargas, Luciano P. Cota, and Gustavo Pessin. Swarm robotics: A perspective on the latest reviewed concepts and applications. *Sensors*, 21(6), 2021. ISSN 1424-8220.

[43] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[44] Marco Dorigo, Guy Theraulaz, and Vito Trianni. Swarm robotics: Past, present, and future [point of view]. *Proceedings of the IEEE*, 109(7):1152–1165, 2021.

[45] Ana Duarte, Franz J. Weissing, Ido Pen, and Laurent Keller. An evolutionary perspective on self-organized division of labor in social insects. *Annual Review of Ecology, Evolution, and Systematics*, 42(1):91–110, 2011.

[46] Ana Duarte, Ido Pen, Laurent Keller, and Franz J Weissing. Evolution of self-organized division of labor in a response threshold model. *Behavioral ecology and sociobiology*, 66(6):947–957, 2012.

[47] Miguel Duarte, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of hybrid robotic controllers for complex tasks. *Journal of Intelligent & Robotic Systems*, 78(3):463–484, 2015.
[48] Miguel Duarte, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of hybrid robotic controllers for complex tasks. *Journal of Intelligent & Robotic Systems*, 78(3):463–484, 2015.
[49] Gregory Dudek, Michael RM Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, 1996.
[50] Phillip J Durst, Christopher Goodin, Burhman Q Gates, Christopher L Cummins, Burney McKinley, Jody D Priddy, Peter Rander, and Brett Browning. The need for high-fidelity robotics sensor models. *Journal of Robotics*, 2011, 2011.
[51] V. Edwards, L. Chaimowicz, P. Rezeck, and M.A. Hsieh. Segregation of heterogeneous robotics swarms via convex optimization. 2016.
[52] Saber Elsayed, Hemant Singh, Essam Debie, Anthony Perry, Benjamin Campbell, Robert Hunjel, and Hussein Abbass. Path planning for shepherding a swarm in a cluttered environment using differential evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2194–2201. IEEE, 2020.
[53] Milan Erdelj, Enrico Natalizio, Kaushik R Chowdhury, and Ian F Akyildiz. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32, 2017.
[54] E. Ferrante, A.E. Turgut, E. Duéñez-Guzmán, M. Dorigo, and T. Wenseleers. Evolution of self-organized task specialization in robot swarms. *PLoS Computational Biology*, 2015.
[55] Dario Floreano and Laurent Keller. Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS biology*, 8(1):e1000292, 2010.
[56] Stephanie Forrest. Genetic algorithms. *ACM Computing Surveys (CSUR)*, 28(1):77–80, 1996.
[57] Harold G Fowler and SW Robinson. Foraging by atta sexdens (formicidae: Attini): seasonal patterns, caste and efficiency. *Ecological Entomology*, 4(3):239–247, 1979.
[58] Shripad Gade, Aditya A Paranjape, and Soon-Jo Chung. Herding a flock of birds approaching an airport using an unmanned aerial vehicle. In *AIAA guidance, navigation, and control conference*, page 1540, 2015.
[59] Veysel Gazi and Barış Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. In *International Workshop on Swarm Robotics*, pages 71–102. Springer, 2006.
[60] Georgia Institute of Technology. The robotarium project: A remotely accessible swarm robotics research platform, n.d. URL https://robotarium.gatech.edu/.
[61] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954, 2004.
[62] O. Gigliotta. Task allocation in evolved communicating homogeneous robots: The importance of being different. *Advances in Intelligent Systems and Computing*, 2016.
[63] O. Gigliotta. Equal but different: Task allocation in homogeneous communicating robots. *Neurocomputing*, 2018.
[64] A Guignard, E Franzi, and F Mondada. Khepera a miniature mobile robot. In *Preprints of the third International Symposium on Experimental Robotics Kioto, Japan*, 1993.
[65] Adám Halász, M Ani Hsieh, Spring Berman, and Vijay Kumar. Dynamic redistribution of a swarm of robots among multiple sites. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 2320–2325. IEEE, 2007.
[66] Adam G Hart and Francis LW Ratnieks. Task partitioning, division of labour and nest compartmentalisation collectively isolate hazardous waste in the leafcutting ant atta cephalotes. *Behavioral Ecology and Sociobiology*, 49(5): 387–392, 2001.
[67] Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28 (2):457–466, 2011.
[68] M. Hiraga, Y. Wei, T. Yasuda, and K. Ohkura. Evolving autonomous specialization in congested path formation task of robotic swarms. *Artificial Life and Robotics*, 2018.
[69] M. Hiraga, T. Yasuda, and K. Ohkura. Evolutionary acquisition of autonomous specialization in a path-formation task of a robotic swarm. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2018.
[70] M. Hiraga, Y. Tamura, and K. Ohkura. Behavioral specialization emerges from the embodiment of a robotic swarm. *Artificial Life and Robotics*, 2020.
[71] Zhiao Huang, Fangchen Liu, and Hao Su. Mapping state space using landmarks for universal goal reaching. *Advances in Neural Information Processing Systems*, 32, 2019.
[72] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. Reactivity and deliberation: a survey on multi-robot systems. In *Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, pages 9–32. Springer, 2000.
[73] I. Jang, H.-S. Shin, and A. Tsourdos. Anonymous hedonic game for task allocation in a large-scale multiple agent system. *IEEE Transactions on Robotics*, 2018.
[74] I. Jang, H.-S. Shin, A. Tsourdos, J. Jeong, S. Kim, and J. Suk. An integrated decision-making framework of a heterogeneous aerial robotic swarm for cooperative tasks with minimum requirements. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2019.

[75] Inmo Jang, Junho Jeong, Hyo-Sang Shin, Seungkeun Kim, Antonios Tsourdos, and Jinyoung Suk. Cooperative control for a flight array of UAVs and an application in radar jamming. *IFAC-PapersOnLine*, 50(1):8011–8018, 2017.

[76] Inmo Jang, Hyo-Sang Shin, and Antonios Tsourdos. Local information-based control for probabilistic swarm distribution guidance. *Swarm Intelligence*, 12(4):327–359, 2018.

[77] Robert L Jeanne. The evolution of the organization of work in social insects. *Monitore Zoologico Italiano-Italian Journal of Zoology*, 20(2):119–133, 1986.

[78] Jin-Hyung Jun, Dong-Wook Lee, and Kwee-Bo Sim. Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, volume 6, pages 614–619. IEEE, 1999.

[79] Anshul Kanakia, John Klingner, and Nikolaus Correll. A response threshold sigmoid function model for swarm robot collaboration. In *Distributed Autonomous Robotic Systems*, pages 193–206. Springer, 2016.

[80] Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1):108–132, 2009.

[81] Karthik Karur, Nitin Sharma, Chinmay Dharmatti, and Joshua E Siegel. A survey of path planning algorithms for mobile robots. *Vehicles*, 3(3):448–468, 2021.

[82] Vera A Kazakova and Annie S Wu. Specialization versus re-specialization: Effects of Hebbian learning in a dynamic environment. In *FLAIRS Conference*, pages 354–359, 2018.

[83] J Kennedy and R Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[84] Soheil Keshmiri and Shahram Payandeh. A centralized framework to multi-robots formation control: Theory and application. In *Collaborative Agents-Research and Development*, pages 85–98. Springer, 2009.

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

[86] Marius Kloetzer and Calin Belta. Temporal logic planning and control of robotic swarms by hierarchical abstractions. *IEEE Transactions on Robotics*, 23(2):320–330, 2007.

[87] Adam Klyne and Kathryn Merrick. Intrinsically motivated particle swarm optimisation applied to task allocation for workplace hazard detection. *Adaptive Behavior*, 24(4):219–236, 2016.

[88] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE, 2004.

[89] G A Korsah, A Stentz, and M B Dias. A comprehensive taxonomy for multi-robot task allocation. *International Journal of Robotics Research*, 32(12):1495–1512, 2013.

[90] M.J.B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 2000.

[91] Thomas H Labella, Marco Dorigo, and Jean-Louis Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(1):4–25, 2006.

[92] W. Lee and D. Kim. Adaptive approach to regulate task distribution in swarm robotic systems. *Swarm and Evolutionary Computation*, 2019.

[93] Wonki Lee, Neil Vaughan, and Daeeun Kim. Task allocation into a foraging task with a series of subtasks in swarm robotic system. *IEEE Access*, 8:107549–107561, 2020.

[94] Yuquan Leng, Cen Yu, Wei Zhang, Yang Zhang, Xu He, and Weijia Zhou. Task-oriented hierarchical control architecture for swarm robotic system. *Natural Computing*, 16(4):579–596, 2017.

[95] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[96] Jian Li, Shuai Guo, Yuanyuan Zhang, Deyun Chen, Wei Yu, and Baohua Zheng. Particle swarm optimization-based clustering algorithm with mobile sink for wsns. *IEEE Sensors Journal*, 16(20):7304–7314, 2016.

[97] Yanming Liang and Haiyang Zhao. An improved algorithm of multi-robot task assignment and path planning. In *Intelligent Robotics: Third China Annual Conference, CCF CIRAC 2022, Xi'an, China, December 16–18, 2022, Proceedings*, pages 71–82. Springer, 2023.

[98] Jyh-Ming Lien and Emlyn Pratt. Interactive planning for shepherd motion. In *AAAI Spring Symposium: Agents that Learn from Human Teachers*, pages 95–102, 2009.

[99] Jyh-Ming Lien, O Burchan Bayazit, Ross T Sowell, Samuel Rodriguez, and Nancy M Amato. Shepherding behaviors. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 4159–4164. IEEE, 2004.

[100] Chun Liu and Andreas Kroll. A centralized multi-robot task allocation for industrial plant inspection by using a* and genetic algorithms. In *International Conference on Artificial Intelligence and Soft Computing*, pages 466–474. Springer,

2012.

[101] Wenguo Liu, A.F.T. Winfield, Jin Sa, Jie Chen, and Lihua Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 2007.

[102] Heyu Luan, Yitao Xu, Dianxiong Liu, Zhiyong Du, Huiming Qian, Xiaodu Liu, and Xiaobing Tong. Energy efficient task cooperation for multi-UAV networks: A coalition formation game approach. *IEEE Access*, 8:149372–149384, 2020.

[103] Heyu Luan, Yitao Xu, Dianxiong Liu, Zhiyong Du, Huiming Qian, Xiaodu Liu, and Xiaobing Tong. Energy efficient task cooperation for multi-UAV networks: A coalition formation game approach. *IEEE Access*, 8:149372–149384, 2020.

[104] Nancy A Lynch. *Distributed algorithms*. Elsevier, 1996.

[105] MHA Majid, MR Arshad, and RM Mokhtar. Swarm robotics behaviors and tasks: a technical review. *Control Engineering in Robotics and Industrial Automation: Malaysian Society for Automatic Control Engineers (MACE) Technical Series 2018*, pages 99–167, 2022.

[106] Abderraouf Maoudj, Ahmed Kouider, and Anders Lyhne Christensen. The capacitated multi-agv scheduling problem with conflicting products: Model and a decentralized multi-agent approach. *Robotics and Computer-Integrated Manufacturing*, 81:102514, 2023.

[107] Ellips Masehian and Mitra Royan. Cooperative control of a multi robot flocking system for simultaneous object collection and shepherding. In *Computational Intelligence*, pages 97–114. Springer, 2015.

[108] Maja J Matarić, Gaurav S Sukhatme, and Esben H Østergaard. Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14(2):255–263, 2003.

[109] Olivier Michel. Cyberbotics ltd. webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5, 2004.

[110] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[111] Renan C Moioli, Patricia A Vargas, Fernando J Von Zuben, and Phil Husbands. Towards the evolution of an artificial homeostatic system. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 4023–4030. IEEE, 2008.

[112] Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[113] Nadia Nedjah and Luneque Silva Junior. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm and Evolutionary Computation*, 50:100565, 2019.

[114] Tung Nguyen, Hung Nguyen, Essam Debie, Kathryn Kasmarik, Matthew Garratt, and Hussein Abbass. Swarm q-learning with knowledge sharing within environments for formation control. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[115] Stefano Nolfi and Marco Mirolli. *Evolution of communication and language in embodied agents*. Springer Science & Business Media, 2009.

[116] NVIDIA. Isaac sim. https://developer.nvidia.com/isaac-sim. Accessed: February 19, 2023.

[117] Anil Özdemir, Melvin Gauci, and Roderich Groß. Shepherding with robots that do not compute. In *ECAL 2017, the Fourteenth European Conference on Artificial Life*, pages 332–339. MIT Press, 2017.

[118] Nunzia Palmieri, Xin-She Yang, Floriano De Rango, and Amilcare Francesco Santamaria. Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team. *Neurocomputing*, 306:17–36, 2018.

[119] Bao Pang, Chengjin Zhang, Yong Song, and Hongling Wang. Self-organized task allocation in swarm robotics foraging based on dynamical response threshold approach. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 256–261. IEEE, 2017.

[120] Lynne E Parker, Daniela Rus, and Gaurav S Sukhatme. Multiple mobile robot systems. In *Springer Handbook of Robotics*, pages 1335–1384. Springer, 2016.

[121] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-organization, embodiment, and biologically inspired robotics. *science*, 318(5853):1088–1093, 2007.

[122] N. Phung, M. Kubo, and H. Sato. Agreement algorithm based on a trial and error method for the best of proportions problem. *Journal of Robotics and Mechatronics*, 2019.

[123] Alyssa Pierson and Mac Schwager. Bio-inspired non-cooperative multi-robot herding. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1843–1849. IEEE, 2015.

[124] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al. Argos: a modular, multi-engine simulator for heterogeneous swarm robotics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5027–5034. IEEE, 2011.

[125] Giovanni Pini, Arne Brutschy, Marco Frison, Andrea Roli, Marco Dorigo, and Mauro Birattari. Task partitioning in swarms of robots: An adaptive method for strategy selection. *Swarm Intelligence*, 5(3):283–304, 2011.
[126] Giovanni Pini, Arne Brutschy, Carlo Pinciroli, Marco Dorigo, and Mauro Birattari. Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adaptive behavior*, 21(2):118–136, 2013.
[127] Francis LW Ratnieks and Carl Anderson. Task partitioning in insect societies. *Insectes sociaux*, 46(2):95–108, 1999.
[128] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.
[129] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE, 2013.
[130] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application.
[131] Bertrand Schatz, Jean-Paul Lachaud, and G Beugnon. Polyethism within hunters of the ponerine ant, ectatomma ruidum roger (formicidae, ponerinae). *Insectes Sociaux*, 43(2):111–118, 1996.
[132] T. Schmickl and K. Crailsheim. An individual-based model of task selection in honeybees. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008.
[133] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.
[134] Thomas D Seeley. *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press, 2009.
[135] Kamran Shafi, Kathryn E Merrick, and Essam Debie. Evolution of intrinsic motives in multi-agent simulations. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 198–207. Springer, 2012.
[136] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.
[137] Samy A Shedied. Optimal trajectory planning for the herding problem: a continuous time model. *International Journal of Machine Learning and Cybernetics*, 4(1):25–30, 2013.
[138] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101 (1-2):165–200, 1998.
[139] NE Shlyakhov, IV Vatamaniuk, and AL Ronzhin. Survey of methods and algorithms of robot swarm aggregation. In *Journal of Physics: Conference Series*, volume 803, page 012146. IOP Publishing, 2017.
[140] Jr. Silva, L. and N. Nedjah. Wave algorithm for recruitment in swarm robotics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.
[141] Jr. Silva, L. and N. Nedjah. *Distributed algorithms for recruitment and coordinated motion in swarm robotic systems*. 2016.
[142] L.D.R.D.S.E. Silva Junior and N. Nedjah. Distributed strategy for robots recruitment in swarm-based systems. *International Journal of Bio-Inspired Computation*, 2016.
[143] George Marios Skaltsis, Hyo-Sang Shin, and Antonios Tsourdos. A survey of task allocation techniques in mas. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 488–497. IEEE, 2021.
[144] Enrica Soria, Fabrizio Schiano, and Dario Floreano. Swarmlab: A matlab drone swarm simulator. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8005–8011. IEEE, 2020.
[145] Joanna Turner, Qinggang Meng, Gerald Schaefer, Amanda Whitbrook, and Andrea Soltoggio. Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system. *IEEE transactions on cybernetics*, 48(9):2583–2597, 2017.
[146] T. Vicsek, A. Czirok, E. Benjacob, I. Cohen, and O. Shochet. Novel type of phase-transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, 1995. ISSN 0031-9007.
[147] Christopher Vo, Joseph F Harrison, and Jyh-Ming Lien. Behavior-based motion planning for group control. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3768–3773. IEEE, 2009.
[148] Yufei Wei, Motoaki Hiraga, Kazuhiro Ohkura, and Zlatan Car. Autonomous task allocation by artificial evolution for robotic swarms in complex tasks. *Artificial Life and Robotics*, 24(1):127–134, 2019.
[149] Uri Wilensky. Netlogo models library. URL http://ccl.northwestern.edu/netlogo/models/.
[150] Annie S Wu, R Paul Wiegand, and Ramya Pradhan. Response probability enhances robustness in decentralized threshold-based robotic swarms. *Swarm Intelligence*, 14(3):233–258, 2020.
[151] Meng Wu, Xiaomin Zhu, Li Ma, Ji Wang, Weidong Bao, Wenji Li, and Zhun Fan. Torch: Strategy evolution in swarm robots using heterogeneous–homogeneous coevolution method. *Journal of Industrial Information Integration*, 25: 100239, 2022.
[152] Jian Yang, Donghui Zhao, Xinhao Xiang, and Yuhui Shi. Robotic brain storm optimization: A multi-target collaborative searching paradigm for swarm robotics. In Ying Tan and Yuhui Shi, editors, *Advances in Swarm Intelligence*, pages 155–167, Cham, 2021. Springer International Publishing. ISBN 978-3-030-78811-7.
[153] Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.

[154] Xudong Yang. Reinforcement learning for multi-robot system: A review. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pages 203–213. IEEE, 2021.

[155] Yongming Yang, Changjiu Zhou, and Yantao Tian. Swarm robots task allocation based on response threshold model. In *2009 4th International Conference on Autonomous Robots and Agents*, pages 171–176. IEEE, 2009.

[156] T. Yasuda, K. Kage, and K. Ohkura. Response threshold-based task allocation in a reinforcement learning robotic swarm. 2014.

[157] P. Zahadat and T. Schmickl. Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition. *Adaptive Behavior*, 2016.

[158] P. Zahadat, S. Hahshold, R. Thenius, K. Crailsheim, and S. Thomas. From honeybees to robots and back: Division of labour based on partitioning social inhibition. *Bioinspiration and Biomimetics*, 2015.

[159] G.-Y. Zhang, J.-C. Zeng, and S.-D. Xue. Research on task allocation of multi-target search with swarm robots. *International Journal of Wireless and Mobile Computing*, 2014.

[160] Guo-You Zhang, Jian-Chao Zeng, and Song-Dong Xue. Research on task allocation of multi-target search with swarm robots. *International Journal of Wireless and Mobile Computing*, 7(3):297–304, 2014.

[161] J. Zhou, D. Mu, F. Yang, and G. Dai. Labor division for swarm robotic systems with arbitrary finite number of task types. 2014.

[162] Bing Zhu, Lihua Xie, and Duo Han. Recent developments in control and optimization of swarm systems: A brief survey. In *2016 12th IEEE international conference on control and automation (ICCA)*, pages 19–24. IEEE, 2016.