Neurocomputing 172 (2016) 292–321

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A review of swarm robotics tasks

Levent Bayındır*

Department of Computer Engineering, Ataturk University, 25240 Erzurum, Turkey

ARTICLE INFO

Article history:
Received 2 August 2014
Received in revised form
8 May 2015
Accepted 11 May 2015
Available online 4 August 2015

Keywords:
Swarm robotics
Distributed task
Cooperation

ABSTRACT

Swarm intelligence principles have been widely studied and applied to a number of different tasks where a group of autonomous robots is used to solve a problem with a distributed approach, i.e. without central coordination. A survey of such tasks is presented, illustrating various algorithms that have been used to tackle the challenges imposed by each task. Aggregation, flocking, foraging, object clustering and sorting, navigation, path formation, deployment, collaborative manipulation and task allocation problems are described in detail, and a high-level overview is provided for other swarm robotics tasks. For each of the main tasks, (1) swarm design methods are identified, (2) past works are divided in task-specific categories, and (3) mathematical models and performance metrics are described. Consistently with the swarm intelligence paradigm, the main focus is on studies characterized by distributed control, simplicity of individual robots and locality of sensing and communication. Distributed algorithms are shown to bring cooperation between agents, obtained in various forms and often without explicitly programming a cooperative behavior in the single robot controllers. Offline and online learning approaches are described, and some examples of past works utilizing these approaches are reviewed.

1. Introduction

Swarm robotics is a field of research which studies how systems composed of multiple autonomous agents (robots) can be used to accomplish collective tasks, where the tasks either cannot be accomplished by each individual robot alone, or are carried out more effectively by the robots as a group. Dudek et al. [1] identified the following categories for tasks executable by robots: tasks that are inherently single-agent, tasks that may benefit from the use of multiple agents, tasks that are traditionally multi-agent, and tasks that require multiple agents. The swarm robotics discipline focuses on the last three categories, and past works have demonstrated in many application domains that using a multitude of agents to solve a task in a distributed manner allows working with significantly less complex agents at the individual level.

Three desired properties have been identified in a seminal paper by Şahin [2] as main motivations for swarm robotics studies: scalability, flexibility and robustness. The author defined a set of criteria to distinguish swarm robotics research from related disciplines: robots are autonomous, i.e. capable of moving and interacting with the environment without centralized control; the task at hand can be carried out collectively by a large number of robots, meaning that the system should be designed with scalability in mind; the swarm is made of relatively few homogeneous groups of robots, the focus being on large numbers of identical individuals rather than on centrally planned heterogeneous teams where each individual has a predefined role; the capabilities of a single robot (such as sensing, communication and computation capabilities) are limited compared to the difficulty of the collective task; and finally, sensing and communication are done by each robot at a local level, ensuring that interactions between swarm members are distributed and do not rely on 协调机制 coordination mechanisms that would hinder scalability. Swarm robotics takes inspiration from the collective behavior observed in nature in many living species, where local interactions between individuals and with the environment lead a group of autonomous agents to solve complex tasks in a distributed manner, without a central control unit. The locality of interactions and communication, which might be seen as a limitation, has a beneficial effect on scalability and robustness of the system, and is thus generally preferred over the use of global communication and sensing.

The expression "swarm intelligence", which is now widely used in the field of swarm robotics, refers to the superior capabilities of a swarm of agents compared to its single individuals. The local events triggered by swarm members during execution of a task translate into a global behavior which often transcends the individual capabilities, to the point that many collective tasks can be successfully done by robots that are not explicitly programmed to execute those tasks: the global, macroscopic dynamics is said to *emerge* from interactions of swarm members between each other and with the environment.

* Tel.: +90 442 231 4905.
E-mail address: levent.bayindir@atauni.edu.tr

http://dx.doi.org/10.1016/j.neucom.2015.05.116
0925-2312/

The possibility to achieve global objectives at the swarm level by means of distributed algorithms acting at the individual level comes at a price: it is often difficult to design the individual robot behavior so that the global performance is maximized. This problem has been widely studied by swarm robotics researchers, and has been addressed with simulation, modeling and learning approaches. Simulation, where a given multi-robot scenario is replicated in a virtual environment in which robot capabilities (sensors and actuators) and interactions are simulated by a computer program, allows assessing the performance of a robot swarm with repeated runs of an experiment, eliminating or mitigating the need for time-consuming experiments with real robots and facilitating algorithm optimization with a trial-and-error approach. Modeling (more precisely, macroscopic modeling) utilizes mathematical formulas to link individual-level control parameters to swarm-level dynamics; with such formulas, the impact of algorithm parameters can be evaluated directly, and often valuable insights on the global dynamics of the swarm can be intuitively obtained. Learning refers generically to adaptation of algorithm parameters based on previous experience; learning methods can be categorized in offline approaches, where the parameter optimization phase is part of the design of robot controllers, and online approaches, where robots dynamically update their control parameters based on their perception of the environment.

A widely used offline learning method is artificial evolution, which, starting from initial values of algorithm parameters, iteratively executes robot experiments evaluating for each experiment a fitness function which estimates the performance of the algorithm in executing a swarm-level task; the most performing parameter values at a given iteration are identified and used as a basis to program robots in subsequent iterations. Similar to what happens in nature with the evolution of species, robots are able to evolve their behavior across different "generations" and accomplish the given task. While neural networks are a common type of robot controllers used with artificial evolution, recent works explored the use of alternative methods such as rule-based grammatical evolution [3].

Online learning methods have been shown to be able to increase the flexibility of a swarm, i.e. its capability to adapt to different environment conditions. By definition, robots learning during task execution must have some form of memory which allows them to remember past experiences in order to adapt their future behavior; thus, inclusion of online learning methods in robot controllers implies an additional level of complexity in robot implementation. But generally the biggest difficulties encountered in this domain are due to different aspects: first, robots often have a limited and noisy perception of the environment and of the progression of a global task; second, as already discussed, the distributed nature of the problem makes it difficult to relate individual behavior to global performance. Two types of online learning methods can be identified in past works: reinforcement learning and parameter adaptation. Reinforcement learning is based on a model where robots, which can be in a given set of states and can execute a given set of actions, receive feedback on the results of their actions through a *reward*; the objective of robots is to choose a mapping between states and actions so as to maximize the reward. Using a local reinforcement paradigm, the reward is assigned only to robots which directly accomplish an objective, while with global reinforcement all the robots are rewarded for each accomplishment; local reinforcement is more coherent with swarm intelligence principles because it does not require sharing global information in the swarm. Other online learning methods can be described as based on dynamic adaptation of robot algorithm parameters triggered by observations of the environment.

In this paper, various tasks for which past works have proposed solutions using a swarm intelligence approach are surveyed, focusing on distributed control, locality of interactions and simplicity of individual robot controllers. The next section is dedicated to previous swarm robotics reviews; then, the subsequent sections describe the different tasks and the corresponding solutions proposed in past studies; finally, future research directions are outlined and concluding remarks are made in the last sections.

2. Previous work

In the last two decades, theoretical research on multi-robot systems has been fueled by technological advances that now allow building relatively cheap small robots. An early categorization of multi-robot systems is given by Dudek et al. [4,1], who identified swarm size, communication range, communication topology, communication bandwidth, swarm reconfigurability, swarm unit processing ability and swarm composition as taxonomy axes to classify natural or engineered multi-agent systems.

The fundamental notion of *cooperation* between robots plays a central role in determining whether a multi-robot system performs better than equivalent single-robot systems, and as such has been discussed in a number of existing surveys. For example, cooperation is the central topic in [5], where swarm robotics systems are analyzed in terms of group architecture of the swarm (indicating with this term properties such as centralization versus decentralization, homogeneity versus heterogeneity, direct or indirect communication between agents, and how agents model each other), interference problems due to sharing of common resources, origin of cooperation (with interesting examples of how cooperation can be achieved implicitly even if each agent acts to maximize its individual utility), and learning mechanisms (with focus on reinforcement learning and genetic algorithms); in addition, a number of studies are grouped under the category of geometric problems, such as multiple-robot path planning and formation and marching problems. Iocchi et al. [6] used cooperation as the first level of a multi-level characterization of robot swarms; cooperative systems are then further differentiated at the knowledge level, where systems with robots aware of the existence of other robots are distinguished from systems where each robot acts as if it was alone. The lower levels of the proposed taxonomy are the coordination level, describing how the actions of each robot take into account the actions of other robots, and the organization level, which determines whether decisions are taken in a centralized or distributed way; it is interesting to note that a centralized system may be compatible with swarm intelligence principles: more precisely, systems defined as *weakly centralized*, where one of the robots temporarily assumes the role of leader, can exhibit the desired property of fault tolerance provided that suitable mechanisms are in place to assign the leadership role.

While early papers provided characterization of swarm robotics systems mainly as an analysis tool to encourage further research and give guidance for the design of new systems, with the increasing number of published studies describing actual implementations of robot swarms, newer surveys have been able to propose taxonomies where each category is represented by several examples of existing works. An extensive review of the state of the art in the mid 2000s is provided in [7], where existing works are categorized based on analytical modeling approaches, design of individual robot behavior, type of interactions between robots, and problems being addressed by a robotic swarm. Gazi and Fidan [8] focused on the aspect of controlling robot movement, dividing existing works based on how the robot dynamics is modeled (i.e. how control inputs map to position variations) and how robot controllers are engineered; in addition, a further classification is done on the problem dimension. Previous works have been classified on the problem dimension also in other surveys [9,10]; Mohan and Ponnambalam [11] analyzed various research domains in swarm robotics, however their review does not provide a clear categorization of the state of the art, mixing a classification of some studies in the problem dimension with a description of how other studies differ on aspects such as biological inspiration, communication between robots, control approach and learning.

A comprehensive survey recently published is the work by Brambilla et al. [12], who proposed two taxonomies for swarm robotics studies: methods and collective behaviors. The first taxonomy includes design methods, differentiated in behavior-based and automatic methods, and analysis methods, i.e. techniques to study the performance of a swarm in executing a given task; analysis methods are divided into microscopic models, macroscopic models and real-robot analysis. The second taxonomy is based on the concept of collective behaviors, i.e. behaviors of a swarm of robots considered as a whole; the main categories identified by the authors under this taxonomy are spatially organizing behaviors, navigation behaviors and collective decision making.

Barca and Sekercioglu [13] analyzed past research by identifying a series of challenges faced by swarm robotics systems and describing how each of these challenges has been addressed by existing studies. Challenges are grouped under five categories: selecting appropriate communication and control schemes, incorporating self-organization, scalability and robustness properties, devising mechanisms to support goal-oriented formations, control and connectivity, implementing functions that enable robots to interact efficiently with the environment, and addressing problems related to energy consumption. The authors outlined a number of issues that need to be tackled to overcome these challenges, and observed how existing works typically focus on only a subset of these issues, suggesting that in order to implement successfully a swarm robotics system in real-world applications a larger set of issues should be tackled simultaneously.

In this review, the problem dimension is used as main taxonomy axis, thus grouping past works according to the collective task being addressed. For each of the most studied tasks, first the main high-level methods employed in past studies are described, then task-specific categories are identified and a more detailed description of distributed algorithms is provided for a representative set of existing works in each category, and finally mathematical models to analyze and predict swarm behavior, and methods and metrics to evaluate swarm-level performance are reviewed. Due to similarities and analogies between different collective tasks, multiple equally valid categorizations can and have been proposed in past reviews under this taxonomy, and many works can be put in more than one category; in this study, partially different categories are identified compared to existing reviews, further categorizations within each task are proposed, and differences, similarities and relationships between tasks are explained.

3. Aggregation

Self-organized aggregation, i.e. the task of gathering a number of autonomous individuals in a common place, is a basic behavior widely observed in nature with many animal species. Various mathematical models have been proposed to describe aggregation, and robotic systems have been engineered with various algorithms to implement aggregation dynamics. This task has been studied either as a standalone problem, or in the context of more specialized tasks which require gathering multiple agents.

3.1. Methods

The majority of swarm robotics studies proposing an algorithm to obtain aggregation in an artificial swarm used one of the following design methods to control robot movement: application of virtual forces (artificial physics), control of robot behavior based on a probabilistic approach, and artificial evolution.

3.1.1. Artificial physics

Artificial physics is a field of research that models the behavior of individual agents using virtual forces. These forces determine the movement of agents and consequently the interactions between agents and the surrounding environment. Many animal formations observed in nature (such as insect swarms, bird flocks and fish schools) can be modeled with attraction forces (through which different agents tend to stay near each other) and repulsion forces (which prevent collisions between individuals). Each agent moves according to the force exerted on it by each of its neighbors, which depends on the neighbor distance. Usually, the force is attractive if the distance between two agents is greater than a target value, and is repulsive for smaller distance values.

Although artificial physics has been successfully used to formally characterize the aggregation dynamics of swarms of autonomous agents [14–18], its practical implementation in artificial systems with real robots imposes some requirements on robot sensing capabilities which may not be cost-effective. Robots with local sensing abilities are typically characterized by a limited range of visibility, which influences the capability to perceive other robots in the environment; determination of the relative orientation of neighbors may be affected by a high error, such as when infrared technology is used; mechanical constraints usually determine a saturation effect in robot actuators, effectively limiting the amplitude of the control inputs which regulate robot motion. For these reasons, only a limited number of studies considered applying artificial physics rules to control robot movement. One of these studies is the work by Priolo [19], where a swarm aggregation algorithm based on artificial physics is implemented and tested with real robots. In this algorithm, distance and relative orientation of neighboring robots are obtained via radio frequency and infrared technologies; as usual, attraction forces are stronger at high distance values between robots, while repulsion dominates at low distance values. In addition, an obstacle avoidance mechanism is devised in which obstacles are modeled as virtual robots; in order to avoid attractive force between robots and obstacles, virtual robots are activated only when their distance to a real robot is such that repulsion is stronger than attraction.

3.1.2. Probabilistic methods

With a probabilistic approach, the behavior of each robot is determined partially in a random manner, and partially based on its interactions with the surrounding environment. This type of behavior is found in nature with many social insects, such as honeybees and cockroaches, and has been extensively used in various studies to obtain aggregation using minimalistic robots.

Taking inspiration from social insects, a number of probabilistic robot control algorithms achieve aggregation using a finite state machine characterized by two basic states: *walk* and *wait* [20–32]. At each state corresponds a specific robot behavior; the different states in use (and the behavior implemented in each state) vary between algorithms, but many similarities can be found in various aggregation studies. In some cases the walk state is split into two different states: one where a robot tries to approach other robots, and one where it distances itself from its neighbors [33,34]. The decision to switch between states can be taken in a purely random way, or based on local cues, which can be as simple as the presence of nearby robots, or may involve more complex algorithms and signaling mechanisms [35,36]. Parameters of the finite state machine, such as the probabilities of switching between states, are typically chosen manually by the swarm designer, but recently alternative methods based on automatic design have been proposed [37].

A common point in probabilistic aggregation methods is the presence of *unstable* aggregates, with robots continuously joining and leaving them. The aggregation dynamics arises from changes in randomness of robot behavior due to the detection of neighboring robots: while disaggregated robots typically move randomly in the environment, the dynamics of aggregated robots is more deterministic. However, a random component in the behavior of aggregated robots is often necessary to facilitate the formation of a small number of large aggregates, avoiding situations where the presence of small aggregates prevents robots from joining larger aggregates. In studies where a different control algorithm than a finite state machine is used, typically there is not a clear distinction between aggregated and non-aggregated robots, but the swarm dynamics can be measured based on metrics such as the average distance between robots, and the randomness of robot movement can be changed on a continuous scale [38].

3.1.3. Evolutionary methods

With evolutionary methods, the aggregation dynamics is obtained using robot controllers whose parameters are selected through artificial evolution. Neural networks linking sensory inputs to actuator outputs are a common type of controllers evolved using these methods. Depending on the algorithm in use, sensory inputs can include devices able to detect a characteristic of the environment [39], and actuator outputs can include devices enabling robots to communicate with other robots [40]. Examples of algorithms used for artificial evolution are genetic algorithms [39,40] and *q*-tournament selection [41].

The standard paradigm used for evolving a population with artificial evolution is based on a fitness value measuring the ability of a given generation of individuals to accomplish a task. Gomes et al. [42] proposed a different algorithm, based on the concept of *novelty search*. As opposed to fitness-based evolution, novelty search-based evolution rewards robots whose behavior differs from the behaviors observed in past generations. This method allows avoiding a possible drawback of fitness-based evolution where local maxima of the fitness function in the parameter space may prevent exploration of other parts of the space and thus limit evolution. In [42], when novelty search is applied to the aggregation task, the characterization of the behavior of a given generation is done by measuring metrics such as the average distance of each robot from the center of mass of all the robots, or the total number of aggregates; these metrics are measured at multiple time instants during a simulation, and their values (averaged over different simulations) are inserted in a behavior characterization vector, which is used to calculate the similarity between behaviors. The authors compared simulation results using novelty search and fitness-based evolution, and showed that, while the latter is generally better in finding the optimum parameters for the aggregation task after many generations, with novelty search better results are obtained during the first phase of the evolution, and are slightly refined in subsequent generations, yielding performance values similar to those obtained with fitness-based evolution.

Novelty search-based methods rely on a measure of similarity between robot behaviors, from which the novelty of a given robot can be assessed. Gomes and Christensen [43] proposed two similarity measures that are independent from a specific swarm-level task and as such can be used without having domain knowledge of the task at hand. Both measures are based on evaluating the state of the neural controller implemented in each robot; the state is defined as a vector containing the values of the inputs and outputs of the controller at a given time. The first measure, called *combined state count*, characterizes a behavior by discretizing the possible states of the controller and counting the occurrences of each state (sampled at defined intervals) during an experiment run. The second measure is called *sampled average state* and is based on calculating a vector containing the average state of the robotic swarm (i.e. the state obtained by averaging the states of all the robots) sampled at fixed intervals. Both measures, when evaluated for the aggregation task, showed good results, comparable with the results obtained using a domain-specific similarity measure.

3.2. Algorithms

3.2.1. Free aggregation

In free aggregation algorithms, robots are given the task of aggregating without any preference for a specific aggregation site, thus robots can gather in principle with equal likelihood at any location within the arena in which they move.

A widely studied probabilistic aggregation algorithm takes inspiration from the behavior observed in cockroaches. In a simplified model of cockroach aggregation, these insects move randomly in the environment and stop at a given location based on the number of detected neighbors: the probability of stopping is a function of the number of cockroaches detected within a defined sensing range, with a higher number of cockroaches corresponding to a higher probability of stopping. Conversely, a stopped cockroach can resume random walking at any time, possibly leaving an aggregate, and the probability of switching to the walk state is higher when the number of detected cockroaches is lower. With this simple behavior, the dynamics of random encounters between cockroaches in a closed arena leads to the formation of aggregates, as demonstrated by various simulation experiments [29].

In [31], where robots are controlled by a finite state machine with three states (*random walk*, *approach* and *wait*), the walk state lasts for a fixed amount of time, after which a robot senses its surrounding area: if it detects other robots, it switches to the approach state, where it moves toward the nearest detected robot and then switches to the wait state, otherwise it switches directly to the wait state; from this state, a robot switches back to random walk with a predefined probability. Using this algorithm, the global aggregation dynamics is regulated by the probability for robots to detect other robots after random walking, with large aggregates being easier to detect than single robots.

Bayındır [34] proposed another algorithm based on a finite state machine, where the aggregation behavior is obtained using four states called *search*, *wait*, *leave* and *change direction*. The wait state, which is entered when a searching robot detects the presence of other robots and aggregates with them, is designed so that the robot tries to keep a fixed distance from each of its neighbors; this allows the formation of compact aggregates with approximately circular shape. As usual, to avoid situations where small aggregates prevent the formation of larger aggregates, robots can leave their aggregate at any time with a defined probability.

In [33], aggregation is obtained in a simulated robotic system where each robot is equipped with an omnidirectional speaker and a set of microphones, and uses the sound emitted by the other robots to determine their relative direction and proximity. The basic states of the finite state machine implemented in the robots are called *approach*, *wait* and *repel*. In the approach state, robots move toward the direction of the loudest sound, while in the repel state they move in the opposite direction; when a robot in the approach state senses another robot in close proximity, it switches to the wait state, where it stays in its current position; from this state, it switches to the repel state with a given probability, and then returns to the approach state with another probability value.

In [35], Trianni et al. presented a system where robots are equipped with a light source that can be used to signal their presence. From its sensory perceptions (robots are able to detect the presence of other nearby robots and to measure the intensity of received light), each robot creates a *context*, i.e. a high-level abstraction of the surrounding environment; at each time instant, the robot selects randomly its behavior among a set of predefined behaviors (which include turning on and off its light, and moving towards or away from other robots); the probabilities to select the various behaviors are defined based on the perceived context. This generic algorithm can be adapted to different collective tasks by specifying how the sensory inputs map to the perceived context and defining the probabilities of activating the basic behaviors from each possible context; as shown by the authors, aggregation is one of the tasks that can be executed.

In [30], the probability of an aggregated robot to leave the aggregate is determined by its orientation relative to the rest of the robots in the aggregate; a robot pointing towards the center of the aggregate has a lower probability of leaving than a robot pointing at other directions. Following various examples found in nature such as the assembly of molecules, the stability of an aggregate is described in terms of the energy of the bonds between its robots, which is a function of the relative orientation of the robots. For simplicity, the authors limited their discussion to two-robot aggregates, where the energy of the aggregate equals the energy of the bond between its two robots.

Gauci et al. [41,44] used minimalistic aggregation algorithms where sensor input obtained by robots from the environment is limited to one binary variable that indicates whether there is another robot in their line of sight. In [41], aggregation is obtained through artificial evolution. In [44], the control algorithm makes robots move backwards along a circular trajectory if no robots are perceived in their line of sight, and rotate on the spot otherwise; this simple mechanism is shown by the authors to provide emergent aggregation if robot sensors have a sufficiently long range. However, due to the lack of a behavior similar to random walk, aggregation cannot be guaranteed if robots are initially placed at a distance from one another larger than their sensing range.

In [40], robots equipped with microphones, proximity sensors, wheels and a speaker are controlled by a neural network whose parameters are evolved with a genetic algorithm. The authors observed the emergence of two types of collective behavior: static and dynamic aggregation. The first type determines the formation of compact and static aggregates, which is shown not to be scalable because many robots in the arena tend to form multiple disjoint clusters. With dynamic aggregation, the formed aggregates are less compact, but keep moving in the arena, and when many robots are present this leads different aggregates to join and form a single aggregate, thus showing more scalability. In [45], a similar robotic system is evaluated running multiple experiments with varying setups, and the effect of different characteristics such as arena size and number of robots is assessed in terms of performance and scalability of the evolved behavior. In [42,43], artificial evolution is implemented exploiting the concept of *novelty search*.

In [36], the aggregation dynamics is obtained using an *active* environment, which is able to propagate potentially at a long distance signals emitted by robots. The aggregation algorithm proposed by the authors is inspired by the formation of multicellular organisms via diffusion of chemical agents. Each robot emits a signal (the emission rate is regulated by a specific parameter, called *firing rate*) that propagates in the surrounding environment; other robots (which in the absence of this signal can either stay at their current position, or move in a random direction) are attracted by this signal and move toward its source. With this mechanism, robots tend to move toward each other and form one or multiple aggregates. In [46], this aggregation method is implemented with real robots.

In other studies, the aggregation process is not considered as the formation of a collection of mostly static individuals in close proximity of each other, but is described in terms of density of robots in a given space. For example, aggregation obtained with artificial physics methods [14–19] generally does not lend itself to a precise identification of discrete aggregates. In [38], where robots are able to sense the population density in their neighborhood, the aggregation dynamics is obtained with robots increasing and decreasing the randomness of their movement based on the local density: while robots in low-density areas move in a highly random fashion, in high-density areas the stochastic component of robot movement is lower, leading to robots "settling down" in regions with a high number of neighbors. In [47], the aggregation algorithm is controlled by wireless connectivity between robots with a limited communication range. Robots in the arena move with a constant speed and transmit periodically a wireless message containing their unique identifier; this message allows receiving robots to detect the presence and count the number of robots within communication range; each robot tries to keep this number above a defined threshold: if the number of neighboring robots falls below the threshold value, the robot inverts the direction of its movement, while when the number of neighbors increases above the threshold, the robot executes a random turn. Aggregates are thus formed as dynamic structures where each robot is in communication range of many other robots.

3.2.2. Environment-mediated aggregation

In environment-mediated aggregation algorithms, the location of a given robot in the environment influences the robot behavior, so that aggregation is achieved with higher probability in some defined "preferential" regions. Nature offers many examples of aggregation between individuals influenced by the local environment. For example, honeybees tend to aggregate in areas with optimal temperature. Honeybees are unable to sense the temperature gradient in the environment. To overcome this difficulty, each honeybee exploits the presence of other conspecifics: it starts moving randomly until it collides with another honeybee; when a collision occurs, the honeybee stops, and remains stopped for a time duration dependent on the local temperature; after this time expires, the honeybee resumes its random walk until the next collision. Cockroaches exhibit a similar behavior: they tend to aggregate in dark places, and stop their random walk with a probability depending (beside on the number of cockroaches detected in close proximity) on local environment conditions.

The aggregation behavior of honeybees has been replicated in a number of studies with physical robots placed in a closed arena, where a light source above the arena is used to simulate a temperature gradient, and robots are able to sense the local luminance [20,22]. In [25], the basic algorithm is enhanced with two modifications: robots change their walking velocity based on the local luminance, with higher velocity corresponding to darker areas, and increase their waiting time when stopped near a high number of neighbors. In [48,49], the aggregation area is signaled with a sound source, and robots are equipped with microphones to measure the sound intensity; in order to increase the aggregation efficiency, each robot has a set of microphones oriented at different directions, and when resuming walking moves toward the estimated direction of the sound source.

In other studies [21,23,50], two distinct light sources are put in the environment, and the aggregation dynamics varies with the relative size and intensity of the light spots. A particular case of the scenario with two light sources in the arena is when the two sources are identical. As explained in [24], where experiments with this scenario are evaluated, robots tend to aggregate under one of the two identical sources. This behavior is an example of *symmetry breaking*, which is observed in many robotic systems, where robots converge to a unanimous decision in front of two options with the same utility.

The aggregation behavior of cockroaches is similar to that of honeybees, but cockroaches have in addition the ability to detect the number of neighbors, and the probabilities of entering the wait and walk states are dependent on the number of detected neighbors [51]. Since cockroaches tend to prefer dark places as resting sites, the size of the available dark places, in terms of the number of cockroaches they can host, influences the aggregation dynamics; in [26–28], this type of scenarios is analyzed with experiments and theoretical studies. In [39], a simulated swarm of robots controlled by a neural network is used to replicate the behavior of cockroaches collectively selecting a single resting site among two identical shelters. Robots are able to detect whether their current location is inside a shelter, and are able to count the number of neighboring robots within a limited range. In the genetic algorithm to select the optimal parameters of the neural controller, the fitness function is chosen so as to reward the behavior where the majority of robots aggregates under a single shelter. In a subsequent study [37], the same behavior is obtained using probabilistic finite state machines whose parameters are selected with an optimization algorithm.

Schmickl et al. [32] studied an aggregation scenario where robots moving in an arena with two differently sized target areas must form in the target areas aggregates of a number of robots proportional to the area sizes. Robots are able to detect whether they are inside a target area, but cannot measure the size of the area. To achieve the aggregation task, each robot keeps in its memory a scalar value which corresponds to its perception of the environment. A robot can receive from other robots within communication range the value in their memory; at predefined time intervals, the robot updates the value in its memory based on whether it is in the target area and on the values received from other robots within communication range. Movement of a robot in the arena depends on the current value in its memory: if the value is above a defined threshold, the robot tries to approach the neighboring robot with the highest value in its memory, otherwise the robot moves randomly. With this mechanism, a “collective perception” of the environment is obtained via local communication, despite the limited sensing capabilities of single robots.

3.3. Analysis

3.3.1. Metrics

Performance metrics for the aggregation task are mainly based on either identifying discrete groups of robots forming aggregates, or measuring the spatial distribution of all robots in the arena. In the first case, a formal definition of an aggregate is needed. Often, an aggregate is defined as a group of robots such that for any pair of robots in the group there is a path connecting them formed by robots within a maximum distance from each other; the value of the maximum distance is usually chosen based on the range of local communication and sensing of robots. In studies where robots are controlled by a finite state machine, aggregates can be identified by robots whose controller is in the “wait” state. Once a suitable definition of aggregates is adopted, performance metrics can be calculated as the ratio of the number of robots forming the largest aggregate to the total number of robots [45,31,44], or as the average aggregate size [33,52]; a more in-depth analysis of the aggregation dynamics can be conducted by observing the distribution of robots belonging to aggregates of different sizes [35]. In tasks where the objective is to aggregate robots in a given area, a common metric is given by the number of robots located in the target area [26,53,32,28,21,48] or within a certain distance from a defined aggregation spot [22,50].

The second type of metrics involves locating all robots in the arena and finding a measure of their spatial relationships. Soysal and Şahin [33] used the sum of distances between each pair of robots; other studies used the average distance of robots from the center of mass of the swarm [40–43]; Gauci et al. [44] used the “second moment of the robots”, calculated by summing the squares of the distances of each robot from the center of mass; Fatès [36] used the “bounding box ratio”, defined as the ratio of the surface of the smallest rectangle containing all the robots over the total surface of the arena.

The temporal dimension in the aggregation task is taken into account in performance metrics that measure the speed at which a swarm achieves an aggregation target [20,25,48,49]; usually, such metrics are calculated as the time elapsed before a given percentage of robots forms an aggregate, starting from initial conditions where robots are positioned randomly in the arena.

3.3.2. Models

The aggregation behavior has been studied extensively in both natural and artificial swarms, and mathematical models have been proposed to predict the performance of a swarm in achieving self-organized aggregation. These models allow calculating macroscopic quantities describing the collective behavior of a swarm from parameters governing the individual robot behavior.

A mathematical description of how the inter-individual relationships between swarm members subject to artificial physics rules determine the swarm dynamics is given in [14]. The spatial density of agents in the environment is modeled using a conservation law expressed with the advection–diffusion equation; this partial differential equation models the density variation over time as the sum of a diffusion term (which models the tendency of randomly moving particles to go toward less concentrated areas) and an advection term (which accounts for attractive and repulsive forces, and determines a non-random velocity component in the agent motion). The authors stated that with local advection (i.e. a model where the velocity at a given location is only a function of the agent density at that location) many swarming behaviors observed in nature cannot be modeled; thus, a non-local advection term is proposed, where the velocity is calculated as the convolution of the particle density with a kernel function. The advection–diffusion equation becomes then an integro-differential equation. Expressing the kernel function as a combination of attraction and repulsion terms, mathematical analysis and numerical simulations show how these two factors influence the spatiotemporal characteristics of the swarm.

In [15], the spatial patterns formed by agents are studied via a conservation law without a diffusion term: the advection equation. A non-local advection type is considered, and steady state solutions for the equation are derived for different kernel functions (called *interaction potentials* in [15]). The solutions show the formation of various aggregation patterns starting from a uniform agent distribution. In addition, the dynamics determining the time needed to reach the steady state, which can offer useful insights on factors influencing the aggregation performance, is analyzed.

In [16], a robotic swarm where the interaction between robots is characterized by long-range attraction and short-range repulsion forces is studied with the Lyapunov theory, which allows finding the equilibrium point of the system. The position of each robot relative to the centroid of the swarm is described by means of a set of ordinary differential equations, and a Lyapunov function with attraction and repulsion components between robots is proposed in order to find a stable equilibrium point. The Lyapunov function is defined in terms of three sets of parameters, called *coupling*, *cohesion* and *convergence* parameters; the equilibrium point, i.e. the set of positions of each robot relative to the swarm centroid when the system converges to a stable state, is shown to change depending on the values of these parameters. Through computer simulations, the authors obtained various swarming behaviors similar to those observed in nature with social animals.

In [17], robots move with a constant speed in a two-dimensional space, and their reciprocal interactions are governed by three forces (repulsion, alignment and attraction) acting at short, intermediate and long distance, respectively. Specifically, the alignment force accounts for the tendency of different agents to move in the same direction, as observed in many groups of animal species. The density of agents is expressed as a function of spatial coordinates, heading direction and time, and is modeled using an integro-differential equation in which the strength of the three interaction forces is determined by specific parameters. Solving the equation with different parameter values, various pattern formations can be obtained; for example, by changing the strength of the alignment force the solution of the equation varies from a swarm-like pattern (with low alignment force) whose single agents move randomly but which does not advance in any direction, to a flock formation (with high alignment force) whose agents aggregate while moving along a common direction.

In [18], the aggregation dynamics of the above model is extended with two additional considerations: the agents may have a non-uniform field of vision, and there may be another group of agents which interacts with the first group. The non-uniform field of vision is introduced to model more realistically the behavior of animals, which typically cannot see behind themselves; thus the position and orientation of an agent is influenced by the presence of other agents depending on the orientation of the other agents relative to the field of vision of the first agent. The second factor introduced in [18] is the presence of a second group of agents: with the combined system it is possible to model the interactions between different animal species, as for example in predator–prey relationships. In the mathematical model, these interactions are expressed by additional terms (taking into account the attraction and repulsion forces between groups of agents) in the integro-differential equations which govern the density of agents of each group. By solving these equations, patterns of predator and prey movement commonly found in hunting and escape strategies of different animals can be reproduced.

In probabilistic aggregation methods, mathematical models characterizing the swarm dynamics are usually based on identifying some types of meaningful events that can occur in the swarm (for example, an encounter between two moving robots) and estimating the probability of occurrence of such events. In [29], the aggregation behavior of cockroaches is studied using a model based on difference equations, which allows calculating the expected number of robots in aggregates of a given size at a given time. This model, which incorporates the events where the size of an aggregate changes, such as when a robot joins or leaves an aggregate, is validated with simulation experiments and is shown to be able to describe qualitatively and quantitatively the swarm aggregation dynamics; however, results are only reported for a relatively low number of robots in the arena. A similar modeling approach is used by Hu et al. [52]. In [30], where aggregates are characterized by an energy value indicating the strength of the bond between aggregated robots, difference equations are used in a model estimating the number of non-aggregated robots and of aggregates with a given energy value. In [34], the aggregation dynamics is described using four basic events: *creation*, *growing*, *shrinking* and *dissipation*; with a probabilistic model utilizing geometric considerations, the probability of occurrence of each event is calculated, and model predictions are shown to match experimental results obtained from robot simulations in various operating conditions, including scenarios with a relatively large number of robots. In [47], where robots can establish wireless connectivity within a limited range, the main events of interest are when two robots acquire or lose connectivity between each other, and the macroscopic model estimates the number of robots with a given number of *connections*, i.e. other robots within communication range.

In algorithms where the aggregation process is influenced by local characteristics of the environment, mathematical models must take into account such characteristics, and the collective dynamics can be expressed either with the number of robots inside the areas designated for aggregation [21], or with the density of robots expressed as a function of spatial coordinates [53,54]. In [20] an experimental setup to simulate honeybee aggregation is described, and the probability of forming aggregates of a certain size in a given area is put in relation with the luminance of the area and other parameters such as the density of robots in the arena, their speed and their sensing range. Hamann et al. [22] proposed two models to estimate the time-varying pattern of moving and aggregated robots simulating the aggregation behavior of honeybees. In the first model, called compartmental model, the arena is split into concentric rings centered below the light source, and the number of moving robots and of aggregates in each ring is estimated with ordinary differential equations. In the second model, the space in the arena is not divided into discrete compartments but represented as a continuous variable, and the densities of moving and aggregated robots are expressed as functions of space and time variables and solved via partial differential equations. Schmickl et al. [23], who analyzed a scenario with two light sources of different intensity, studied the aggregation dynamics using a *stock and flow* model, with which the number of robots aggregated near each of the two light sources and the number of moving robots are predicted, and a second model that estimates the spatial distribution of moving and aggregated robots using partial differential equations.

4. Flocking

Flocking is a behavior observed in nature in many bird species, which form large groups of individuals moving together toward a common target location. Other examples of analogous collective behaviors found in animals are fish schooling and formation of herds in ungulates. These behaviors emerge at the collective level in a distributed manner, as a consequence of local interactions between autonomous agents, and as such are of interest to swarm robotics researchers, who have studied the mechanisms at the basis of animal behavior and tried to replicate flocking in robotic swarms. In the majority of existing works, robots with limited sensing capabilities must keep a compact formation by measuring distance and relative orientation of their neighbors; cases where single or groups of robots are outside the sensing and communication range of the rest of the robots are typically not considered in flocking studies, where the usual assumption is that all robots have at least one neighbor which connects them to the rest of the swarm.

4.1. Methods

As mentioned above, a fundamental component in robot behavior necessary to implement flocking is the ability to measure the distance and relative orientation of neighboring robots; the limited sensing and communication range typically found in real-world scenarios has the practical implication that only a limited number of neighbors, and not the entire population of the swarm, is detected by a given robot, but this limitation (which on the other hand can be seen as an advantage when scalability and processing complexity factors are taken into account) does not hinder the ability to implement flocking provided that there are not isolated individuals or groups within the swarm.

Compared to simple aggregation, flocking has an important additional characteristic at the swarm level: *alignment* of robot movement, which allows a group to move collectively on a given direction. In a seminal paper, Reynolds [55] simulated the behavior of flocking animals with three basic rules: collision avoidance, velocity matching and flock centering. If robots are endowed with the ability to know the heading direction of their neighbors, this ability can be exploited to implement flocking [17,56–61]. However, knowledge of the heading of neighbors is not a fixed requirement, as demonstrated in various studies [62–66] where robots do not have this knowledge.

4.2. Algorithms

Flocking animal species typically are able to orient themselves in the environment, and thus can move toward a common target location known by all individuals. This situation is replicated in engineered systems by giving individual robots the capability to reach a global target location, or at least to move along a given direction on a common reference frame. However, the swarm robotics discipline usually favors the use of minimalistic robots, with little or no global information shared by swarm members, and past works have demonstrated that flocking can be obtained also without global information. In the following, the algorithms proposed in some notable studies on the flocking behavior are reviewed, differentiating methods relying on a global target location or direction from methods without such shared knowledge.

4.2.1. Direction by global target

If some or all members of a swarm have knowledge of the target location to be reached by the swarm, such knowledge can be used to guide robot movement, and typically determines the approximate heading direction of flocking agents. The situation where swarm members share a global target direction on a common reference frame is conceptually analogous. If the target is known by all robots in the swarm, local interactions between robots serve mainly the purpose of maintaining a compact formation while avoiding collisions. In the case where only some robots (often referred to as "informed robots") have knowledge of the target, local interactions can be used also for spreading this knowledge to the entire swarm.

In [62] a flocking algorithm is proposed in which robots are tasked with navigating to a target location while keeping a target distance with neighboring robots; to achieve this task, at each time step individual robots calculate a "center of mass", which represents the desired location to be reached in the short term, and generate actuator commands to move toward this location. The center of mass is calculated taking into account the distance and relative orientation of neighboring robots, as well as the final target location. In addition, a collision avoidance mechanism is implemented and takes over control of a robot when an obstacle in close proximity is detected. A cost function is defined which measures the performance of the algorithm in achieving the task; this cost function is used as a performance metric for an off-line machine learning technique by which optimal values of algorithm parameters are found.

Baldassarre et al. [63] used artificial evolution to implement flocking with a swarm of robots equipped with infrared sensors (used to detect the nearby presence of obstacles and other robots), microphones (through which the relative position of other robots can be determined) and light sensors (used to detect the global target). Each robot continuously emits a fixed amplitude sound so that it can signal its presence to other robots beyond the coverage area of the infrared sensors. The fitness function used for artificial evolution includes a group compactness component, and a speed component measuring the progress of the swarm in approaching the target. From various replications of the evolution process, the authors obtained different flocking behaviors.

In [64], Null-Space-based Behavioral (NSB) control is proposed for the task of flocking. NSB control is used to determine the behavior of a robot in the presence of multiple sub-tasks, when each sub-task is accomplished by a specific behavior. With NSB control, different priorities are assigned to different behaviors, and the final behavior of a robot is obtained by combining the single behaviors after having removed from lower-priority behaviors the components which conflict with the higher-priority behaviors. In the specific case of flocking, three behaviors are identified, namely obstacle avoidance, flock formation and navigation to target, in decreasing priority order. Each behavior corresponds to a control input for robot movement, and the control inputs for lower-priority behaviors are adjusted in order not to conflict with those of higher-priority behaviors. The sum of the adjusted control inputs determines how a robot must be controlled to accomplish the task.

In cases where only a fraction of robots has knowledge of the global target, the motion control input to which robots are subjected depends on whether they are informed or not; a typical approach followed in various studies [57,66,61] is to include the target direction vector as an additional component in the motion control input of informed robots. Using local interactions, informed robots can propagate the information on the target to the other robots and steer the swarm toward the target. In [58], an *information-aware* communication strategy is implemented where informed robots transmit to their neighbors the goal direction vector, while non-informed robots transmit the average heading vector received from their neighbors. In [59], the above strategy is extended with a mechanism that allows dealing with multiple goal directions with different priorities, so that when a higher-priority direction appears the robots informed of the new target are able to steer the swarm and override a lower-priority direction; the heading vector transmitted by each robot to its neighbors is a weighted sum of the average heading in the neighborhood and the goal direction (if known), where the weight values are chosen dynamically based on the degree of local consensus among robots about the preferred direction: when there is a high consensus, informed robots give a large weight to their goal direction, otherwise their behavior is more similar to that of non-informed robots, which facilitates propagating to the swarm information on higher-priority targets. In [61], no explicit communication between robots is used, and flocking is implemented with each robot detecting its neighbors and defining a set of regions around itself: a repulsion area, a direction matching area, an attraction area and a frontal interaction area; the robot heading direction vector is then calculated from the position of neighbors inside each area.

4.2.2. Emergent direction

In studies reviewed in the previous section, information on a goal direction is either directly encoded in robot controllers, or sensed by some or all robots in the swarm, and determines the flocking direction. As demonstrated in this section, flocking can be obtained even in the absence of this information and with individual robots initially moving on random and uncorrelated directions: in this case, a swarm-level global direction of movement can emerge from local interactions.

The work by Fetecau [17] to study animal formations provides a mathematical model not only for aggregation behaviors, but also for flocking: specifically, the alignment forces by which each agent tends to align its direction of movement with that of its neighbors are shown to be able to produce flocking patterns, with a global direction emerging from local interactions.

Möslinger at al. [65] described a flocking algorithm using simulated robots with four infrared sensors. Each sensor can be used actively, i.e. it emits infrared light and detects whether the light is reflected by a nearby object (either another robot or an obstacle), or passively, i.e. it detects whether another source of infrared light (i.e. another robot) is within its sensing range. Robots move with constant speed in a two-dimensional arena and monitor continuously the presence of nearby objects in all directions with their sensors, adjusting their movement direction according to an algorithm based on *collision avoidance*, *flock separation* and *flock cohesion* rules. The authors verified with experiments that using appropriate values of the algorithm parameters a flocking behavior emerges from local robot interactions.

Turgut et al. [56] implemented flocking with real robots, equipped with an infrared sensing system to measure the distance and relative orientation of nearby robots and obstacles, and with a digital compass and a short-range wireless communication module to measure their heading direction and the direction of their neighbors. Flocking is obtained by combining proximal control and heading alignment. The proximal control behavior uses the infrared sensing system, which is capable of distinguishing neighboring robots from obstacles, and tries to keep a desired distance from neighboring robots and to avoid obstacles; the heading alignment system measures the heading direction with the digital compass, periodically transmits this information with wireless messages, and tries to align a robot with the average direction of its neighbors, calculated from messages received from the neighbors. The combination of proximal control and heading alignment produces a desired heading direction, which is translated into actuator commands regulating robot movements.

Ferrante et al. [66] obtained a flocking behavior using uniquely proximal control, without alignment control. In their system, the implementation of proximal control is done analogously to that in [56]; the novelty of this approach lies in the motion control formulas which translate a desired velocity vector into forward and angular velocities. In these formulas, robots tend to move forward due to a bias term in their forward velocity, but can also move backwards; as shown by experiments in simulation and with real robots, this motion control method is able to generate an emergent global direction of movement.

In [60], flocking in a three-dimensional space is implemented with real flying robots. The control algorithm of the robots incorporates a repulsion force to avoid collisions and an alignment force to align the heading directions of neighbors; relative position and heading of neighbors are obtained via local wireless communication. The alignment force is defined as a viscous, friction-like force, which according to the authors prevents possible instabilities of the swarm due to noise and delays in sensing and communication.

4.3. Analysis

4.3.1. Metrics

Various metrics have been proposed to give a quantitative assessment of flocking performance. A common and intuitive metric is given by the distance covered by the center of mass of the robot swarm [56,65], where the center of mass at a given time is calculated as the average of robot positions. In [62], where robots are located in a square arena and are tasked with moving from one corner of the arena to the opposite corner, the system performance is defined as a combination of the time required to complete the task, the average distance traveled by the robots, and the average inter-robot distance; these factors are combined to form a cost metric that is used to learn optimal values of robot control parameters. Baldassarre et al. [63] used three statistical measures to characterize individual and group behavior in a flocking swarm: a "group stability index", a "group role index" and a "rotational index". In [17], the flocking behavior is described by measuring the spatial density and the distribution of heading direction values at specific time instants after the beginning of experiments: by taking different "screenshots" during the course of an experiment, insightful representations of the swarm behavior can be analyzed. In [60], the *coherence* of flocking is calculated by comparing the heading direction of each robot with the direction of any other robot in the swarm. In [56], Turgut et al. used a number of metrics to evaluate the performance of their robot control algorithm: the *swarm order*, given by the sum of the heading direction vectors for each robot (normalized with respect to the swarm population), with values approaching 1 indicating an *ordered state*, while a swarm in *disordered state* is characterized by a value approaching 0; the variation with time of the entropy of the swarm, where the entropy is calculated defining clusters via a maximum value of inter-individual distance and counting the number of robots belonging to each of these clusters (a good flocking behavior is one where the change of entropy over time is approximately zero); the swarm velocity, i.e. the average velocity of the center of the swarm during an experiment; and finally, the number of robots in the largest cluster that is not fragmented during flocking.

If the flocking task is defined so that robots should move toward a global goal direction, performance metrics must incorporate this direction. For example, in various studies [57–59,66] the *accuracy* value measures how the average direction of robots differs from the goal direction, and in [66] the *effective traveled distance* projects the vector representing the displacement of the center of mass of the swarm on the vector indicating the goal direction.

4.3.2. Models

Starting from the work of Reynolds [55], various mathematical models of self-organized flocking have been proposed to study the dynamics of natural flocks or to inspire the design of robot controllers able to reproduce this collective behavior in artificial systems. These models are all based on the common principles of inter-robot attraction and collision avoidance, and can include additional factors such as velocity matching and presence of a target direction (which may be known by all robots or only by a fraction of them). Equations governing swarm behavior can be expressed either via functions of macroscopic quantities (such as in [17], where a swarm is characterized in terms of local density of particles), or more often, as control laws for individual robot movements. However, calculating analytically a performance measure of the flocking behavior achievable with a given robot control algorithm is generally not possible, and to evaluate the performance of an algorithm existing works have made extensive use of computer simulations, in addition to performing experiments with real robots.

5. Foraging

The collective foraging task, inspired by the behavior of ants in colonies, is another commonly studied scenario in swarm robotics. Ants and other social animals are able to efficiently exploit food sources using local interactions between individuals. In an artificial swarm robotics system for the foraging task, a specific area is designated as the "nest", and the objective of the swarm is finding items scattered in the environment and bringing them to the nest. Multi-foraging is an extension of the foraging task in which different types of items must be collected and each item is delivered to a nest specific for the item type. Practical applications of this type of tasks include demining, hazardous waste cleanup, search and rescue, and planetary exploration.

Various studies analyzed the dynamics of the swarm energy resulting from the foraging activity: items collected by the robots and brought to the nest, similar to food sources, bring energy to the swarm, while the search activity entails an energy loss. In order to maximize the net energy income, the control algorithm of each robot should determine when the robot searches the environment for items to bring to the nest and when the robot stays idle. Since this aspect is more relevant to dynamic task allocation between robots than to the foraging activity itself, it is not considered in this section, but past works proposing solutions for this problem are surveyed in the section dedicated to task allocation.

5.1. Methods

The foraging task can be decomposed into a sequence of sub-tasks of two types: a robot is either looking for items in the environment or carrying an item to bring to the nest. In a group of robots, execution of each of these sub-tasks can be facilitated by mechanisms of cooperation between robots. Cooperation can also be useful for mitigating negative effects due to interference between robots and thus for improving the scalability of the system. In order to achieve cooperation, there must be some form of communication between individuals, so that the actions executed by a robot are influenced by the activity of the rest of the swarm. Such communication can be achieved in different forms: via a shared memory, via local modifications of the environment, and with direct exchange of information between agents. The following subsections describe in more detail each form of communication giving examples of relevant past studies.

5.1.1. Shared memory

With shared memory, all robots in a swarm are able to read and write information on a shared medium; a conceptually analogous mechanism is broadcast communication, by which each robot can exchange information with any other robot in the swarm, because the net effect is that all robots can have access to the same shared information. In principle, these mechanisms pose some issues related to scalability and simplicity of individual robots; however, such systems can be useful to analyze the impact of shared information on foraging performance, and can offer insights for other communication mechanisms.

Usually, the main challenge in a foraging task is finding the places of interest in the arena, i.e. the location of items to collect and the location of the nest. Even if a robot has already been in a place of interest in the past, the absence of a global positioning system and/or the inaccuracies in calculating relative displacements may cause the robot to "forget" the path to reach the same place in the future. With a shared memory, robots can communicate their recent experience (e.g. the path they followed to reach a given place) and thus help other robots to localize places of interest; in other words, each robot contributes with its imperfect information to build a shared map of the arena that, although being a partial and imperfect representation of the arena, is nonetheless better than the representation each robot can build alone [67,68]. The trail information that is gradually built by foraging robots can be used not only to follow the same trails to reach the same locations, but also to avoid trails used to reach different target locations [69].

5.1.2. Communication through the environment

Ants and other social animals are known to produce chemical substances called pheromone and use them to mark the environment. With pheromone, swarm members establish a mechanism of indirect communication, which uses the environment as a medium for sharing information. Pheromone presence can usually be sensed from a short distance, and pheromone-mediated communication allows overcoming to some extent the limited sensing and communication capabilities of swarm members. In the foraging task, pheromone can be used, analogously to shared memory mechanisms, to create trails which facilitate finding places of interest (such as rich food sources) in the environment [70–72].

In artificial systems, if robots do not have navigation capabilities that allow them to find quickly the nest, in order to optimize the return path to the nest (whose location is fixed) the environment can be marked statically with a gradient field whose intensity increases near the nest [70,71]. Static gradient fields cannot be considered as a type of pheromone communication, because pheromone intensity in the environment is typically dynamic and determined by the robot activity. However, there are similarities between these two mechanisms, because both produce local modifications to the environment aimed at guiding robots with limited sensing and navigation capabilities.

A mechanism conceptually analogous to pheromone communication is put in place when some robots in a swarm assume the role of "markers" of the environment, by staying at a fixed location and providing to nearby foraging robots an indication of the proximity to the nest and to item sources. This approach works with static robots storing and communicating to foraging robots a value corresponding to the current pheromone level at their location. For example, in [73] the pheromone level stored by "beacon robots" changes over time based on short-range communications with foraging robots, and thus follows the swarm dynamics similar to what happens with physical pheromone dropped by foraging ants; in [74], beacon emitters signal their distance to the nest allowing robots that carry an item to easily return to the nest following a chain of beacon emitters within communication range of each other.

Another mechanism of communication through the environment can be put in place using the items being collected, such as when the process of transporting an item from its source to the nest is done by more than one robot [75]; in some studies [76,77], robots are allocated to different regions of the arena and cooperate with each other by picking up and dropping the items at the boundaries of these regions.

5.1.3. Direct communication

In this context, direct communication refers to a process where robots exchange information directly between each other, often (but not necessarily) explicitly transmitting data to signal a particular status. Usually, according to the principle of local communication, information can be exchanged between nearby robots, which can then act upon received information modifying their behavior to improve the foraging performance.

Direct communication can be used to reduce the effects of interference in potentially crowded areas such as the nest [76], to facilitate finding a place of interest such as an item source [78,79] or the nest [80], or to implement mechanisms of cooperative transport of items [81–83]. Communication can take place also by simply sensing the relative position of nearby robots [84], or even using contact sensing [85].

5.2. Algorithms

In the following, algorithms implemented in past studies dedicated to the foraging task are described in more detail, categorized based on the main mechanism used to achieve cooperative behavior.

5.2.1. Path formation

Path formation refers to building one or more "preferential routes" in the foraging arena so that robots either searching for items or carrying an item can reach their current target in an optimized way. These routes are built incrementally by robots as they execute the foraging activity, and can disappear if the reason for which they were created no longer exists, such as when a source of items is depleted.

In studies where robots can communicate through a shared memory or with broadcast messages, if a robot shares information about the path it followed when reaching a place of interest, such information can be used by the other members of the swarm to optimize their movements. This principle is put in practice in [67]: robots, which are assumed to be able to determine their spatial location in the arena, although with limited accuracy, continually record the path along which they move; when a robot, which initially moves randomly searching for items, finds an item to bring to the nest, it puts in the shared memory the path it followed to go from the nest to the item. Other searching robots modify their movement following the path read from the shared memory, and then in turn share information on their path when they find an item. With this algorithm, a map of preferred directions of movement is gradually built in the shared memory, and allows optimizing the foraging task. In [68], this scenario is implemented and tested with real robots.

In [69], shared trail information is built gradually in a similar manner as in [67]. To address interference problems arising when the path from rich item sources to the nest tends to be overcrowded and robots traveling in one direction interfere with robots in the opposite direction, the basic path following algorithm is adjusted so that robots heading toward a target, in addition to being attracted by the path followed by other robots for the same target, are repelled by paths used for different targets. For instance, a robot carrying an item and going toward the nest will tend to avoid the opposite path followed by robots heading toward the item source.

Methods based on pheromone communication, taking inspiration from ant behavior, are based on marking the environment with signs meant to guide robots through the optimal path to their destination. In [70], a mechanism analogous to pheromone communication is used: robots carrying an item drop "crumbs" on their return path to the nest, while robots searching for items are attracted by crumbs. Hamann and Wörn [71] studied a similar scenario, where foraging robots are guided by pheromone that is dropped by robots when they return to the nest to deposit a collected item. The amount of pheromone dropped by robots is maximum where the item has been collected and decreases going toward the nest; with this mechanism, a pheromone gradient is formed in the arena, and robots follow this gradient when they are searching for items. In [72], pheromone is dropped by robots on their return path only if they detect other items near the location of the item they are carrying.

In various studies, the problem of finding a food source or the nest is addressed with robots forming chains of interconnected individuals which guide other robots in their search activity, similar to pheromone trails. In [74], the nest emits a beacon signal which can be sensed by robots up to a certain distance; a robot that reaches the limits of the beacon coverage area stops searching for items and emits a new beacon, thus extending the search area for the other robots, which may in turn become beacon emitters if they reach the limits of the area covered by the beacon signal of the first robot. With this mechanism, chains of beacon-emitting robots are formed which allow searching robots to explore new areas and efficiently return to the nest. In [85], a chain is formed, starting from the nest, by robots in physical contact of each other, and basic behaviors allow robots to execute actions such as forming and extending the chain, following the chain (e.g. when returning to the nest), and performing excursions in proximity of the chain (to search for items).

In [73], each robot can either engage directly in the foraging activity or become a beacon. Robots acting as beacons form networks starting from either the nest or an item source; each beacon stores an integer number indicating its position in the network, with higher numbers corresponding to higher distances from the nest or an item source. These robot chains create a virtual gradient field, which can be "sensed" by foraging robots via local communication with the beacon emitters and can be used to efficiently reach a location of interest.

The above algorithm is integrated in a subsequent work [84] in an adaptive system capable of utilizing multiple algorithms, dynamically switching at the swarm level from one algorithm to the next, based on the characteristics of the environment. The virtual gradient-based algorithm, which performs better when item sources are near the nest, is executed first by robots when starting the foraging task; if it is not successful (as indicated by all the robots becoming beacons), then a different algorithm is activated, in which robots form a single chain starting from the nest and sweep a circular area centered at the nest. If this algorithm, which is capable of exploring a larger area than the gradient-based algorithm, is not effective either, then random walk is used as a last resort.

5.2.2. Cooperative transport

This section contains a review of past works where the activity of transporting an item from its source to the nest is done cooperatively by more than one robot. With cooperative transport, negative effects of interference in crowded areas such as the nest can be mitigated, and foraging performance can be enhanced by restricting the "working area" of single robots. This section does not cover the task of transportation of large items by groups of robots (which is reviewed in the section dedicated to collective manipulation), but focuses on cooperation mechanisms by which the process of moving an item to the nest is made more efficient with the contribution of multiple robots.

A widely studied cooperative transport mechanism is bucket brigading, where items are passed from one robot to another until they reach the nest. Drogoul and Ferber [81] extended the algorithm described in [70] by providing additional capabilities to the robots: a robot carrying an item signals its status (switching a light on), and robots searching for items are attracted by item-carrying robots and are able to discharge them by picking up the item they are carrying. With these additional capabilities, robots can form chains from sources of items to the nest, where the items are passed from robot to robot; the authors showed that with this mechanism foraging performance improves considerably due to the reduced effects of interference. In [82], if a robot carrying an item detects a robot not carrying any item in front of it, it drops the item and reverses the direction of its movement, implicitly "inviting" the other robot to pick up the item.

A mechanism similar to bucket brigading is territorial division, where robots are assigned different working areas. Goldberg and Matarić [76] described a system where robots are divided into two groups (castes) operating in different areas: the "search caste" operates outside the nest area and brings collected items to the boundary of the nest area, while the "goal caste" takes collected items from the boundary of the nest area and deposits them in the nest. In [77], the arena is divided into a number of working areas corresponding to the number of robots. Each robot is assigned a working area, where it searches for items; collected items are deposited at the boundary of the next working area in the direction toward the nest, except for the robot in the last working area, which drops the items directly in the nest. The authors showed how "invasions" of contiguous areas by robots, due to either localization errors or the mechanism of depositing collected items, are a major source of interference between robots, and determine the existence of a critical number of robots, above which the foraging performance decreases.

In the scenario described by Pini et al. [75], territorial division emerges dynamically as a result of task partition between robots. Items to be collected are grouped in a single location, and after transporting an item robots return to the location where they found the item; however localization errors cause this mechanism to be imperfect, thus robots may not be able to return quickly to the item source. To tackle this problem, each robot transports a collected object for a limited distance instead of traveling directly to the nest, and relies on other robots to complete the transportation task; since the amount of localization errors increases with the distance walked, using this mechanism robots improve their success rate in finding the item source. The distance to cover is chosen autonomously by each robot based on the estimation of a cost function which maps the amount of work executed by the robot to the overall cost of the task to deliver the item to the nest.

Arkin et al. [83] studied a cooperative transport mechanism where one item can be carried by more than one robot at a time; when this happens, the process of bringing the item to the nest is accelerated, thus cooperating robots accomplish the task more efficiently. An explicit communication mechanism is used by robots carrying an item can signal their status, so that other robots are attracted by signaling robots and help them to bring the item to the nest.

5.2.3. Other cooperation mechanisms

Beside path formation and cooperative transport, the existing literature contains also other examples of cooperation mechanisms proposed for the foraging task. Rybski et al. [79] implemented two communication strategies where robots turn on a light source to which other robots are attracted; the purpose of these strategies is to attract robots toward item sources. In the first strategy, called reflexive communication, a robot in the process of picking up an item signals its status with light emission; in the second strategy, called deliberate communication, a robot which detects an item but cannot collect it because it is carrying another item stops near the detected item and signals its presence for a fixed time duration (trying to "recruit" other robots to pick up the detected item) before returning to the nest. In [78], a robot knowing the location of a set of items can recruit another robot when returning to the nest: the recruited robot follows the recruiter from the nest to the item location, thus minimizing the time spent searching for items. In [80], direct communication is used to facilitate reaching the nest: robots indicate their proximity to the nest with a light signal which can be sensed by nearby robots, thus effectively extending the range of visibility of the nest. In [76], to address the high level of inter-robot interference in the nest area a message passing system is implemented: robots carrying an item signal their status with specific messages, and using a distributed algorithm avoid entering the nest area simultaneously.

5.3. Analysis

5.3.1. Metrics

Two widely used metrics to extract a measure of foraging performance from experimental results are the number of items collected in a fixed time as a function of swarm size [67,69,73,75,77,80,82] and the time needed to collect a given number of items [70,76,77,79,81,83,72,84]. In some studies, the cost of the foraging activity (i.e. the total energy consumed by the swarm) is quantified, for example by calculating the distance traveled by robots [76,83]; Krieger and Billeter [78] assigned a different weight to the traveled distance depending on whether robots are carrying an item, with the assumption that more energy is consumed when carrying an item; Hoff et al. [73] included also communication between robots in their cost metric.

Special attention is dedicated to the problem of inter-robot interference, which arises when multiple robots share a constrained space. The incidence of this phenomenon can be quantified as the number of collisions between robots [76,77], or the time spent in all instances when a robot tries to perform a task but is hindered by another robot [78].

5.3.2. Models

In the majority of existing studies, analysis of the performance of a swarm in the foraging task is done with either real robot experiments or computer simulations, while mathematical modeling is usually not done due to its inherent difficulty. However, some notable exceptions exist. For example, in [86] it is shown how a foraging scenario of the type described in [70] can be modeled using stochastic Petri nets and Markov chain analysis; with this model, performance metrics such as the probability to bring to the nest all items placed in the environment within a given time frame can be predicted. In [71], a foraging scenario based on pheromone communication is modeled with a set of partial differential equations, which allow estimating the spatial density of robots in the search state and robots carrying an item, as well as quantifying the flow of items to the nest based on the rate of transitions to the search state.

A mathematical model that studies the effect of inter-robot interference is presented by Lerman and Galstyan [87], who showed that in a non-cooperative system there exists an optimal number of robots which minimizes the time needed to bring to the nest the items contained in the arena: above this optimal number, the interference between robots outweighs the work executed by the additional robots. Thus, many efforts have been devoted in various studies to devising cooperative mechanisms for interference reduction.

6. Object clustering and sorting

Object clustering refers to a task where objects scattered in the environment must be grouped together. Compared to foraging, in the object clustering task there is not a predefined destination place for collected objects, the goal being to place the objects near each other. In a variation of this task, there is more than one type of objects, and clusters must be formed separately for each object type; in this case, the task is often referred to as sorting, because the objects are sorted according to their type.

6.1. Methods

Similar to the foraging task, in most object clustering scenarios robots are assumed to be unable to sense the presence of distant objects, thus the typical behavior when searching for objects is to move randomly; the randomness of movement is either encoded directly in the robot controller or the result of trajectory changes executed by robots when obstacles are encountered. When robots do not have localization capabilities, given that there are no predefined locations where objects must be delivered, the walking behavior of robots carrying an object is usually similar to the behavior adopted when searching for objects, thus clusters are formed at random locations, due to the fact that robots carrying objects move randomly.

Robots used in the clustering task are usually characterized by short-range sensing (by which only nearby objects can be detected) and short-term memory (by which a robot can remember the presence of objects encountered in the near past): at the extreme, a robot can only detect objects at zero distance (such as when contact sensing is used) and cannot remember previous object encounters. Given these limitations, the decision of each robot on whether to pick up or drop an object at any given time or location (which determines the emergence of clusters at the macroscopic level) can only be taken (either deterministically or with a stochastic component) based on the limited representation of the environment allowed by robot capabilities; the algorithms implementing this decision are often based on intuitive considerations, but in some cases [88–90] artificial evolution methods have been used.

In the clustering task, interactions between robots working in the same arena typically cannot be used to boost the performance of task execution, and increasing the number of robots employed for a given task can only lead to a sub-linear increase of global performance, due to the possibility of inter-robot interference. Thus, mechanisms of communication between robots are usually not considered in clustering studies. An exception is represented by scenarios where robots have localization capabilities which allow them to build clusters in a "smarter" way: in this case, coordination between robots becomes important, and communication mechanisms (either direct [91] or indirect [92]) play a role in the global clustering dynamics.

6.2. Algorithms

Most existing literature on clustering and sorting tasks makes use of simple robots with no localization capabilities, where clustering of objects emerges as a mere result of probabilistic interactions with the environment; in other studies, robots have the ability to localize themselves, and clustering is done with a more deterministic approach. These two types of scenarios are described in the rest of this section with an overview of relevant past works.

6.2.1. Probabilistic cluster formation

If robots are unable to determine their location, can perceive nearby objects only from a limited distance, and cannot remember the location of existing clusters, a specific place where objects have to be clustered cannot be chosen based on optimality criteria; nonetheless, it is possible to obtain emergent clusters using local sensing capabilities. As usual for swarm robotics tasks, nature offers inspiring behaviors which can be used as a basis for engineering artificial systems: for example, some algorithms for cluster formation are inspired by the behavior of ants during brood sorting.

An often cited paper in this field is the work by Deneubourg et al. [93], who described a simple algorithm for implementing object clustering by a group of robots. Each robot moves randomly in the arena, and has a short-term memory of the objects encountered during its last steps; each time a new object of a particular type is encountered, robots pick it up with a probability dependent on the number of objects of the same type encountered previously, with larger numbers corresponding to lower probability values. Conversely, a robot carrying an object has a probability of dropping the object which increases for increasing numbers of objects of the same type encountered during the last steps. The authors showed with computer simulations how this simple algorithm leads to the formation in the arena of groups of same-type objects.

In other simulation studies, robots are assumed to be able to sense their immediate vicinity. For example, in [94] robots and objects are placed on a two-dimensional grid of square cells, and a robot, which can move from a cell to another in 8 possible directions, can sense the presence and type of objects in 3 of the 8 cells surrounding its current position: the cell on its movement direction, and the cells on its left and right side. When a robot not carrying any object senses an object in front of it, it picks up the object if the object type is different from objects on either its left or its right; conversely, an object is dropped if its type is the same as the objects on either its left or its right.

In [88], Hartmann studied a scenario where robots controlled by a neural network are placed on a two-dimensional grid of square cells and are able to detect the presence and type of objects located in each of the 8 cells surrounding their current position. The outputs of the neural controller allow robots to move forwards and backwards, to turn left and right, and to pick up and drop objects. Evolving the neural controller parameters with a genetic algorithm, robots can achieve good results in clustering both single-type and multi-type objects.

In [95], the clustering task is executed by real robots equipped with a frontal C-shaped shovel that they can use to push up to two objects at the same time. Robots move in a straight line until either they sense an obstacle, or their shovel indicates that more than two objects are being pushed: in both cases, robots change randomly their direction of movement, but in the second case robots first release the objects being pushed. With this simple mechanism, robots are able to cluster all objects in a single group.

Martinoli et al. [96] developed an algorithm to position objects in a chain formation: robots equipped with a gripper and a set of infrared sensors are able to estimate the size of obstacles in front of them; small obstacles are considered objects to be clustered, while robots and arena boundaries are perceived as large obstacles; when an object is detected, the robot picks it up if it is not carrying one, otherwise it drops the carried object: when a large obstacle is detected, the robot avoids it. With this algorithm, groups of two or more objects are detected as large obstacles, and thus avoided, unless a robot approaches them from an angle such that the first object hides the others; as a consequence of this behavior, clusters are built as chain formations, instead of compact groups, as demonstrated by the authors with real-robot experiments and simulations.

Another set of experiments with physical robots is described in [97]. In this scenario, a robot cannot distinguish between the objects to be clustered and other obstacles such as arena walls and other robots: when the proximity of any type of obstacle is sensed, an avoidance mechanism makes the robot steer away from the obstacle. Moving of objects to be clustered is obtained by deactivating the frontal infrared sensor, so that robots cannot sense the presence of small objects located frontally: such objects are thus pushed until another obstacle is sensed in a direction different from the frontal direction. This approach allows obtaining clusters, but as noted by the authors, when an object is pushed against a wall, it becomes "lost", thus experiments always terminate with a non-negligible amount of objects not clustered.

In the study by Holland and Melhuish [98], robots are able to push objects and sense obstacles, and the clustering behavior is achieved using an algorithm similar to that in [95]. In addition, robots can distinguish objects of different types, and have the ability to pull one object with a gripper when moving backwards. Using these capabilities, the authors demonstrated that robots are able to sort objects of two types, forming an inner cluster of objects of the first type, surrounded by objects of the second type. This spatial organization emerges when robots pull backwards by a specified distance objects of the second type before releasing them.

In [89], the object sorting method of [98] is extended to more than two types of objects, by using a different pullback distance for objects of different types. In order to achieve good results in terms of both separation between types and cluster compactness, the pullback distance values for each type are calculated dynamically by robots based on the rate of encounters of the different object types during the clustering activity. Optimized parameters of the distance adaptation algorithm are found by the authors using an evolutionary approach.

Vardy [99] used simulated robots with vision capabilities to achieve clustering of multi-type objects differentiated by color. Similar to the experiments described in [95], object pickup and deposit is obtained with forward and backward movements using a C-shaped shovel. A robot estimates the presence of objects on its trajectory by counting the pixels of specific colors in its field of view; if it is carrying an object of a given color and the number of pixels of the same color exceeds a defined threshold, then it releases the object. Two variations of the algorithm are presented: in the basic version, in the absence of obstacles robots typically move in straight lines, thus their interactions with the objects are consequence of random encounters; with an enhanced algorithm, robots exploit their vision system to accelerate the clustering activity, by identifying the direction of a target object to pick up or a cluster where to deposit the object being carried, and moving toward the target.

In [100], single-type object clustering is obtained using a simple neural network with 2 inputs and 3 outputs. Robots push objects with a frontal shovel, and are able to detect and count the number of objects in two detection areas, one in immediate proximity and one farther away; the numbers of objects counted in each area are used as input to the neural controller, which selects one of the 3 basic behaviors: *BackUpAndTurn*, *Turn* and *MoveStraightAhead*. Weights of the neural network are calculated with a supervised learning technique.

Gauci et al. [90] used robots with a sensor able to detect the presence of another robot or an object in line of sight; based on their sensor input, robots react by setting a given angular velocity to their left and right wheels, thus determining their direction of movement and the curvature of their trajectory. Objects are moved by pushing them. The optimal values of the angular velocity of the wheels for each possible value of the sensor input (i.e. an object detected, a robot detected and nothing detected) are selected via artificial evolution. The authors showed that object clustering can be achieved with this method provided that robot sensors have a sufficiently long range; interestingly, clustering can be obtained (although less efficiently) also if the sensors are unable to detect other robots or to distinguish robots from objects.

6.2.2. Deterministic cluster formation

If robots are endowed with the capability to localize themselves in the environment, the clustering activity can be performed more efficiently, by moving encountered objects to a fixed location instead of relying on probabilistic approaches. In this case, the presence of multiple robots requires a coordination mechanism, because all robots must agree on a common location where objects are clustered. Thus, while initially each robot may select an arbitrary location and begin clustering in that location, during task execution all robots must eventually converge to a single, shared cluster.

In [91], agreement between robots is achieved by means of direct communication: robots encountering each other exchange their current cluster location, and based on the received information decide probabilistically to switch to a location used by another robot. In [92], no explicit communication is used: instead, robots are able to measure the size of the different clusters being built, and switch their preferred cluster location based on the observed sizes, giving preference to larger clusters.

6.3. Analysis

The clustering dynamics can be seen as the result of the application of a mechanism of *positive feedback* by which an action executed at a given location increases the likelihood that the same action is repeated at the same location in the future. This is an example of *stigmergy*, a process widely observed in nature and replicated in artificial systems where local changes to the environment produced by past actions determine the execution of future actions. In the clustering task, the presence of stigmergic mechanisms can be identified in two complementary processes, i.e. the addition of objects to large clusters (which increases the probability that further objects will be added in the future) and the removal of objects from small clusters (which makes those clusters more likely to be shrunk again at a later time). In [94], Wang and Zhang identified in their implementation a "critical" size such that clusters smaller than this size tend to disappear (i.e. the probability that robots remove objects from them is higher than the probability to add new objects), while clusters larger than this size tend to grow with time.

6.3.1. Metrics

Quantitative measures of task execution performance often used to assess the effectiveness of an algorithm include the number of clusters being created [95–97,91], the average cluster size [96,100], the size of the largest cluster [95,96,100,91,90], and the time for task completion [95]. Task completion can be defined as the achievement of a condition where there is a single cluster per object type (in scenarios where this can be obtained with a significant probability), or the number and size of clusters is "satisfactory" according to given criteria. In [94], the percentage of task completion in the presence of multi-type objects is indicated by a performance metric where each object type can be given a different weight; the metric is calculated by measuring the size of all clusters and putting it in relation with the total number of objects of a given type. A similar metric is used in [92], where only the largest cluster of objects of each type is considered in the percentage of task completion. To evaluate the temporal pattern of task progress during an experiment, Vardy et al. [92] used another metric, called time-weighted completion, which tracks the percentage of completion during the entire experiment.

With multi-type object clustering, the performance of task execution is determined not only by how objects of the same type are clustered together, but also by how objects of different types are separated from each other. In [89,88], these considerations are expressed in metrics that include a compactness component (measuring the quality of clusters of same-type objects) and a separation component (measuring the degree to which objects of different types are separated from each other).

As mentioned earlier, in the clustering task the presence of multiple robots working simultaneously usually does not bring a super-linear increase of performance: on the contrary, inter-robot interference is likely to cause a decrease in the amount of useful work that can be done by each robot. The impact of this phenomenon can be described quantitatively by counting the number of events where the interaction between robots prevents them from doing useful work; as can be intuitively understood, given a fixed arena size the number of such events per robot increases with the number of robots [98], thus the total number of interactions increases more than linearly with the number of robots [95].

6.3.2. Models

One of the few attempts to build a complete mathematical model of the clustering dynamics is represented by the work of Martinoli et al. [96], who developed a probabilistic model and applied it to two different clustering scenarios. The model represents the clustering activity as a sequence of probabilistic events, such as a robot entering the detection area of a cluster, or an object being added to or removed from a cluster; the probability of occurrence of each type of events is calculated from the robot control algorithm using geometric considerations. As shown by the authors, the model is able to predict the clustering dynamics (using metrics such as the average cluster size, the number of clusters, the size of the largest cluster, or the time needed to obtain a single cluster) in good agreement with experimental results.

A detailed formal analysis of the object clustering problem is presented by Kazadi et al. [101]. In their study, the evolution of clusters is expressed as a function of the probability to add or remove an object to/from a cluster; the authors found that the ratio between these two probabilities, which is a function of cluster size since generally both probabilities depend on the cluster size, and is denoted by g in [101], is a fundamental property of object clustering scenarios which determines the equilibrium state of the system (expressed as the number of clusters and their size) starting from given initial conditions. This means that it is possible to obtain the desired clustering dynamics in a given scenario by appropriately choosing the g function in the robot implementation.

7. Navigation

A collective navigation scenario is one where a robot with limited sensing and localization capabilities is able to reach a target in an unknown location with the help of other robots. Studies where multiple robots must navigate to the same location are not considered in this category, where the focus is on scenarios where the target location needs to be reached by a single robot, which can exploit the presence of the other robots to facilitate its task.

7.1. Methods

Given that each robot in a swarm has a limited knowledge of the local environment in which it operates, the key to a successful navigation strategy in a multi-robot scenario is sharing knowledge between robots. Such knowledge usually consists of an estimate of the distance to the target, which can be expressed as Euclidean distance [102–104], as a “hop count” [105–107], or through some other means. Communication between robots is usually direct short-range communication, but can be achieved also indirectly through the environment [107]. Since a robot cannot always assume the presence of nearby robots with which it can communicate or nearby “messages” left in the environment by other robots, typically a random component is present in robot controllers, by which a robot moves randomly until it receives some information that helps it to optimize the navigation task. Once initial information has been received, robot behavior is usually more deterministic, because this information allows it to determine an optimal direction of movement.

7.2. Algorithms

In typical navigation scenarios, a robot seeking a goal location acquires knowledge about the target in an iterative process, moving with small steps following a route toward the target, at each step receiving the information needed to execute the subsequent step. In this section, past studies for the navigation task are categorized based on whether the route from the current robot location to the target is defined at the beginning of robot navigation (or at least at the moment when initial information about the target location is received), or is formed dynamically during navigation.

7.2.1. Static routing

Studies reviewed in the following are characterized by the fact that the route taken by a robot seeking a target location is defined as a sequence of landmarks whose location does not change during robot navigation. This survey is restricted to studies where the location of such landmarks in the environment is not defined a priori (since this would imply the presence of a central entity with global knowledge of the environment), but is randomized based on the movement of robots in the swarm.

An early study that envisions the use of a robot swarm for navigation purposes is the work by Cohen [105], where robots form a grid of interconnected nodes which can guide an agent to an unknown target location. When a robot senses a target, it stops walking and starts sending messages indicating its closeness to the target; in turn, other robots receiving these messages stop walking and start sending other messages indicating their closeness to the first robot. This mechanism replicates in the swarm, thus expanding the region in the arena from which the target location can be easily reached. A similar approach is used in [106], where wireless messages exchanged by robots produce a sort of pheromone gradient, using an analogy with chemical pheromone utilized by ant colonies. Following this gradient, the target location can be easily reached from any position where there is at least one robot in communication range.

Another navigation algorithm where robots position themselves in the environment in order to allow another robot to reach its destination has been proposed by Mullins et al. [108]. In this scenario, the target location emits an audio tone perceived by robots within a limited range; robots are able to measure the sound intensity, but cannot perceive its direction. To reach the target, robots implement a search mechanism inspired by the behavior of bacteria which move toward areas with higher presence of nutrient. A robot alternates between two states: *tumble* and *run*; in the first state, the robot makes a random turn, while in the second state the robot moves on a straight line for a time duration which depends on the variation of sound intensity measured during the move: if the sound increases, the robot moves for a longer duration compared to the case of decreasing sound intensity. With this approach, under the assumption of a bounded arena, the robot is always able to reach the target without help from other robots, even if its initial location is outside the range of the audible tone, but it may require a long time to accomplish the task. To improve the efficiency in reaching the target, a distributed algorithm involving the other robots in the arena is proposed: the searching robot emits a specific sound indicating that it needs to reach the target; robots hearing this signal propagate the signal and begin moving randomly, provided that they are not in the range of the target; if a robot is near the target (as indicated by the perception of the audio tone from the target), it stops and retransmits the tone at a higher frequency; in turn, other robots hearing the new tone stop and retransmit the tone at a higher frequency; in this way, a tree-like structure of robots is formed starting from the target. The searching robot can navigate through this tree with the algorithm based on the tumble and run states described above, changing repeatedly its target during the navigation according to the lowest frequency tone perceived.

The studies described so far employ algorithms in which robots in the swarm are dedicated to the navigation task of another robot.

While this behavior minimizes the effort of the single agent to reach the target location, it also potentially utilizes a large amount of resources in the swarm. For this reason, it is desirable for robots to be able to execute their own tasks while still providing help to other robots for navigation. Wurr and Anderson [107] approached the navigation problem using numbered markers dropped by robots in the arena, by which a robot can reach a target location following the sequence of markers from the highest numbered to the lowest numbered. A robot which locates the target drops a marker at its current location if it perceives no other markers in sight; the number assigned to the marker indicates its closeness to the target. Analogously, a robot which locates a marker and perceives no other higher-numbered markers in sight drops another marker. With this mechanism, trails of markers are formed in the environment, providing a means for a robot to reach the target without requiring other robots to assume the role of static landmarks.

7.2.2. Dynamic routing

In the previous section, the route followed by a robot during navigation is determined at the beginning of navigation, as a sequence of landmarks represented by either robots, or markers dropped by robots. As discussed previously, using robots as landmarks has the disadvantage of dedicating a potentially large amount of resources to a single robot; on the other hand, implementation with real robots of a technique using a mechanism of indirect communication between robots (communication through the environment) presents practical challenges. Methods using direct communication and not requiring static positioning of robots can avoid these drawbacks. In such methods, the route of a robot from its current location to the target is determined dynamically based on the presence of other robots in the vicinity of its current location; these methods are potentially highly flexible, but are characterized by more complex algorithms.

In a study by Sgorbissa and Arkin [102], robots are given sensing and communication capabilities limited to line-of-sight conditions, and use a navigation mechanism based on the following algorithm. A robot which has sensed a target location shares this information with other robots in line of sight, so that robots seeking the target can use the location of the first robot as a temporary target and head toward it in order to get closer to the real target. This mechanism is extended to chains of robots in line of sight of each other, each of which transmits information on the estimated distance to the target, calculated either directly (if the target is in line of sight), or by summing the distance to the next robot in the chain with the distance to the target advertised by that robot. The algorithm does not require robots in a chain to be stationary, and deals with moving robots introducing the concept of "ghost" robots: if a robot seeking a target has sensed another robot which transmits information on the estimated target distance, and if the transmitting robot moves in the environment such that it either becomes invisible to the first robot, or increases its distance to the target, the first robot moves toward the previous location of the second robot, i.e. it moves as if there was a "ghost" robot still transmitting its target distance from that location.

To deal with the constantly changing connectivity structure between communicating robots in a swarm (due to the fact that robots move and their communication range is limited), Ducatelle et al. [103] proposed solving the navigation problem using a dynamic routing algorithm where the route between a robot and a target location (expressed as a sequence of intermediate nodes) is continuously updated and improved based on the current configuration of the swarm; the algorithm requires the capability of robots to estimate the relative position of neighboring robots, and is based on the transmission of short-range wireless messages that carry routing information associated with path quality information; path quality is measured from the relative distance between neighboring robots. In subsequent studies [109,110,104], the authors used for the same scenario a routing algorithm where navigation information is dynamically adjusted by each robot based on odometry measurements; this algorithm is shown to be effective also in situations with intermittent connectivity between robots, such as cluttered environments with low robot densities.

In [111], the collective navigation strategy is based on a signal propagation technique. Robots in the detection range of the target emit a light pulse for a fixed time duration; neighboring robots are able to detect the light pulse, and in turn emit the signal; after signal emission, each robot enters in a "refractory" state, where it is insensitive to received signals. If there is at least one robot in the detection range of the target and the robot density in the arena is sufficiently high, this mechanism produces signal waves propagating from the target through the swarm. Each robot is able to determine the direction of a detected light pulse, and can thus navigate to the target by following this direction.

7.3. Analysis

The most immediate measure of the performance of a navigation algorithm is given by the time needed for a robot to reach the target location [107], often compared to the time needed with a non-cooperative method such as random walk [108]. While non-cooperative methods may provide a baseline to assess the advantages of a distributed algorithm, an upper limit to the efficiency of a navigation method is found by considering the shortest path between a given location and the target, thus some studies evaluated the performance of their proposed methods by calculating the ratio between the traveled distance and the length of the shortest path [103], or by comparing the time to reach the target with the time necessary to follow the shortest path [110,104].

Since the presence of nearby robots in the environment provides an opportunity for a robot to learn information useful to reach its target location, the performance of a distributed navigation algorithm is expected to improve with a larger swarm size; this is confirmed by various studies [102–105,109,110] where the average time to reach the target is plotted as a function of the swarm size and is shown to decrease with increasing numbers of robots. In scenarios where the navigation process of a robot is guided by other swarm members modifying their behavior in order to assume the role of landmarks, the decreased time to reach the target location is obtained at the cost of a potentially significant use of resources in the swarm; in [108], this cost is quantified as the total time committed to the navigation process of a single robot by all swarm members, and with simulation experiments is shown to exceed the average time the single robot would need to reach the target using a simple random walk technique. For these reasons, when implementing a distributed navigation algorithm a tradeoff between navigation performance and total use of swarm resources may need to be considered.

8. Path formation

Path formation in swarm robotics refers to a process where robots are able to build collectively a path between two locations in the environment, so that the time needed to reach one location from the other is minimized. This task can also be referred to as chain formation, because often the path is marked by a chain of robots, either stationary or moving. As described in the section dedicated to the foraging task, path formation mechanisms can be observed in many instances in the foraging context, because often robots share locations of interest and thus can benefit from sharing information on how to reach those locations. Studies reviewed in this section focus on scenarios with two target locations, where robots must be able to move efficiently from one location to the other.

8.1. Methods

In past studies, the path formation problem has been addressed using mainly pheromone-based, probabilistic, and evolutionary methods. Pheromone communication has been extensively studied for the foraging task, where it is used to dynamically build optimized paths between the nest and rich food sources [70,71,73]; a mechanism analogous to the use of pheromone is put in place when robots exchange local messages whose content provides an indication of the current position of communicating robots with respect to the path being formed [112]. In probabilistic methods, a chain of robots is created as a sequence of stochastic events determined by robots continuously joining and leaving the chain; this dynamic process allows robots to explore new areas in the environment until both locations of interest are found and an optimized path between them is formed [113]. In evolutionary methods, robot behavior is evolved according to a fitness function that measures the quality of a path established between two locations: for example, in [114] robots are controlled through a simple neural network whose parameters are evolved with a genetic algorithm, and the fitness function rewards robots based on how many times they travel from one target location to the other.

8.2. Algorithms

The following brief survey describes some examples of path formation algorithms differentiated based on whether the path between the two target locations is marked by a chain of stationary robots, or is sustained by a continuous flow of robots moving along the path.

8.2.1. Stationary robot chains

A stationary robot chain is formed when one target location is connected to the other through a sequence of robots, where each robot in the chain is within sensing or communication range of its two neighbors. A robot that is not part of the chain can move from one location to the other by simply following the sequence of stationary robots. An example of this type of chain formation in the foraging context is the work by Werger and Matarić [85], where a chain is used to connect the nest with a food source.

The algorithm proposed by Szymanski et al. [112] to find the shortest path between two locations is inspired by the work of Payton [106], where a virtual pheromone concept is used to provide navigation information by means of a network of interconnected robots. After a first phase where robots uniformly spread in the arena, a “negotiation” phase is started where neighboring robots exchange messages via line-of-sight communication; the messaging algorithm allows each robot to know whether it is on the shortest path and to transmit this information to other robots. As demonstrated by the authors, this algorithm works also in challenging environments such a maze-like arena.

In [113], the path between two locations is formed by robots communicating through colored LEDs and sensor cameras. The algorithm is based on first finding one of the two locations (the “nest”) and then progressively forming sequences of intercommunicating robots departing from the nest, until the other location is found. Two variations of the algorithm are proposed: with the first variant, one or more linear chains are formed starting from the nest until one of them comes in contact with the other location; in the second variant, a robot can join an existing chain at any position, so that tree-like structures can be formed. In both cases, robots at the end of a chain can leave the structure at any time, using probabilistic rules. The task is considered as successfully completed when a robot in one of the chains departing from the nest senses the other target location.

8.2.2. Robot flows

Studies discussed under this category are characterized by the fact that path formation does not require the continuous presence of stationary robots “marking” the path, but is the consequence of a positive feedback mechanism triggered by robots moving repeatedly between the two target locations, so that a robot following a path at a given time increases the likelihood that other robots will follow the same (or a similar) path in the future. This positive feedback mechanism can be observed in many studies dedicated to the foraging task, where often different robots share the same path when heading toward the nest or toward a rich food source. Cooperation methods based on a shared memory [67] or pheromone communication [70,71] are examples where this mechanism is applied.

An interesting study where path formation is achieved using only direct robot-to-robot communication is described in [114], where robots are controlled by a neural network that regulates their movement via two differentially driven wheels and operates two colored LEDs used for signaling robot position and direction of movement; inputs of the neural network are a sensor camera and infrared proximity sensors. Selecting the parameters of the neural controller with an evolutionary method, the authors obtained the emergence of a collective behavior where robots form two parallel “traffic” flows between the two target locations, repeatedly traveling from one location to the other. Such evolved behavior is made possible by the interaction between robots (through their colored LEDs), which allows the path between the target locations to be sustained by the continuous flow of robots; in this respect, an interesting analogy can be found with pheromone-based trail formation in ants.

8.3. Analysis

The path formation problem offers an interesting example of how a swarm of robots allows performing a task whose difficulty exceeds the individual robot capabilities: even though robots are unable to sense simultaneously the two target locations (due to their limited sensing range) and do not have localization capabilities that would allow them to determine their position relative to a location of interest, the dynamics of the swarm generates a “collective perception” of the environment that individual robots can exploit.

In studies where the path between two locations is established by a chain of stationary robots, the path formation task can be considered complete when such chain extends from one location to the other; the performance of this type of systems can be evaluated by measuring the completion time, i.e. the time elapsed from an initial configuration with randomly placed robots to the successful creation of the chain. In [113], the completion time is evaluated as a function of the distance between the extremes of the path, and as a function of the swarm size, showing that while a larger distance usually implies a more challenging task, a larger swarm size allows tackling better the difficulty of the task: calculating a measure of the overall effort by multiplying the completion time with the swarm size, the authors demonstrated a super-linear increase of performance when increasing the number of robots.

If the path between two locations is sustained by a continuous flow of robots moving in either direction, two separate phases can be identified in the swarm dynamics: first, randomly distributed robots “explore” the environment searching for the two locations of interest and establishing a path between them (initial coordination period), then the established path is sustained (and potentially optimized) by the robot flow. Sperati et al. [114] evaluated their path formation method by calculating a measure of energy that takes into account the number of times a robot travels from one location to another in a given time interval, compared to the maximum attainable number given the distance between the two locations and the robot speed; this evaluation, done after the initial coordination period (whose duration is chosen arbitrarily), gives a measure of the optimality of the chosen path and the ability of robots to follow it without negatively interfering with each other. From results of multiple experiments, the authors found a correlation between the path distance and the optimal swarm size, indicating that a larger swarm performs better with a larger distance.

9. Deployment

In a self-deployment scenario, robots must deploy themselves in an environment without central coordination. This task has potentially many practical applications, ranging from mapping of unknown environments to autonomous surveillance systems.

9.1. Methods

9.1.1. Direct communication

Direct robot-to-robot communication is the most used mechanism to achieve cooperation in self-deployment tasks. Communication can take place using explicit messages [115–119], or implicitly, by sensing the nearby presence and relative position of other robots [106,115,120–127]. Information acquired from nearby robots can be used to implement simple mechanisms of robot avoidance [123], or, more often, to regulate the position and velocity of a robot according to a desired behavior. In the latter case, robot movement can be determined following principles of artificial physics [120,121], with the objective of preserving connectivity between swarm members [106,116,117], or to obtain formations described by a specific geometric relationship between neighboring robots [124,126,119].

9.1.2. Stigmergy

Stigmergic communication gives robots moving in an area of the environment an indication of the actions done previously by other robots in the same area. In various scenarios (e.g. in the foraging task), it is used with a positive feedback mechanism, i.e. an action done by a robot increases the probability that the same action is repeated by other robots. Conversely, for the deployment task a negative feedback mechanism can be put in place, preventing different robots from repeating the same action, and specifically preventing the same areas of the environment from being explored multiple times, or (in tasks where the environment must be covered repeatedly) maximizing the time between two successive visits to the same place.

Stigmergic communication has been implemented in past works using simulated pheromone traces [128–132]: the presence and intensity of pheromone at a given location is used as an indication that the location has been visited before. In some studies, pheromone is assumed to evaporate over time, analogously to what happens in nature with chemical traces, and this property is used to optimize repeated coverage of the same area [128], or to dynamically assign non-overlapping patrolling areas to different robots [132].

9.2. Algorithms

A survey of past works in the deployment task is presented in the following, where two main variations of the task (dispersion and pattern formation) are identified and used to categorize the different studies.

9.2.1. Dispersion

In the dispersion task, swarm members must position themselves away from one another, with the objective of maximizing the area covered globally by the swarm and/or minimizing the time needed to cover the area. A robot dispersion technique can be applied for example in scenarios where robots must find either particular locations in the environment or objects located in unknown places (as in the case of foraging robots), thus the dispersion task can be used as a sub-task of more complex activities. In some cases, an additional constraint is given by the requirement that the connectivity of the swarm must be preserved, i.e. each robot must be able to sense or communicate with at least another robot so that there are not isolated groups; the practical effect of this constraint is to limit the maximum area that can be covered simultaneously by the swarm.

It is intuitively understood that programming robots so that they avoid each other while moving in the environment increases the capability of the swarm to cover a large area, compared to a simple random walk technique. Morlok and Gini [123] evaluated the performance of four basic algorithms (random walk, wall following, avoiding all obstacles, and avoiding other robots) in maximizing the area covered by a swarm of robots in closed environments. From simulation results, the algorithm where robots avoid each other performs better than the other algorithms, indicating that knowledge of the location of nearby robots is useful for dispersing the swarm. Also the study done by Batalin and Sukhatme [115] gives experimental evidence that control algorithms where nearby robots are distinguished from other obstacles perform better than schemes where robots are simply considered as obstacles.

To overcome the problem where a robot with limited sensing abilities is unable to determine whether a given location has been visited before, various stigmergic techniques have been studied. Wagner et al. [128] devised some algorithms where robots drop evaporating pheromone along their path, and when choosing their walking path give precedence to areas with the lowest pheromone level. The techniques described in [128] rely on partitioning the environment in a discrete set of tiles, and modeling the partitioned arena as a graph where vertices represent the tiles; in [130], an analogous algorithm is proposed, which operates on continuous domains as opposed to graphs. In [129], experiments with real robots are reported, where the pheromone is represented by ink trails drawn by robots along their path with a pen. Kuyucu et al. [131] used a genetic algorithm to evolve a set of parameters (e.g. pheromone production rate) which influence the swarm performance in the deployment task: according to simulation results, parameter values obtained with the evolutionary method lead to better performance compared to manually tuned values. In [132], stigmergic communication allows a group of robots to coordinate to dynamically partition an area in contiguous territories, with each territory patrolled by one robot; an adaptive variant proposed by the authors to the basic algorithm is shown to allow swarm members to dynamically learn an optimal size of their respective territories based on the arena size and the total number of robots.

In the context of navigation, Payton et al. [106] described a deployment technique based on short-range infrared communication, where robots are able to approximately measure reciprocal distances while within communication range of each other. In the "gas expansion" model, robots initially grouped in a small space explore the environment trying to maintain a target distance with neighboring robots in order not to lose connectivity. If the total number of robots is not sufficiently high to cover the entire arena, this model can be complemented by a "guided growth" model, such as when one robot starts to move away from the swarm to explore uncovered areas of the arena and doing so "pulls" the entire swarm, because the other robots follow the exploring robot in order to keep connectivity with it and between each other. A similar approach for robot dispersion and for exploration of new areas is presented in [116].

As opposed to tasks where robots must continuously move in order to cover a given area (typically employed when the arena size is much larger than the union of the sensing areas of all robots in the swarm), other studies focus on tasks where the swarm reaches an equilibrium status with robots occupying static positions. To regulate the mutual distance between robots where the objective is to disperse the swarm, some studies utilized artificial physics methods based on the concept of virtual potential fields and virtual forces. Reif and Wang [133] introduced the concept of *social potential fields*, which reflect the "social relations" between robots performing distributed tasks. Howard et al. [120] proposed a control law for the velocity of robots based on a potential field determined by the presence of other robots and obstacles. Each robot in close proximity of other robots or obstacles is repelled by nearby entities, and moves according to the virtual force determined by this repulsion. This mechanism leads the swarm to optimize the occupation of the arena according to the total number of robots. Podury and Sukhatme [121] used potential fields to maximize the area covered by a swarm of robots with defined sensing and communication ranges, with the constraint that each robot must stay within communication range of a minimum number of other robots. In [127], the task of maximizing the area covered by a swarm of connected robots is tackled with an automatic design method using probabilistic finite state machines, where parameters of robot controllers are selected with an optimization algorithm.

In many deployment tasks, and especially in those using potential field approaches, it is assumed that a robot can measure with a reasonable precision the distance and relative orientation of nearby robots. When using real robots, this capability is usually offered by infrared technology, which offers line of sight communication with highly directive signal radiation patterns with known attenuation characteristics. In [117], the swarm deployment task is performed by robots using radio frequency communication, characterized by a much less predictable mapping between signal strength and distance; in addition, relative orientation cannot generally be inferred from the received signal. Despite these difficulties, the algorithm proposed by the authors is successful in dispersing a robot swarm in the environment.

9.2.2. Pattern formation

Pattern formation is a variant of the deployment task where robots occupy relative positions such that when viewed globally their ensemble can be described by a specific pattern. Such formations can be used for example in surveillance tasks where each robot is assigned a specific area to be monitored, and the swarm must prevent situations where there are uncovered spots. The capability of a robot to measure the relative distance and orientation of its neighbors allows a high degree of flexibility in determining the desired positions of neighbors, from which multi-robot formations can emerge. Thus, by using local rules, if each robot in a swarm positions itself with the purpose of obtaining a desired distance and orientation with respect to neighboring robots (i.e. forming a geometric shape with its neighbors), at a global level the swarm can converge to a state where it is deployed optimally in the environment.

In [122], an extensive analysis is performed on the dynamics of formation of different patterns with robots controlled by virtual forces. The authors described how two- and three-dimensional hexagonal lattices of self-controlled particles can be obtained using attraction and repulsion forces. In addition, particles are subjected to a viscous friction force, which is proportional to the particle speed and whose purpose is to avoid continuous oscillations around an equilibrium state. The authors demonstrated with computer simulations that starting from particles in random locations the system can converge to a state where particles form a hexagonal lattice. Mikkelsen et al. [125] used a similar virtual force-based approach to obtain hexagonal lattice formations; their model includes the effects of communication errors between neighboring robots which occur due to the limited range and non-uniform coverage of the infrared sensors. Another study utilizing virtual forces is the work by Mathews [118], where formation of triangular grids is obtained; this pattern is shown to optimize area coverage by robots with given sensing and communication ranges, provided that the communication radius is sufficiently large compared to the sensing radius.

In [126], each robot chooses two other robots among its neighbors, and then positions itself so as to form a triangular shape with those neighbors. The distance between the robot and its neighbors is chosen based on a measured local characteristic of the environment. If all robots operate with the same algorithm and if the environment characteristic which determines the distance has the same value in the entire covered area, this technique leads the formed triangles to be equilateral, and thus a regular mesh pattern is observed at the swarm level. In [124], an analogous technique is used in a three-dimensional space, where each robot selects three neighbors and tries to form a tetrahedron.

In [119] desired formations are obtained by defining weight functions which vary with the spatial position relative to robots. The control law used by each robot for achieving a specific formation is determined with a consensus-based approach by calculating the corresponding weight function on the relative position of each neighbor. As opposed to methods based on potential fields, a consensus-based approach is more robust in reaching an "agreement" between neighboring robots on their relative positions, even in the presence of communication delays.

9.3. Analysis

9.3.1. Metrics

Evaluation of the performance of a self-deployment algorithm is usually carried out by executing experiments where robots are initially either clustered at a specific place, or scattered randomly in the arena. In scenarios where the arena size is much larger than the union of the sensing areas of all the robots in the swarm, the deployment task is executed with robots visiting in succession different areas of the environment until the entire arena is covered. Empirical results to evaluate how a given algorithm works in practice can be expressed as the percentage of arena space visited by robots in a given time [115,123], or as the time needed to cover the entire arena [128–131]; often, these metrics are expressed as a function of the number of robots, to evaluate how inter-robot interactions affect task execution.

In scenarios where the goal of the deployment task is to reach an equilibrium state where the swarm assumes a static configuration, a good indication of the performance of an algorithm is obtained by analyzing the temporal evolution of the area covered by the swarm [120,121,117,132,118]: a faster dynamics where the swarm reaches the equilibrium in a small time usually indicates a more performing algorithm. To evaluate the quality of the equilibrium state, especially in pattern formation techniques, some useful measures are the distribution of inter-robot distance [125] and the mean square error between desired and actual distance [119].

9.3.2. Models

In various studies proposing self-deployment algorithms, researchers have been able to demonstrate with formal analysis the validity of their proposed approaches. For example, for the dispersion task Wagner et al. [128] and Osherovich et al. [130] calculated analytically an upper limit to the time needed for a swarm of robots to cover the arena with their self-deployment algorithms; for the pattern formation task, Lyapunov theory has been used in [126,124] to prove the ability of the swarm to converge to an equilibrium state characterized by the desired spatial relationships between robots.

In [129], where robots are programmed to cover an arena repeatedly, the deployment task is modeled with a modified version of “node counting”, which is an algorithm for executing searches on a connected graph; even though node counting operates on a discrete space, it is shown to provide a theoretical foundation to predict the performance of the proposed robot deployment algorithm, which operates on a continuous domain. Moreover, the authors reported an interesting analysis of the effect of pheromone marking: if pheromone dropped at a given place does not evaporate, a saturation effect causes the performance of area coverage to decrease with time, thus the average time needed to cover the entire arena increases with the number of times the arena is being covered; if pheromone is subject to evaporation, this drawback can be partially avoided. The choice of the evaporation time is thus subject to a tradeoff between the ability to detect previously visited areas and the ability to estimate when a given area has been visited last.

10. Collaborative manipulation

Usually, swarm systems allow agents to execute collective tasks more efficiently than each individual alone can do; in some instances, the task at hand cannot be executed by any single individual, but requires cooperation between multiple individuals. A typical example taken from insect societies is the retrieval of large food items by groups of ants: depending on the item size, this task may require a large number of ants, which must work in coordination in order to bring the task to successful completion (i.e. transport the food item to the nest). Observation of ant behavior has shown that such coordination is achieved without central control and with each ant using simple rules governed by local interactions with other ants and with the environment. Even though the task may not be completed with optimal efficiency (e.g. multiple unsuccessful attempts may be made by small groups of ants before recruiting other ants), the absence of global knowledge and centralized control allows the system to be fault-tolerant and to work with simple individual agents.

Collaborative manipulation refers to swarm robotics tasks where groups of robots work together to manipulate objects in the environment. As shown by ant behavior, such tasks can be done with robots obeying simple rules without central control, and thus are another example of global dynamics obtained with swarm intelligence principles. In the rest of this section, some past works on collaborative manipulation tasks are presented.

10.1. Methods

As often seen in swarm robotics, a complex task involving multiple robots can be executed without explicit coordination mechanisms, with robots engaging in behaviors seemingly unrelated to the task at hand. For example, in [134] a simple *follow* behavior in which a robot is attracted by other robots moving in front of it modifies the otherwise random distribution of robots in the arena so that robots are able to cooperatively push a box. In other works, cooperation arises with robots simply engaging in the same behavior, such as when a box must be pushed to a given goal location and each robot is able to sense the goal [135,136]; however, it must be noted that in this case to be able to engage in the same behavior all robots need some global knowledge of the environment (specifically, the goal location).

When ants are not able to transport a large item because of its weight or size, they recruit other ants by moving toward the nest and secreting pheromone, thus laying down a trail. Other ants sensing the presence of pheromone move along this trail, going toward the item to be transported. A similar principle can be applied in artificial systems, as demonstrated in [137] where pheromone communication is used by robots to transport large items. An alternative means to recruit workers is studied in [138], where explicit communication between robots is implemented with LED signaling. In other studies [139,140], collaborative object transport is obtained engineering robot controllers with artificial evolution.

A common characteristic found in object manipulation problems is that robots in proximity of an object that cannot be manipulated because of an insufficient number of nearby robots tend to stay in proximity of the object, waiting for other robots to arrive. In scenarios where the number of objects is high compared to the number of robots needed to manipulate them, this behavior can lead to a stall where all robots are waiting for other robots. To avoid this potential deadlock, some object manipulation studies [141,142] adopted a mechanism by which robots can abandon the object they found and restart searching for other objects with a potentially higher number of nearby robots.

10.2. Algorithms

Two main types of object manipulation problems can be found in the existing literature: tasks where robots must collectively transport large objects, and tasks where sticks must be pulled from the ground. The rest of this section contains a survey of a representative sample of past works addressing these problems.

10.2.1. Object transportation

A frequently studied example of collaborative manipulation is the box pushing task, where robots must move boxes located in the arena by pushing them; the size of each box is such that it cannot be pushed by a single robot alone, but requires multiple robots pushing in the same direction.

Without a preferred direction, different robots may push the box from different angles, thus negatively interfering with each other. A simple mechanism to decrease the probability of negative interference is by using an attractive force between robots. In the scenario described in [134], a group of robots must find a box located in the arena and push it at one edge of the arena. The robots are programmed with two basic control mechanisms: first, a common goal with non-interference, by which robots are attracted by the box and avoid collisions with each other; second, a follow behavior, by which robots tend to follow other robots located in front of them. The second mechanism is found to increase the probability that the distribution of robots around the box is asymmetric, meaning that the overall force exerted by all robots makes the box move. With these two control mechanisms, the robots are shown to be able to collaboratively push the box.

As described above, if in a box pushing task there is not a specific direction along which the box should be moved, different robots may push it from different directions, thus negatively interfering with each other. If there is a predefined goal location to which the box must be moved, and if robots are able to determine the direction toward the goal, this negative interference can be avoided. In [135], the goal location is identified by a light source, and robots are able to move the box to its goal by pushing it only if they sense the light emitted by the goal with their frontal sensors, and repositioning themselves otherwise. In [136], robots are equipped with a vision system which can sense the goal from different angles, and the height of the box to be pushed is such that the box occludes the view of the goal location when robots are pushing along the correct direction. Exploiting this fact, the robot control algorithm ensures that robots push the box only when the goal location is not in their field of vision.

Fujisawa et al. [137] used pheromone communication to replicate ant behavior with real robots, utilizing ethanol as an artificial pheromone: robots are equipped with a tank and a pump for laying down ethanol trails, and with a sensor able to perceive the presence of ethanol. According to experimental results, communication between robots through artificial pheromone improves the performance of the robot swarm in transporting large items.

In other studies, robots are equipped with grippers (which can be attached to other robots or to the object to be transported) mounted on a rotating turret, so that object transportation can be done on an arbitrary direction compared to the relative orientation of robots with respect to the object. Using their grippers, robots can physically connect to each other and to the object, and thus exert force on the connected entity. In [138], large items are transported by a group of robots surrounding the item. In order to recruit other robots for the transport activity, robots use direct communication with light signals: a robot which perceives an item to be transported turns on an LED; if another robot searching for the item senses the LED, it stops searching, follows the robot with the LED on and in turn activates its LED, thus potentially recruiting other nearby robots. When the item is surrounded by robots, it can be transported collaboratively. Groß and Dorigo [139,140] used robots controlled by a neural network whose sensory capabilities include an omnidirectional camera which can sense objects to be transported, other robots and the target location for the objects. By selecting neural network parameters with artificial evolution, and assuming that the target can be sensed from any location in the arena, the swarm is shown to be able to transport to the target location objects of different sizes and shapes.

10.2.2. Stick pulling

Another example of collaborative manipulation is the stick pulling task [143], where cylindrical sticks placed inside holes in the environment must be pulled outside the holes by robots equipped with grippers; the length of the sticks and the capabilities of robots are such that two robots are needed to pull a stick, thus this task requires collaboration between robots. In [141] it is shown how this task can be accomplished by robots with local sensing capabilities and use of implicit communication. A stick must first be pulled half-way by one robot, which waits (up to a timeout) for another robot to arrive and complete the pulling operation. If the second robot arrives before the timeout, the stick is pulled successfully, otherwise the first robot abandons the stick and restarts searching for other sticks. Since the robots do not know the location of sticks in advance, the execution performance of this task depends on the random interactions between robots and the environment.

In [142], the stick pulling task is executed with reinforcement learning techniques where the timeout parameter value is updated dynamically based on a reward determined by successfully pulled sticks. Both local and global reinforcement schemes are analyzed. Experimental results show that learning improves the swarm performance with both schemes; local reinforcement, where each robot updates its timeout independently, is found to be more effective because it leads to specialization.

10.3. Analysis

10.3.1. Metrics

In simple tasks where the objective is to find an item in the arena and transport it to a predefined location, the performance of an object manipulation algorithm can be measured by the time needed to transport the object to its destination [136]; if the nature of the task allows implementing a successful collaboration mechanism, this metric, when evaluated as a function of the number of robots, shows decreasing values for increasing swarm size [137], while in the presence of inter-robot interference task completion time tends to increase with the number of robots [135]. To measure the quality of swarm behavior, Chen et al. [136] used the path efficiency, defined as the ratio between the length of optimal path from the initial object location to its destination and the length of the path actually followed by the object. If experiments are run for a fixed time, the distance by which the transported object approaches the target location during an experiment [139,140] and the task success rate [138] provide a good indication of swarm performance.

Robot implementations for the stick pulling task are usually evaluated by measuring the collaboration rate, defined as the number of sticks successfully pulled from the ground per unit time [141,142,144]. Plotting the collaboration rate as a function of swarm size not only indicates that more robots are able to pull more sticks, but also shows a super-linear increase of performance, because the collaboration rate per robot increases with the number of robots. Conversely, the failed collaboration rate, i.e. the rate of occurrence of events where a robot abandons a stick, is shown to decrease with increasing swarm size [141].

10.3.2. Models

In general, collaborative manipulation tasks require multiple robots to be located in the same place at the same time, and can be modeled as stochastic processes, whose dynamics is regulated by the characteristics of the environment and the robot control parameters. Probabilistic models for the stick pulling task are proposed in [141,144], and are shown to provide good agreement with experimental results in characterizing the swarm performance. Defining an experiment as a series of stochastic events (e.g. when a robot encounters a stick or another robot), with probabilities calculated based on geometrical considerations, the evolution of the state of each robot can be modeled as a probabilistic finite state machine, whose probabilities of state transitions define the swarm dynamics and allow calculating the expected value of performance measures.

Lerman et al. [145] proposed a generic probabilistic model for predicting the dynamics of a swarm robotics scenario where the execution of object manipulation tasks requires the simultaneous presence of more than one robot. In their model, when a robot encounters an object to be manipulated, it stops near the object and waits for other robots to arrive; if the number of robots required to execute the task arrive within a certain amount of time, then the task can be completed successfully, otherwise the robots near the object time out and restart moving in search of other objects to be manipulated. The rate of successful task execution thus depends on various factors such as robot and object density in the arena, number of required robots for a single manipulation, and the time robots wait before deciding to abandon an object. The authors applied this model to the stick pulling task and showed that it is able to predict the dynamics encountered with simulations and real robot experiments.

11. Task allocation

Task allocation or division of labor in a swarm robotics system refers to the ability to dynamically change the task executed by each robot based on local perception of the environment. With this ability, robotic systems can exhibit efficient work dynamics by adapting the ratio of robots engaged in a given task (or not engaged in any task) based on the current demand for the task or the gain expected from task execution. Even though robots are endowed with local sensing capabilities and thus cannot measure directly the global state of the environment, local interactions between robots and the environment can be used to adapt the behavior of single robots in order to benefit the efficiency of the swarm.

11.1. Methods

Most task allocation mechanisms are based either on thresholds, where robots decide to switch activities when an observed quantity exceeds a threshold value, or on probabilistic methods, where task switching is regulated by probability values.

11.1.1. Threshold-based methods

In threshold-based methods, robots observe a given quantity in the environment and change their activity when this quantity reaches a threshold value. The observed quantity can be either local, i.e. relative to environment characteristics perceived in the neighborhood of robot locations, or global, i.e. relative to a global state of the environment which all robots are able to measure. The value of the threshold can be either fixed or variable. In case of a global quantity and a fixed threshold, having homogeneous robots (i.e. robots sharing the same control algorithm, including the same threshold value) implies that a potentially high number of robots switch activity at the same time when the threshold is crossed, which can lead to undesired oscillations in the swarm dynamics. Thus, local quantities and/or variable thresholds are generally preferred.

In task allocation methods applied to a foraging scenario, observed quantities can be related to the energy stored in the nest, so that robots begin searching for items when the energy level falls below a given threshold [78], or can be calculated based on the amount of time spent by each robot in previous searches [146], or more generally based on previous experience gained during the foraging activity [147,148]. In similar scenarios where robots can measure the success of the activity in which they are engaged, threshold values can be calculated from the success of previous activities [149]. Besides observations of the environment, in some studies [146,150] direct communication between robots has been employed to allow each robot to have a better knowledge of the environment and thus calculate more optimized threshold values.

11.1.2. Probabilistic methods

In probabilistic methods, the decision of each robot to switch activities at any given time is taken randomly, with a probability value which is usually changed dynamically based on environment observations. The random component in the control algorithm of each robot prevents a large number of robots from switching activities at the same time, even if all robots are controlled by the same probability value. Similar to threshold values in threshold-based methods, probability values can be calculated by each robot from previous experience gained during a given activity, following the principle that robots should perform an activity when the probability of success is high [151–153]. In the foraging scenario, other strategies that have been used involve sensing of a stimulus, so that an activity is executed more likely when the corresponding stimulus is higher [154], or indirect sensing of the proportion of robots engaged in a given task compared to the effort required by that task [155].

11.2. Algorithms

In most existing works, distributed task allocation techniques have been implemented in scenarios where robots are tasked with locating items scattered in the environment and operating on the items found. Often, the density of items in the environment is the main factor influencing the swarm dynamics, and robot control algorithms include mechanisms to allow each robot to estimate from local observations a global characteristic of the environment. The foraging task is an often used case study where task allocation algorithms have been applied. This section contains a review of past works dealing with distributed task allocation applied to foraging or other tasks.

11.2.1. Foraging

In the foraging scenario, task allocation has been used mainly as a mechanism to determine when each robot should engage in the foraging activity and when it should rest. Two main principles can be found in the existing literature to regulate the amount of activity executed by robots: the expected success, by which robots search for items when items are likely to be found, and the stimulus, which determines execution of the search activity when it is most needed.

In [146], *motivational behavior* is used to decide when to start and stop the search activity. Two types of motivation, called *impatience* and *acquiescence*, regulate the robot activity: impatience increases with time as long as a robot stays idle, and triggers the activation of the search activity when reaching a threshold value, while acquiescence increases with time during the search activity, and causes robots to abort their activity if not successful. Communication between robots may affect the distribution of work among swarm members by changing their motivational status: when other robots in the arena report that they are busy in the search activity, impatience of idle robots becomes lower, and acquiescence of searching robots becomes higher, so that the same activity is not executed by many robots at the same time.

Liu et al. [147] utilized a larger set of cues to modulate the robot activity. In their proposed approach, each robot keeps track of a searching time threshold (i.e. the maximum time spent searching for items before returning to the nest) and a resting time threshold (amount of time which must elapse before a robot in the nest engages in the foraging activity). These two thresholds are dynamically varied based on internal cues (item collection success rate during past search activities), environmental cues (number of collisions with other robots while searching) and social cues (results of the search activity of other robots). As pointed out by the authors, threshold variation is an adaptive mechanism by which robots change their behavior based on learned environment characteristics. To preserve scalability, communication between robots, needed for the social cues, is limited to local interactions inside the nest area, and makes use of pheromone: robots entering the nest deposit pheromone indicating whether their search activity was successful. In a subsequent work [148], the authors used a genetic algorithm to select a near-optimal set of adaptation parameters for the time thresholds of individual robots.

In [151], when a robot searching for items reaches a timeout without finding any item, it abandons the search and goes to the nest. When a robot is in the nest, previous successes and failures in item collection attempts influence the probability to leave the nest again to search for items; specifically, every time a search operation terminates (with or without success), this probability is either increased by an amount proportional to the number of consecutive successes, or decreased by an amount proportional to the number of consecutive failures. Thus, each robot learns based on its previous experience the difficulty level of finding items in the environment, and the global number of robots engaged in the foraging activity increases with the availability of items to be collected. This algorithm, tested by the authors in a scenario where items are randomly added in the arena during an experiment, shows an improved swarm efficiency compared to a simpler algorithm in which the probability of leaving the nest is fixed. Updating the probability value based on the outcome of the foraging activity also means that more successful robots are less likely to be in the resting state compared to less successful robots. This effect is analyzed in [156], where it is explained with small mechanical differences between robots, which impact foraging performance.

Campo and Dorigo [152] derived an equation for calculating the instantaneous rate of energy gained by a swarm of foraging robots, expressed in terms of robot control parameters and characteristics of the environment. This equation is embedded in the decision algorithm of each robot, which dynamically evaluates a subset of possible behaviors in terms of the expected energy gain, and adjusts its control parameters (which are essentially probabilities to execute certain actions during the foraging activity) in order to maximize the energy income. Parameter adjustment is a means of adaptation of robot behavior to the environment: as shown by the authors, robots are able to dynamically learn optimal values for their control parameters, even though in the presence of sudden changes in environment characteristics a decrease in performance is observed. The authors conjectured that this performance decrease may be due to a “memory effect”, which hinders optimal adaptation to new environment conditions.

Many task allocation methods found in animal societies are based on sensing a stimulus which quantifies the “need” for the execution of a task: larger stimulus values increase the number of robots engaging in task execution. In a foraging scenario, the stimulus is usually determined by the amount of food or energy in the nest: since the amount of food decreases with time if no robots bring new food to the nest, the corresponding increase of stimulus activates the foraging activity in resting robots, which will likely result in a future increase of food at the nest and a corresponding decrease of stimulus. Krieger and Billeter [78] controlled robots via a fixed activation threshold: when the energy falls below a given value, resting robots begin searching for items. In this case, to prevent all resting robots from starting their search activity at the same time, a heterogeneous system is used where each robot is assigned a different threshold. In [154], the mapping between stimulus and response is obtained with a probabilistic decision taken by each robot, based on a parameter, called response threshold, which regulates the “sensitivity” of robots to the stimulus; the value of this parameter is varied adaptively based on the dynamics of the amount of food in the nest.

Brutschy et al. [155] considered a foraging task divided into two sub-tasks, called *harvest* and *store*, which are sequentially interdependent, in that execution of one sub-task requires previous execution of the other. A harvesting robot transports an item from a source area to a *task interface* area, where it waits for an available robot engaged in the other sub-task, then passes the item to that robot, which will transport it to the nest. Analogously, a storing robot waits at the task interface area if there are no robots engaged in the other sub-task from which it can pick an item. Dynamic task allocation is obtained with robots measuring the waiting time and probabilistically switching from one sub-task to the other when waiting at the task interface; probability values are calculated from previous experience (i.e. the amount of time spent waiting in the past when doing each of the two sub-tasks) and from the current waiting time. In [157], task allocation is obtained in a similar scenario without encoding specific sub-tasks in robot controllers: using an automatic design method based on grammatical evolution, a swarm of robots is able to dynamically partition itself in two “specialized” groups, exploiting a characteristic of the environment that rewards division of labor.

11.2.2. Other tasks

Agassounon et al. [149] devised a task allocation method in a scenario where robots collect items randomly scattered in the environment and group them in a single cluster. Since the number of scattered items decreases as items are clustered, the probability of each robot to find items to collect (and thus the efficiency of the work done by the robot) decreases as time passes. The task allocation algorithm proposed by the authors makes each robot pause its search activity and move to a resting area if it has not found any item after a specified timeout, while resting robots resume their work after a fixed time. This mechanism increases the efficiency of the swarm by minimizing the number of robots doing useless work and at the same time ensures that the global task (grouping all items in a single cluster) is successfully executed. In [150], two variants of this algorithm are proposed: in the first variant, the timeout used to pause the search activity is not fixed but calculated dynamically based on the amount of time spent to find an item during past searches; in the second variant, robots are endowed with local direct communication capabilities and exchange their estimation of the demand for work when they encounter each other, so that the estimation of each robot can be updated based on the work done by other robots. The first variant is shown to adapt more to different sizes of the environment, while with the second variant robots are able to better handle cases where the number of items in the environment changes during an experiment.

In [153], robots are located in an arena with randomly scattered items of two types, and must locate and consume items of one of these types; to keep the number of items in the arena constant, a consumed item reappears instantly in another random location. Robots must choose an item type and search for items of the chosen type, and the objective of the swarm is to allocate robots to each type proportionally to the number of items of that type. Robots, which signal with a visual cue the task they are currently executing, are equipped with a camera vision system capable of sensing nearby items and other robots. In the distributed control algorithm devised by the authors for this scenario, robots periodically observe their neighborhood and change their current task (i.e. change the type of items to consume) probabilistically, based on a short-term memory of the number of items and robots detected for each type. Simulation experiments show that the robot swarm is able to dynamically change the proportion of robots dedicated to each task, adapting it to the proportion of items of each type in the environment.

11.3. Analysis

11.3.1. Metrics

A general objective of task allocation methods is to regulate the activities done by swarm members based on the expected utility gained from those activities. The concept of energy has often been used to describe and measure the performance of task allocation: robots are assumed to consume energy at a rate that depends on their current activity [78,147], energy income derives from successful completion of a task, and the objective of the swarm is to maximize the net energy income. Quantitative measures used in past studies to assess swarm efficiency in foraging scenarios include the ratio between the number of collected items and the total time spent searching [151], the ratio between the net energy income and the energy available from the environment [147], and the average time spent by robots to retrieve an item [147]. A measure of robustness of a swarm is proposed in [78] as the lowest energy level recorded during an experiment, while Castello et al. [154] considered the average deviation of the energy level as a performance indicator.

In scenarios where a level of completion for the collective task can be defined, the objective of the swarm is to complete the task with the minimum use of resources. As a general rule, the number of robots actively engaged in an activity at a given time can be considered as a measure of resource usage. For example, in [149] the ratio between the average cluster size (which is a measure of task completion) and the average number of active workers is used to assess the efficiency of the swarm in allocating resources. In [150], Agassounon and Martinoli defined a cost function calculated from the average cluster size, the number of clusters and the number of active workers, and the performance of the swarm during an experiment is measured by integrating the cost function over the experiment duration.

11.3.2. Models

An analytical model of distributed task allocation where each robot decides to switch from one task to another based on local observations of the environment is described in [158]. Observations, which can be limited to items in the environment or can include other robots, are treated in the model as a stochastic process, and based on their statistical properties and the robot control algorithm the dynamic evolution of the global swarm behavior can be predicted analytically. When applied to the scenario described in [153], this model is shown to predict well the swarm dynamics.

In the foraging context, extensive analytical studies on the influence of task allocation on global swarm performance have been done by Liu et al. [159,160,148]. Their model uses a probabilistic finite state machine to represent the number of robots in a given state (e.g. resting, searching for items or carrying an item toward the nest), and predicts the temporal evolution of the swarm via difference equations; the probabilities of robots to switch from one state to another are determined by the robot control parameters and the environment characteristics. Indicators of global swarm performance such as the overall energy level can then be predicted with this model by solving the difference equations.

12. Other swarm robotics tasks

This section contains a brief description of other swarm robotics tasks with some examples of relevant past works.

12.1. Odor source localization

The odor source localization problem consists of finding the source of an odor in the environment. Propagation of chemical substances in the air is subject to unpredictable dynamics due to turbulence and other wind characteristics, which cause the appearance and disappearance of the odor in various places in the environment. In [161], robots are equipped with a binary odor sensor, from which the presence of an odor can be sensed, and with an anemometer to infer the wind direction. In the basic algorithm for odor localization, a robot first moves on a spiral-like path in order to explore a given region; when it detects the odor, it moves against the wind direction until it either reaches the odor source, or does not detect the odor anymore; in the latter case, it resumes its search with the spiral-like path. Collaboration between different robots is achieved by introducing a communication mechanism by which a robot detecting the odor signals this event, and other robots in its proximity in the downwind direction move toward the first robot. Experiments show that with this mechanism a group of robots is able to reach the odor source in less time than a single robot. Values of algorithm parameters are selected off-line with a machine learning technique. In [162], robots are able to measure also the intensity of an odor, and calculate a gradient formula which includes both the odor intensity and the wind velocity. Robots are deployed in the environment forming a lattice pattern, regulated by local interaction forces, which is used as a distributed sensor network, with each robot communicating its sensor readings to its neighbors. Based on this combined sensing mechanism, each robot calculates a virtual force heading toward the estimated odor source; this force is combined with the lattice-preserving force, so that the swarm remains compact and navigates in a direction determined by the average of the forces calculated by each robot.

12.2. Object assembly

In object assembly and construction problems, a swarm is tasked with building structures from objects located in the environment. Compared to the object clustering task, construction tasks focus on specific spatial relationships between adjacent objects, so that the assembled structure has a defined shape. In nature, an interesting example of distributed construction is offered by the behavior of termites, which are able to build complex structures using local sensing and without central coordination. In [163], a wall composed of two alternating types of blocks is constructed by robots using local sensing and a minimalistic communication scheme where a robot after having deposited a block signals to other robots the color of the block with broadcast messages. In [164], robots deposit objects in spots characterized by a given luminance range; using an organizer robot that moves a beam of light according to a specific rule set, worker robots are able to build a wall with deposited objects. In [165], *active* blocks exchange information with neighboring blocks to determine whether a new block of a given type can be attached at a given site in the structure, and a robot carrying a block chooses the attachment site by “asking” (with local communication) adjacent blocks in the structure if the block being carried can be attached to them. Allwright et al. [166] simulated a stigmergic process using building blocks equipped with colored LEDs; robots communicate with each other indirectly by updating the color of the LEDs of deposited blocks, thereby giving instructions for placement of the next blocks in the structure.

12.3. Self-assembly and morphogenesis

Self-assembly in swarm robotics refers to the ability of autonomous robots to physically connect to each other utilizing only local interactions. The swarm-bots project [167] is a well known research activity that led to the creation of physical robots, called *s-bots*, which can attach to each other using a gripper and a connection ring. In [167], the self-assembly process is guided by the activation of colored LEDs that s-bots use to indicate whether other s-bots can attach to them. In [168], an evolutionary method is used to program robot behavior so that two identically programmed s-bots are able to autonomously decide which of them will activate its gripper to connect to the other robot. The self-assembly capability of s-bots has been used to accomplish various tasks, such as collaborative object transportation [138–140], hole avoidance [169] and navigation in harsh environments [170]. In the hole avoidance experiment, robots must move in the arena while avoiding holes. Robots assemble in a compact formation using their grippers; holes are detected by means of ground-facing proximity sensors, while a traction sensor detects the traction force exerted by other robots in the formation. These sensors are connected to inputs of a neural network, whose parameters are selected using an evolutionary method where the fitness function rewards robot formations able to move far from their initial positions without falling in the holes. Simulation experiments showed that with the evolved behavior robots can avoid holes by collectively changing their direction of movement, using the proximity and traction sensors to evaluate the presence of nearby holes. As described in [171], the same neural controller can also allow a robot formation to pass over small holes: by connecting with each other, robots are able to collectively estimate the size of holes and to traverse holes which cannot be traversed by a robot alone. In [170], s-bots collectively decide to self-assemble in a group when a task cannot be accomplished by a single robot. The task consists in traversing an area containing a hill; each robot estimates the difficulty of climbing the hill with its accelerometer: if the hill steepness is below a certain value, robots climb it autonomously, otherwise they self-assemble in order to be able to climb the hill.

Morphogenesis is an extension of the self-assembly concept that allows modular robotic structures to assume specific shapes: each robot joining the structure attaches to it at a position and orientation such that the desired shape is gradually built. O'Grady et al. [172] used s-bots with an LED signaling mechanism by which robots attached to a structure indicate how other robots should attach; by specifying a set of pattern extension rules, the assembled structure can be formed according to various predefined shapes. In [173], the growth of a structure is guided by radio messages sent by robots assembled in the structure, which are able to measure the relative position of neighboring robots and recruit them to dock at a desired location. In [174], robots are equipped with a rotating docking surface that can be attached to a passive surface on the front, left and right sides, and can self-assemble into linear and multiped structures. In [175], robots have a cubic shape with four docking units, one at each vertical side; the self-assembly process begins when a seed robot changes its state from *swarm mode* to *organism mode* and sends wireless messages with its infrared transmitters to recruit other robots; once a recruited robot has attached to the seed, it switches to organism mode and in turn starts recruiting other robots, until the desired structure is built; the shape of the structure is chosen by the seed robot and communicated via the infrared transmitters to recruited robots.

12.4. Coordinated motion

In order to allow a modular robotic structure to move efficiently in the environment, part or all of its constituent robots must coordinate to move on a common direction; if robots are not subject to centralized control and do not share a preferred direction of movement, coordinated motion can be achieved with a distributed approach using swarm robotics principles. For example, s-bots self-assemble using the gripper unit which is mounted on a turret that can rotate with respect to the robot chassis; in [169,171,176], to align their chassis assembled robots use their traction sensors as inputs to a neural controller which is synthesized with artificial evolution; if the fitness function used to evolve the controller rewards behaviors where a group of robots moves on a common direction, during the evolution robots "learn" how to use the input from their traction sensors to rotate their chassis.

12.5. Group size estimation

In different applications where multiple agents are grouped together, knowledge of the group size by each agent can be useful, for example to regulate the group size based on task-specific criteria. As described in [177], it is possible to obtain an accurate estimation of group size in a distributed manner, even if agents have limited communication capabilities which do not allow direct exchange of information between each member of the group. The algorithm is based on signal transmission which propagates through the group: each robot transmits a broadcast signal, which is re-transmitted by nearby robots until it propagates to the entire swarm; by calculating the ratio of self-generated transmissions to the total number of transmissions, each robot can estimate the group size.

12.6. Distributed rendezvous

The rendezvous problem for a robotic swarm scattered in the environment consists in agreeing on a single location at which all robots must converge. A usual assumption is that robots form a connected graph, i.e. any robot is in communication range of at least another robot, and following local communication links it is possible to reach any robot from any other robot. Under this assumption, distributed algorithms for the rendezvous task are described in [178]. A key characteristic of the algorithms is that they guarantee that the swarm remains a connected graph during robot movement; the authors prove this property for both a synchronous scenario, where all robots share a common clock and move in synchronized steps, and an asynchronous scenario, where individual robot movements are not dictated by a common clock. The second scenario is more aligned with the swarm robotics perspective, in that it does not require passing global information to all robots.

12.7. Collective decision-making

In many of the collective tasks described in this review, the swarm dynamics leads to a group of robots to converge to a unanimous decision; this happens when the different options available to each robot are associated to different rewards (even if robots cannot measure directly the reward associated to each option), but also in the case of options with the same utility. Examples of collective decision making can be seen in the choice of an aggregation site, a flocking direction, a path between two target locations, an object clustering site, a direction of movement in robot assemblies, etc. In [179], Montes de Oca et al. proposed a mechanism for decision making based on the majority rule opinion formation model, where robots agree on a common decision by applying a local majority rule to small groups; if a latency period is introduced after each robot takes a decision, during which the robot cannot be influenced by other robots, and if the duration of this period depends on the decision being taken (*differential latency*), then a global consensus is achieved by the swarm on the decision associated with the lowest latency. In [180], global consensus on the fastest action is achieved with robots entering an *observation state*, where they exchange their current preferred action, after each action execution. In [181], robots that must agree on a site selection transmit their preference to neighboring robots for a time duration proportional to the perceived quality of the site associated to their current preference. Yu et al. [182] used an implicit leadership algorithm to make a swarm converge to a collective decision influenced by a few informed agents; the behavior of each robot is a combination of the tendency to reach a specific goal (if the robot is informed about the goal) and the tendency to follow the average behavior of its neighbors; if the relative weight given to these two components is changed dynamically based on the local consensus observed from neighboring robots, the swarm can converge to a collective decision even in the presence of informed agents with conflicting goals.

12.8. Human-swarm interaction

Regardless of the specific task assigned to a robotic swarm, if a human operator is able to influence the swarm dynamics, for example by transferring information useful to execute the task, this capability can be used to increase the performance of task execution. According to swarm robotics principles, human-swarm interaction should not take the form of direct control exercised on the entire swarm, because this type of interaction would lack scalability; the preferred approach is to allow humans to interact *locally* with single or small groups of robots or with the environment. The technique proposed in [183] consists in taking control of one robot in the swarm, transforming it in an *avatar*, so that local modifications obtained with a changed behavior of the avatar are transferred gradually to the rest of the swarm. A similar method is used by Walker et al. [184], who investigated how a swarm can be controlled by dynamically selecting leader robots and guiding their movement. Kolling et al. [185] proposed two types of interaction, called *intermittent* and *environmental*: the first type consists in selecting individual robots to make them switch from their current behavior to a new behavior; with the second type, users manipulate a local characteristic of the environment in order to induce a new behavior in robots located in the nearby area.

13. Future research directions

As seen in the previous sections, there is an abundance of research work on many different aspects of swarm robotics systems, and swarm implementations have been proposed using a variety of design methods. To validate proposed solutions, mathematical modeling, computer simulations and experiments with real robots have been extensively used. However, to this date the use of robotic swarms in real-world applications is still lacking: while laboratory experiments can give a sense of what a given robotic system might achieve, large-scale deployments in the field would provide new insights on the different factors affecting the operation of a swarm and would stimulate further research.

An interesting topic for future studies is represented by transferring the swarm robotics discipline to the micro- and nano-scale [186]: nanorobotic systems could take different forms, for example protein- and DNA-based agents, which will respond to physical and chemical stimuli, magnetically guided systems, with robots built from ferromagnetic material that can be guided by applying magnetic fields, or bacterial systems, characterized by a powerful propulsion mechanism that can be leveraged controlling natural bacteria with external stimuli (e.g. magnetic fields) or building artificial systems replicating bacterial behavior. Possible applications of nanorobotics research include environmental sensing with nano-sensor networks, space exploration (where having small and fault-tolerant systems is of utmost importance), underground exploration of oil reservoirs, and most importantly medical applications: swarms of nanorobots can be used for precise drug delivery in the human body, maximizing therapy efficiency and minimizing negative side effects, for cell repairing, for early diagnosis of diseases or to fight tumor cells. The main challenges that researchers face for bringing nanorobotics to real implementations are related to the requirement of equipping robots with adequate power, propulsion, computation and communication capabilities.

Robots deployed in the real world will likely need to be able to handle different types of tasks in order to accomplish a given objective: depending on the scenario, the same robots will have to engage in various tasks such as aggregation, dispersion, pattern formation and collaborative object manipulation. Even though each single task may be handled relatively easily by specialized robots, combining all of them in the same robot controller would inevitably pose complexity issues that have not been fully studied by current research. Operating in diverse and changing environments, robots may need to reconfigure themselves automatically (e.g. switching from a control algorithm to another) based on the current operating conditions. This flexibility will come at the price of higher complexity in swarm design, and will also increase the risk of unpredictability of robot behavior under unexpected conditions. In light of this, the topic of human-swarm interaction could assume more importance, so that adequate mechanisms of controlling and influencing swarm operation by human supervisors are put in place.

Recent progress in technologies such as low-power wireless communication and optical sensors, especially at small scale, can be exploited in robotic systems by endowing small robots with relatively “advanced” capabilities: even though such capabilities might be seen as deviating from the minimalistic approach usually followed in swarm robotics, the definition of minimalism is subject to debate and must be put in relation with the state of the art in technology. Moreover, the presence of “advanced” capabilities such as vision may allow obtaining a desired behavior without the use of other resources, as demonstrated by recent studies where computational resources are minimized to the extreme of processing only one bit of information.

Many studies cite the lack or inadequacy of a formal mathematical analysis as a topic for future research: providing a solid theoretical foundation to results obtained empirically could help in understanding the potential capabilities of a system, as well as its inherent limitations; this, in turn, would allow deploying swarms in the real world with more confidence on what can be expected from them. Given the main inspirational source behind many swarm robotics tasks, and given the relatively simple and well understood behavior of many species of insects in nature, some researchers are envisioning the use of swarms of real insects to perform useful tasks, possibly with the help of robots that could play the role of “guides” to steer insect behavior toward the desired goal (mixed insect-robot societies). Small insects have locomotion and object manipulation properties that are unparalleled by even the most sophisticated robots with comparable size.

Despite the numerous possibilities ahead, building and deploying large swarms is still an open issue, because hardware limitations and costs need to be addressed. With the continuous technological improvements we are witnessing, envisaging real-world swarm deployments is not utopic, and in the future we will likely see applications where large numbers of robots are deployed in the environment, for example to cover large areas or perform activities that are unfeasible or dangerous for humans: de-mining, monitoring of large production or distribution plants to detect leakages or other potential hazards, environmental monitoring of underwater surfaces, cleanup of areas affected by an oil spill, etc. But currently the required technology for enabling these applications in a cost-effective way is not available yet.

14. Conclusions

In the swarm robotics discipline, a variety of tasks have been analyzed in a multitude of studies published in the last decades; some of them are directly derived from collective behaviors found in nature, for example cockroach and honeybee aggregation, bird flocking, and ant foraging, others are specific to artificial systems. All these tasks have a common point, i.e. they can be solved by a group of robots using a distributed algorithm where each robot is guided only by local interactions with the environment.

This paper presented a review of a number of past studies where swarm robotics problems are analyzed and solved with a distributed algorithm. For each problem, design methods are described, task variations are identified, past studies are grouped into task-specific categories, and an overview of analysis tools and metrics is provided. Cooperation between members of a swarm is shown to emerge from different mechanisms, ranging from stigmergy to direct communication between robots. In many instances, the global swarm dynamics emerges simply as a result of statistic properties of local interactions between robots and the environment, and the collective task executed by a swarm can be highly complex compared to the individual capabilities. As opposed to using single robots endowed with sophisticated capabilities, a large number of minimalistic robots can be deployed to produce scalable, flexible and robust systems with swarm intelligence properties.

References

[1] G. Dudek, M.R. Jenkin, E. Milios, D. Wilkes, A taxonomy for multi-agent robotics, Auton. Robots 3 (4) (1996) 375–397.
[2] E. Şahin, Swarm robotics: from sources of inspiration to domains of application, in: Swarm Robotics, Springer, Berlin, 2005, pp. 10–20.
[3] E. Ferrante, E. Dué nez-Guzmán, A.E. Turgut, T. Wenseleers, GESwarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, 2013, pp. 17–24.
[4] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, A taxonomy for swarm robots, in: Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems' 93, IROS'93, vol. 1, IEEE Press, Piscataway, 1993, pp. 441–447.
[5] Y.U. Cao, A.S. Fukunaga, A. Kahng, Cooperative mobile robotics: antecedents and directions, Auton. Robots 4 (1) (1997) 7–27.
[6] L. Iocchi, D. Nardi, M. Salerno, Reactivity and deliberation: a survey on multi-robot systems, in: Balancing reactivity and social deliberation in multi-agent systems, Springer, Berlin, 2001, pp. 9–32.
[7] L. Bayındır, E. Şahin, A review of studies in swarm robotics, Turk. J. Electr. Eng. 15 (2) (2007) 115–147.
[8] V. Gazi, B. Fidan, Coordination and control of multi-agent dynamic systems: Models and approaches, in: Swarm Robotics, Springer, 2007, pp. 71–102.
[9] I. Navarro, F. Matía, An introduction to swarm robotics, ISRN Robotics 2013, http://dx.doi.org/10.5402/2013/608164.
[10] Y. Tan, Z.Y. Zheng, Research advance in swarm robotics, Def. Technol. 9 (1) (2013) 18–39.
[11] Y. Mohan, S. Ponnambalam, An extensive review of research in swarm robotics, in: World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, IEEE, Coimbatore, 2009, pp. 140–145.
[12] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, Swarm Intell. 7 (1) (2013) 1–41.
[13] J.C. Barca, Y.A. Sekercioglu, Swarm robotics reviewed, Robotica 31 (3) (2013) 345–359.
[14] A. Mogilner, L. Edelstein-Keshet, A non-local model for a swarm, J. Math. Biol. 38 (6) (1999) 534–570.
[15] E.J. Hackett-Jones, K.A. Landman, K. Fellner, Aggregation patterns from nonlocal interactions: discrete stochastic and continuum modeling, Phys. Rev. E 85 (4) (2012) 041912.
[16] J. Vanualailai, B. Sharma, A Lagrangian-based swarming behavior in the absence of obstacles, in: Workshop on Mathematical Control Theory, Kobe University, 2010, pp. 8–10.
[17] R.C. Fetecau, Collective behavior of biological aggregations in two dimensions: a nonlocal kinetic model, Math. Models Methods Appl. Sci. 21 (7) (2011) 1539–1569.
[18] R. Fetecau, J. Meskas, A nonlocal kinetic model for predator–prey interactions, Swarm Intell. 7 (4) (2013) 279–305.
[19] A. Priolo, Swarm aggregation algorithms for multi-robot systems (Ph.D. thesis), University of Roma Tre, 2013.
[20] S. Kernbach, R. Thenius, O. Kernbach, T. Schmickl, Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system, Adapt. Behav. 17 (3) (2009) 237–259.
[21] S. Kernbach, D. Häbe, O. Kernbach, R. Thenius, G. Radspieler, T. Kimura, T. Schmickl, Adaptive collective decision-making in limited robot swarms without communication, Int. J. Robot. Res. 32 (1) (2013) 35–55.
[22] H. Hamann, H. Wörn, K. Crailsheim, T. Schmickl, Spatial macroscopic models of a bio-inspired robotic swarm algorithm, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, IEEE Press, Los Alamitos, 2008, pp. 1415–1420.
[23] T. Schmickl, H. Hamann, H. Wörn, K. Crailsheim, Two different approaches to a macroscopic model of a bio-inspired robotic swarm, Robot. Auton. Syst. 57 (9) (2009) 913–921.
[24] H. Hamann, B. Meyer, T. Schmickl, K. Crailsheim, A model of symmetry breaking in collective decision-making, in: From Animals to Animats 11, Springer, Berlin, 2010, pp. 639–648.
[25] F. Arvin, K. Samsudin, A.R. Ramli, M. Bekravi, Imitation of honeybee aggregation with collective behavior of swarm robots, Int. J. Comput. Intell. Syst. 4 (4) (2011) 739–748.
[26] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, G. Theraulaz, Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots, in: Advances in Artificial Life, Springer, Berlin, 2005, pp. 169–178.
[27] J. Amé, J. Halloy, C. Rivault, C. Detrain, J.L. Deneubourg, Collegial decision making based on social amplification leads to optimal group formation, Proc. Natl. Acad. Sci. 103 (15) (2006) 5835–5840.
[28] S. Garnier, J. Gautrais, M. Asadpour, C. Jost, G. Theraulaz, Self-organized aggregation triggers collective decision making in a group of cockroach-like robots, Adapt. Behav. 17 (2) (2009) 109–133.
[29] N. Correll, A. Martinoli, Modeling Self-Organized Aggregation in a Swarm of Miniature Robots, in: IEEE 2007 International Conference on Robotics and Automation Workshop on Collective Behaviors inspired by Biological and Biochemical Systems, 2007.
[30] G. Mermoud, J. Brugger, A. Martinoli, Towards multi-level modeling of self-assembling intelligent micro-systems, in: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, International Foundation for Autonomous Agents and Multiagent Systems, 2009, Budapest, pp. 89–96.
[31] T. Schmickl, C. Möslinger, K. Crailsheim, Collective perception in a robot swarm, in: Swarm Robotics, Springer, Berlin, 2007, pp. 144–157.
[32] T. Schmickl, C. Möslinger, K. Crailsheim, Collective perception in a robot swarm, in: Swarm Robotics, Springer, Berlin, 2007, pp. 144–157.
[33] O. Soysal, E. Şahin, Probabilistic aggregation strategies in swarm robotic systems, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, IEEE Press, Piscataway, 2005, pp. 325–332.
[34] L. Bayındır, A probabilistic geometric model of self-organized aggregation in swarm robotic systems (Ph.D. thesis), Middle East Technical University, 2012.
[35] V. Trianni, T.H. Labella, R. Groß, E. Şahin, M. Dorigo, J.L. Deneubourg, Modeling Pattern Formation in a Swarm of Self-assembling Robots, Technical Report, IRIDIA, Université Libre de Bruxelles, 2002.
[36] N. Fatès, Solving the decentralised gathering problem with a reaction–diffusion-chemotaxis scheme, Swarm Intell. 4 (2) (2010) 91–115.
[37] G. Francesca, M. Brambilla, A. Brutschy, V. Trianni, M. Birattari, AutoMoDe: a novel approach to the automatic design of control software for robot swarms, Swarm Intell. 8 (2) (2014) 89–112.
[38] M. Burger, J. Haškovec, M.T. Wolfram, Individual based and mean-field modeling of direct aggregation, Physica D: Nonlinear Phenom. 260 (2013) 145–158.
[39] G. Francesca, M. Brambilla, V. Trianni, M. Dorigo, M. Birattari, Analysing an evolved robotic behaviour using a biological model of collegial decision making, in: From Animals to Animats 12, Springer, Berlin, 2012, pp. 381–390.
[40] V. Trianni, R. Groß, T.H. Labella, E. Şahin, M. Dorigo, Evolving aggregation behaviors in a swarm of robots, in: Advances in Artificial Life, Springer, Berlin, 2003, pp. 865–874.
[41] M. Gauci, J. Chen, T. J. Dodd, R. Groß, Evolving aggregation behaviors in multi-robot systems with binary sensors, in: Distributed Autonomous Robotic Systems, Springer, Berlin, 2014, pp. 355–367.
[42] J. Gomes, P. Urbano, A.L. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intell. 7 (2–3) (2013) 115–144.
[43] J. Gomes, A.L. Christensen, Generic behaviour similarity measures for evolutionary swarm robotics, in: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation, ACM, New York, 2013, pp. 199–206.
[44] M. Gauci, J. Chen, W. Li, T.J. Dodd, R. Groß, Self-organized aggregation without computation, Int. J. Robot. Res. (2014), http://dx.doi.org/10.1177/0278364914525244.
[45] E. Bahçeci, E. Şahin, Evolving aggregation behaviors for swarm robotic systems: a systematic case study, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, IEEE Press, Piscataway, 2005, pp. 333–340.
[46] N. Fatès, N. Vlassopoulos, A robust aggregation method for quasi-blind robots in an active environment, in: ICSI 2011, 2011.
[47] A.F. Winfield, W. Liu, J. Nembrini, A. Martinoli, Modelling a wireless connected swarm of mobile robots, Swarm Intell. 2 (2–4) (2008) 241–266.
[48] F. Arvin, A.E. Turgut, F. Bazyari, K.B. Arikan, N. Bellotto, S. Yue, Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method, Adapt. Behav. 22 (3) (2014) 189–206.
[49] F. Arvin, A.E. Turgut, N. Bellotto, S. Yue, Comparison of different cue-based swarm aggregation strategies, in: Advances in Swarm Intelligence, Springer, Cham, 2014, pp. 1–8.
[50] T. Schmickl, R. Thenius, C. Möslinger, G. Radspieler, S. Kernbach, M. Szymanski, K. Crailsheim, Get in touch: cooperative decision making based on robot-to-robot collisions, Auton. Agents Multi-Agent Syst. 18 (1) (2009) 133–155.
[51] J.M. Amé, C. Rivault, J.L. Deneubourg, Cockroach aggregation based on strain odour recognition, Animal Behav. 68 (4) (2004) 793–801.

[52] D. Hu, M. Zhong, X. Zhang, Y. Yao, Self-organized aggregation based on cockroach behavior in swarm robotics, in: 2014 Sixth International Conference on Intelligent Human–Machine Systems and Cybernetics (IHMSC), vol. 1, IEEE, Hangzhou, 2014, pp. 349–354.
[53] H. Hamann, H. Wörn, A space- and time-continuous model of self-organizing robot swarms for design support, in: First International Conference on Self-Adaptive and Self-Organizing Systems, 2007. SASO'07, IEEE, Cambridge, 2007, pp. 23–23.
[54] H. Hamann, H. Wörn, A framework of space–time continuous models for algorithm design in swarm robotics, Swarm Intell. 2 (2–4) (2008) 209–239.
[55] C.W. Reynolds, Flocks, herds and schools: a distributed behavioral model, in: ACM Siggraph Computer Graphics, vol. 21(4), 1987, pp. 25–34.
[56] A.E. Turgut, H. Çelikkanat, F. Gökçe, E. Şahin, Self-organized flocking in mobile robot swarms, Swarm Intell. 2 (2–4) (2008) 97–120.
[57] H. Çelikkanat, E. Şahin, Steering self-organized robot flocks through externally guided individuals, Neural Comput. Appl. 19 (6) (2010) 849–865.
[58] E. Ferrante, A.E. Turgut, N. Mathews, M. Birattari, M. Dorigo, Flocking in stationary and non-stationary environments: a novel communication strategy for heading alignment, in: Parallel Problem Solving from Nature, PPSN XI, Springer, Berlin, 2010, pp. 331–340.
[59] E. Ferrante, A.E. Turgut, A. Stranieri, C. Pinciroli, M. Birattari, M. Dorigo, A self-adaptive communication strategy for flocking in stationary and non-stationary environments, Nat. Comput. 13 (2) (2014) 225–245.
[60] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, T. Vicsek, Flocking algorithm for autonomous flying robots, Bioinspiration Biomim. 9 (2) (2014) 025012.
[61] T. Yasuda, A. Adachi, K. Ohkura, Self-organized flocking of a mobile robot swarm by topological distance-based interactions, in: 2014 IEEE/SICE International Symposium on System Integration (SII), IEEE, Tokyo, 2014, pp. 106–111.
[62] A.T. Hayes, P. Dormiani-Tabatabaei, Self-organized flocking with agent failure: off-line optimization and demonstration with real robots, in: IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02, vol. 4, IEEE, Washington, 2002, pp. 3900–3905.
[63] G. Baldassarre, S. Nolfi, D. Parisi, Evolving mobile robots able to display collective behaviors, Artif. Life 9 (3) (2003) 255–267.
[64] G. Antonelli, F. Arrichiello, S. Chiaverini, Flocking for multi-robot systems via the null-space-based behavioral control, Swarm Intell. 4 (1) (2010) 37–56.
[65] C. Möslinger, T. Schmickl, K. Crailsheim, Emergent flocking with low-end swarm robots, in: Swarm Intelligence, Springer, 2010, pp. 424–431.
[66] E. Ferrante, A.E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, M. Dorigo, Self-organized flocking with a mobile robot swarm: a novel motion control method, Adapt. Behav. (2012) http://dx.doi.org/10.1177/1059712312462248.
[67] R.T. Vaughan, K. Støy, G.S. Sukhatme, M.J. Matarić, Whistling in the dark: cooperative trail following in uncertain localization space, in: Proceedings of the Fourth International Conference on Autonomous Agents, ACM, New York, 2000, pp. 187–194.
[68] R.T. Vaughan, K. Støy, G.S. Sukhatme, M.J. Matarić, Blazing a trail: insect-inspired resource transportation by a robot team, in: Distributed Autonomous Robotic Systems, vol. 4, Springer, Tokyo, 2000, pp. 111–120.
[69] S.A. Sadat, R.T. Vaughan, SO-LOST: an ant-trail algorithm for multi-robot navigation with active interference reduction., in: ALIFE, 2010, pp. 687–693.
[70] L. Steels, Cooperation between distributed agents through self-organisation, in: IEEE International Workshop on Intelligent Robots and Systems '90. Towards a New Frontier of Applications, Proceedings. IROS'90, IEEE, Ibaraki, 1990, pp. 8–14.
[71] H. Hamann, H. Wörn, An analytical and spatial model of foraging in a swarm of robots, in: Swarm Robotics, Springer, Berlin, 2007, pp. 43–55.
[72] J.P. Hecker, K. Letendre, K. Stolleis, D. Washington, M.E. Moses, Formica ex machina: ant swarm foraging from physical to virtual and back again, in: Swarm Intelligence, Springer, Berlin, 2012, pp. 252–259.
[73] N.R. Hoff, A. Sagoff, R.J. Wood, R. Nagpal, Two foraging algorithms for robot swarms using only local communication, in: 2010 IEEE International Conference on Robotics and Biomimetics (ROBIO), Piscataway: IEEE, 2010, pp. 123–130.
[74] S. Goss, J.L. Deneubourg, Harvesting by a group of robots, in: Proceedings of the First European Conference on Artificial Life, 1992, pp. 195–204.
[75] G. Pini, A. Brutschy, C. Pinciroli, M. Dorigo, M. Birattari, Autonomous task partitioning in robot foraging: an approach based on cost estimation, Adapt. Behav. 21 (2) (2013) 118–136.
[76] D. Goldberg, M.J. Matarić, Robust behavior-based control for distributed multi-robot collection tasks, in: Robot Teams: From Diversity to Polymorphism, 2000.
[77] M. Schneider-Fontán, M.J. Matarić, A study of territoriality: the role of critical mass in adaptive task division, in: From Animals To Animats IV, Cambridge: MIT Press, 1996, pp. 553–561.
[78] M.J. Krieger, J.B. Billeter, The call of duty: self-organised task allocation in a population of up to twelve mobile robots, Robot. Auton. Syst. 30 (1) (2000) 65–84.
[79] P.E. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, M. Gini, Communication strategies in multi-robot search and retrieval: experiences with MinDART, in: Distributed Autonomous Robotic Systems, vol. 6, Tokyo: Springer, 2007, pp. 317–326.
[80] J. Timmis, L. Murray, M. Neal, A neural-endocrine architecture for foraging in swarm robotic systems, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Berlin: Springer, 2010, pp. 319–330.
[81] A. Drogoul, J. Ferber, From Tom Thumb to the dockers: some experiments with foraging robots, in: From Animals to Animats II, 1993, pp. 451–459.
[82] E.H. Ostergaard, G.S. Sukhatme, M.J. Matarić, Emergent bucket brigading: a simple mechanism for improving performance in multi-robot constrained-space foraging tasks, in: Proceedings of the Fifth International Conference on Autonomous Agents, New York: ACM, 2001, pp. 29–30.
[83] R.C. Arkin, T. Balch, E. Nitz, Communication of behavorial state in multi-agent retrieval tasks, in: Proceedings of 1993 IEEE International Conference on Robotics and Automation, Atlanta, IEEE, 1993, pp. 588–594.
[84] N. Hoff, R. Wood, R. Nagpal, Distributed colony-level algorithm switching for robot swarm foraging, in: Distributed Autonomous Robotic Systems, Berlin: Springer, 2013, pp. 417–430.
[85] B.B. Werger, M.J. Matarić, Robotic "food" chains: externalization of state and program for minimal-agent foraging, in: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, Maes et al., The MIT Press, 1996, pp. 625–634.
[86] P. Rongier, A. Liegeois, Analysis and prediction of the behavior of one class of multiple foraging robots with the help of stochastic Petri nets, in: 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings, vol. 5, Tokyo, IEEE, 1999, pp. 143–148.
[87] K. Lerman, A. Galstyan, Mathematical model of foraging in a group of robots: effect of interference, Auton. Robots 13 (2) (2002) 127–141.
[88] V. Hartmann, Evolving agent swarms for clustering and sorting, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, New York: ACM, 2005, pp. 217–224.
[89] M. Wilson, C. Melhuish, A.B. Sendova-Franks, S. Scholes, Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies, Auton. Robots 17 (2–3) (2004) 115–136.
[90] M. Gauci, J. Chen, W. Li, T.J. Dodd, R. Groß, Clustering objects with robots that do not compute, in: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Paris, 2014, pp. 421–428.
[91] G. Vorobyev, A. Vardy, W. Banzhaf, Conformity and nonconformity in collective robotics: a case study, in: Advances in Artificial Life, ECAL, vol. 12, 2013, pp. 981–988.
[92] A. Vardy, G. Vorobyev, W. Banzhaf, Cache consensus: rapid object sorting by a robotic swarm, Swarm Intell. 8 (1) (2014) 61–87.
[93] J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chrétien, The dynamics of collective sorting: robot-like ants and ant-like robots, in: Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats, 1991, pp. 356–363.
[94] T. Wang, H. Zhang, Multi-robot collective sorting with local sensing, in: IEEE Intelligent Automation Conference (IAC), Citeseer, 2003.
[95] R. Beckers, O. Holland, J.L. Deneubourg, From local actions to global tasks: stigmergy and collective robotics, in: Artificial Life IV, vol. 181, 1994, p. 189.
[96] A. Martinoli, A.J. Ijspeert, L.M. Gambardella, A probabilistic model for understanding and comparing collective aggregation mechanisms, in: Advances in Artificial Life, Springer, Berlin, 1999, pp. 575–584.
[97] M. Maris, R. Boeckhorst, Exploiting physical constraints: heap formation through behavioral error in a group of robots, in: Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems 96, IROS 96, vol. 3, IEEE, Osaka, 1996, pp. 1655–1660.
[98] O. Holland, C. Melhuish, Stigmergy, self-organization, and sorting in collective robotics, Artif. Life 5 (2) (1999) 173–202.
[99] A. Vardy, Accelerated patch sorting by a robotic swarm, in: 2012 Ninth Conference on Computer and Robot Vision (CRV), Toronto, IEEE, 2012, pp. 314–321.
[100] G. Vorobyev, A. Vardy, W. Banzhaf, Supervised learning in robotic swarms: from training samples to emergent behavior, in: Distributed Autonomous Robotic Systems, 2014.
[101] S. Kazadi, A. Abdul-Khaliq, R. Goodman, On the convergence of puck clustering systems, Robot. Auton. Syst. 38 (2) (2002) 93–117.
[102] A. Sgorbissa, R.C. Arkin, Local navigation strategies for a team of robots, Robotica 21 (5) (2003) 461–473.
[103] F. Ducatelle, G.A. Di Caro, L.M. Gambardella, Robot navigation in a networked swarm, in: Intelligent Robotics and Applications, Berlin: Springer, 2008, pp. 275–285.
[104] F. Ducatelle, G.A. Di Caro, A. Förster, M. Bonani, M. Dorigo, S. Magnenat, F. Mondada, R. O'Grady, C. Pinciroli, P. Rétornaz, et al., Cooperative navigation in robotic swarms, Swarm Intell. (2014) 1–33.
[105] W.W. Cohen, Adaptive mapping and navigation by teams of simple robots, Robot. Auton. Syst. 18 (4) (1996) 411–434.
[106] D.W. Payton, M.J. Daily, B. Hoff, M.D. Howard, C.L. Lee, Pheromone robotics, in: Intelligent Systems and Smart Manufacturing, International Society for Optics and Photonics, Dordrecht: Kluwer Academic Publishers, 2001, pp. 67–75.
[107] A. Wurr, J. Anderson, Multi-agent trail making for stigmergic navigation, in: Advances in Artificial Intelligence, Berlin: Springer, 2004, pp. 422–428.
[108] J. Mullins, B. Meyer, A.P. Hu, Collective robot navigation using diffusion limited aggregation, in: Parallel Problem Solving from Nature-PPSN XII, Berlin: Springer, 2012, pp. 266–276.
[109] F. Ducatelle, A. Förster, G. Di Caro, L.M. Gambardella, Supporting navigation in multi-robot systems through delay tolerant network communication, in: Proceedings of the IFAC Workshop on Networked Robotics (NetRob), 2009, pp. 25–30.

[110] F. Ducatelle, G.A. Di Caro, C. Pinciroli, F. Mondada, L.M. Gambardella, Communication assisted navigation in robotic swarms: self-organization and cooperation, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, IEEE, 2011, pp. 4981–4988.

[111] T. Schmickl, K. Crailsheim, A navigation algorithm for swarm robotics inspired by slime mold aggregation, in: Swarm Robotics, Berlin: Springer, 2007, pp. 1–13.

[112] M. Szymanski, T. Breitling, J. Seyfried, H. Wörn, Distributed shortest-path finding by a micro-robot swarm, in: Ant Colony Optimization and Swarm Intelligence, Berlin: Springer, 2006, pp. 404–411.

[113] S. Nouyan, A. Campo, M. Dorigo, Path formation in a robot swarm, Swarm Intell. 2 (1) (2008) 1–23.

[114] V. Sperati, V. Trianni, S. Nolfi, Self-organised path formation in a swarm of robots, Swarm Intell. 5 (2) (2011) 97–119.

[115] M.A. Batalin, G.S. Sukhatme, Spreading out: a local approach to multi-robot coverage, in: Distributed Autonomous Robotic Systems, vol. 5, Tokyo: Springer, 2002, pp. 373–382.

[116] J. McLurkin, J. Smith, Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots, in: Distributed Autonomous Robotic Systems, vol. 6, Tokyo: Springer, 2007, pp. 399–408.

[117] E. Ugur, A.E. Turgut, E. Şahin, Dispersion of a swarm of robots based on realistic wireless intensity signals, in: 22nd International Symposium on Computer and Information Sciences, 2007. ISCIS 2007, Ankara, IEEE, 2007, pp. 1–6.

[118] E. Mathews, Self-organizing ad-hoc mobile robotic networks (Ph.D. thesis), Paderborn, Universität Paderborn, Diss., 2012.

[119] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, C. Melchiorri, Edge-weighted consensus-based formation control strategy with collision avoidance, Robotica (2013) 1–16.

[120] A. Howard, M.J. Matarić, G.S. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: Distributed Autonomous Robotic Systems, vol. 5, Tokyo: Springer, 2002, pp. 299–308.

[121] S. Poduri, G.S. Sukhatme, Constrained coverage for mobile sensor networks, in: 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04, vol. 1, New Orleans, IEEE, 2004, pp. 165–171.

[122] W.M. Spears, D.F. Spears, J.C. Hamann, R. Heil, Distributed, physics-based control of swarms of vehicles, Auton. Robots 17 (23) (2004) 137–162.

[123] R. Morlok, M. Gini, Dispersing robots in an unknown environment, in: Distributed Autonomous Robotic Systems, vol. 6, Tokyo: Springer, 2007, pp. 253–262.

[124] G. Lee, Y. Nishimura, K. Tatara, N.Y. Chong, Three dimensional deployment of robot swarms, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, IEEE, 2010, pp. 5073–5078.

[125] S.B. Mikkelsen, R. Jespersen, T.D. Ngo, Probabilistic communication based potential force for robot formations: a practical approach, in: Distributed Autonomous Robotic Systems, Berlin: Springer, 2013, pp. 243–253.

[126] G. Lee, N.Y. Chong, Self-configurable mobile robot swarms: adaptive triangular mesh generation, in: Networking Humans, Robots and Environments, 2013, pp. 59–75.

[127] G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, et al., An experiment in automatic design of robot swarms, in: Swarm Intelligence, Cham: Springer, 2014, pp. 25–37.

[128] I.A. Wagner, M. Lindenbaum, A.M. Bruckstein, Distributed covering by ant-robots using evaporating traces, IEEE Trans. Robot. Autom. 15 (5) (1999) 918–933.

[129] J. Svennebring, S. Koenig, Building terrain-covering ant robots: a feasibility study, Auton. Robots 16 (3) (2004) 313–332.

[130] E. Osherovich, V. Yanovki, I.A. Wagner, A.M. Bruckstein, Robust and efficient covering of unknown continuous domains with simple, ant-like a(ge)nts, Int. J. Robot. Res. 27 (7) (2008) 815–831.

[131] T. Kuyucu, I. Tanev, K. Shimohara, Evolutionary optimization of pheromone-based stigmergic communication, in: Applications of Evolutionary Computation, Berlin: Springer, 2012, pp. 63–72.

[132] B. Ranjbar-Sahraei, G. Weiss, A. Nakisaee, A multi-robot coverage approach based on stigmergic communication, in: Multiagent System Technologies, Berlin: Springer, 2012, pp. 126–138.

[133] J.H. Reif, H. Wang, Social potential fields: a distributed behavioral control for autonomous robots, Robot. Auton. Syst. 27 (3) (1999) 171–194.

[134] C.R. Kube, H. Zhang, Collective robotic intelligence, in: Second International Conference on Simulation of Adaptive Behavior, 1992, pp. 460–468.

[135] C.R. Kube, E. Bonabeau, Cooperative transport by ants and robots, Robot. Auton. Syst. 30 (1) (2000) 85–101.

[136] J. Chen, M. Gauci, R. Groß, A strategy for transporting tall objects with a swarm of miniature mobile robots, in: 2013 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Karlsruhe, 2013, pp. 863–869.

[137] R. Fujisawa, H. Imamura, F. Matsuno, Cooperative transportation by swarm robots using pheromone communication, in: Distributed Autonomous Robotic Systems, Springer, Berlin, 2013, pp. 559–570.

[138] G.C. Pettinaro, L.M. Gambardella, A. Ramirez-Serrano, Adaptive distributed fetching and retrieval of goods by a swarm-bot, in: 12th International Conference on Advanced Robotics, 2005. ICAR'05. Proceedings, IEEE, Seattle, 2005, pp. 825–832.

[139] R. Groß, M. Dorigo, Cooperative transport of objects of different shapes and sizes, in: Ant Colony Optimization and Swarm Intelligence, Springer, Berlin, 2004, pp. 106–117.

[140] R. Groß, M. Dorigo, Towards group transport by swarms of robots, Int. J. Bio-Inspired Comput. 1 (1) (2009) 1–13.

[141] A.J. Ijspeert, A. Martinoli, A. Billard, L.M. Gambardella, Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment, Auton. Robots 11 (2) (2001) 149–171.

[142] L. Li, A. Martinoli, Y.S. Abu-Mostafa, Learning and measuring specialization in collaborative swarm systems, Adapt. Behav. 12 (3–4) (2004) 199–212.

[143] A. Martinoli, F. Mondada, Collective and Cooperative Group Behaviours: Biologically Inspired Experiments in Robotics, Springer, Berlin, 1997.

[144] A. Martinoli, K. Easton, W. Agassounon, Modeling swarm robotic systems: a case study in collaborative distributed manipulation, Int. J. Robot. Res. 23 (4–5) (2004) 415–436.

[145] K. Lerman, A. Galstyan, A. Martinoli, A.J. Ijspeert, A macroscopic analytical model of collaboration in distributed robotic systems, Artif. Life 7 (4) (2001) 375–393.

[146] L.E. Parker, ALLIANCE: an architecture for fault tolerant multirobot cooperation, IEEE Trans. Robot. Autom. 14 (2) (1998) 220–240.

[147] W. Liu, A.F. Winfield, J. Sa, J. Chen, L. Dou, Towards energy optimization: emergent task allocation in a swarm of foraging robots, Adapt. Behav. 15 (3) (2007) 289–305.

[148] W. Liu, A.F. Winfield, Modelling and optimisation of adaptive foraging in swarm robotic systems, Int. J. Robot. Res. 2010, http://dx.doi.org/10.1177/0278364910375139.

[149] W. Agassounon, A. Martinoli, R. Goodman, A scalable, distributed algorithm for allocating workers in embedded systems, in: 2001 IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, IEEE, Tucson, 2001, pp. 3367–3373.

[150] W. Agassounon, A. Martinoli, Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems, in: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, ACM, New York, 2002, pp. 1090–1097.

[151] T.H. Labella, M. Dorigo, J.L. Deneubourg, Division of labor in a group of robots inspired by ants' foraging behavior, ACM Trans. Auton. Adapt. Syst. 1 (1) (2006) 4–25.

[152] A. Campo, M. Dorigo, Efficient multi-foraging in swarm robotics, in: Advances in Artificial Life, Springer, New York, 2007, pp. 696–705.

[153] C. Jones, M.J. Matarić, Adaptive division of labor in large-scale minimalist multi-robot systems, in: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings, vol. 2, IEEE, 2003, pp. 1969–1974.

[154] E. Castello, T. Yamamoto, Y. Nakamura, H. Ishiguro, Task allocation for a robotic swarm based on an adaptive response threshold model, in: 2013 13th International Conference on Control, Automation and Systems (ICCAS), IEEE, Gwangju, 2013, pp. 259–266.

[155] A. Brutschy, G. Pini, C. Pinciroli, M. Birattari, M. Dorigo, Self-organized task allocation to sequentially interdependent tasks in swarm robotics, Auton. Agents Multi-agent Syst. 28 (1) (2014) 101–125.

[156] T.H. Labella, M. Dorigo, J.L. Deneubourg, Self-organised task allocation in a group of robots, in: Distributed Autonomous Robotic Systems, vol. 6, Springer, Tokyo, 2007, pp. 389–398.

[157] E. Ferrante, A.E. Turgut, E. Duenez-Guzman, M. Dorigo, T. Wenseleers, Evolution of self-organized task specialization in robot swarms, PLoS Comput. Biol. 11 (8) (2015), http://dx.doi.org/10.1371/journal.pcbi.1004273.

[158] K. Lerman, C. Jones, A. Galstyan, M.J. Matarić, Analysis of dynamic task allocation in multi-robot systems, Int. J. Robot. Res. 25 (3) (2006) 225–241.

[159] W. Liu, A.F. Winfield, J. Sa, Modelling swarm robotic systems: a case study in collective foraging, in: Towards Autonomous Robotic Systems (TAROS 07), 2007, pp. 25–32.

[160] W. Liu, A.F. Winfield, A macroscopic probabilistic model of adaptive foraging in swarm robotics systems, http://dx.doi.org/10.1.1.150.1463.

[161] A.T. Hayes, A. Martinoli, R.M. Goodman, Swarm robotic odor localization: off-line optimization and validation with real robots, Robotica 21 (04) (2003) 427–441.

[162] D. Zarzhitsky, D.F. Spears, W.M. Spears, Swarms for chemical plume tracing, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, IEEE, Pasadena, 2005, pp. 249–256.

[163] J. Wawerla, G.S. Sukhatme, M.J. Matarić, Collective construction with multiple robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, vol. 3, IEEE, Lausanne, 2002, pp. 2696–2701.

[164] R.L. Stewart, R.A. Russell, A distributed feedback mechanism to regulate wall construction by a robotic swarm, Adapt. Behav. 14 (1) (2006) 21–51.

[165] J. Werfel, Y. Bar-Yam, R. Nagpal, Building patterned structures with robot swarms, in: International Joint Conference on Artificial Intelligence, vol. 19, Morgan Kaufmann Publishers, San Francisco, 2005, pp. 1495–1502.

[166] M. Allwright, N. Bhalla, H. El-Faham, A. Antoun, C. Pinciroli, M. Dorigo, SRoCS: leveraging stigmergy on a multi-robot construction platform for unknown environments, in: Swarm Intelligence, Springer, Cham, 2014, pp. 158–169.

[167] R. Groß, M. Bonani, F. Mondada, M. Dorigo, Autonomous self-assembly in swarm-bots, IEEE Trans. Robot. 22 (6) (2006) 1115–1130.

[168] E. Tuci, C. Ampatzis, V. Trianni, A.L. Christensen, M. Dorigo, Self-assembly in physical autonomous robots—the evolutionary robotics approach, in: ALIFE, 2008, pp. 616–623.

[169] V. Trianni, S. Nolfi, M. Dorigo, Cooperative hole avoidance in a swarm-bot, Robot. Auton. Syst. 54 (2) (2006) 97–103.
[170] R. O'Grady, R. Groß, A.L. Christensen, M. Dorigo, Self-assembly strategies in a group of autonomous mobile robots, Auton. Robots 28 (4) (2010) 439–455.
[171] V. Trianni, M. Dorigo, Emergent collective decisions in a swarm of robots, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, IEEE, Pasadena, 2005, pp. 241–248.
[172] R. O'Grady, A.L. Christensen, M. Dorigo, SWARMORPH: multirobot morphogenesis using directional self-assembly, IEEE Trans. Robot. 25 (3) (2009) 738–743.
[173] N. Mathews, A.L. Christensen, R. O'Grady, P. Rétornaz, M. Bonani, F. Mondada, M. Dorigo, Enhanced directional self-assembly based on active recruitment and guidance, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, San Francisco, 2011, pp. 4762–4769.
[174] H. Wei, Y. Cai, H. Li, D. Li, T. Wang, Sambot: A self-assembly modular robot for swarm robot, in: 2010 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Anchorage, 2010, pp. 66–71.
[175] W. Liu, A.F. Winfield, Autonomous morphogenesis in self-assembling robots using IR-based sensing and local communications, in: Swarm Intelligence, Springer, Berlin, 2010, pp. 107–118.
[176] G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, S. Nolfi, Self-organized coordinated motion in groups of physically connected robots, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 37 (1) (2007) 224–239.
[177] M. Brambilla, C. Pinciroli, M. Birattari, M. Dorigo, A reliable distributed algorithm for group size estimation with minimal communication requirements, in: International Conference on Advanced Robotics, 2009. ICAR 2009, IEEE, Munich, 2009, pp. 1–6.
[178] J. Lin, A.S. Morse, B.D. Anderson, The multi-agent rendezvous problem, in: 42nd IEEE Conference on Decision and Control, 2003. Proceedings, vol. 2, IEEE, Munich, 2003, pp. 1508–1513.
[179] M.A. Montes de Oca, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, M. Dorigo, Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making, Swarm Intell. 5 (3–4) (2011) 305–327.
[180] A. Scheidler, A. Brutschy, E. Ferrante, M. Dorigo, The k-unanimity rule for self-organized decision making in swarms of robots, Int. J. Robot. Res. 2015, http://dx.doi.org/10.1109/TCYB.2015.2429118.
[181] G. Valentini, H. Hamann, M. Dorigo, Efficient decision-making in a self-organizing swarm of simple robots: on the speed versus accuracy trade-off, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, 2015, pp. 1305–1314.
[182] C.H. Yu, J. Werfel, R. Nagpal, Collective decision-making in multi-agent systems by implicit leadership, in: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 3, International Foundation for Autonomous Agents and Multiagent Systems, Toronto, 2010, pp. 1189–1196.
[183] S. Bashyal, G.K. Venayagamoorthy, Human swarm interaction for radiation source search and localization, in: IEEE Swarm Intelligence Symposium, 2008. SIS 2008, IEEE, Saint Louis, 2008, pp. 1–8.
[184] P. Walker, S. Amirpour Amraii, N. Chakraborty, M. Lewis, K. Sycara, Human control of robot swarms with dynamic leaders, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), IEEE, Chicago, 2014, pp. 1108–1113.
[185] A. Kolling, K. Sycara, S. Nunnally, M. Lewis, Human swarm interaction: an experimental study of two types of interaction with foraging swarms, J. Hum. Robot Interact. 2 (2) (2013) 103–128.
[186] C. Mavroidis, A. Ferreira, Nanorobotics: Current Approaches and Techniques, Springer Science & Business Media, New York, 2013.

Levent Bayındır received his B.Sc. degree in Computer Science from Ege University, Turkey, in 2002 and his Ph. D. degree in Swarm Robotics at the KOVAN research lab from Middle East Technical University, Turkey, in 2012. He is currently an assistant professor of Computer Engineering at Atatürk University in Erzurum, Turkey. His research interests include swarm robotics and pervasive computing.