Swarm and Evolutionary Computation 50 (2019) 100565

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Review of methodologies and tasks in swarm robotics towards standardization

Nadia Nedjah [a,*], Luneque Silva Junior [b]

[a] Department of Electronics Engineering and Telecommunication, Faculty of Engineering, State University of Rio de Janeiro, Brazil
[b] Post Graduate Program on System Engineering and Computation, Federal University of Rio de Janeiro, Brazil

ARTICLE INFO

Keywords:
Swarm robotics
Multi-robot systems
Modeling
Cooperative tasks

ABSTRACT

Swarm Robotics (SR) is an extension of the study of Multi-Robot Systems that exploits concepts of communication, coordination and collaboration among a large number of robots. The massive parallelization yielded by the robots working together can make a task faster than in the case of the usage of a single complex robot. One of the main aspects in robotic swarms is that the control is decentralized by definition and distributed among the robots of the swarm, improving the system robustness and fault-tolerance. Furthermore, this characteristic often allows the emergence of collective behaviors from the robot's interaction with each other and with the environment through their embodied sensors and actuators. In most cases, the number of inputs from sensor readings turns analytical solutions hard or even impossible. Thus, many *ad-hoc* approaches are contributed to deal with the situation at hand. The main goal of this review is to find out, through the study of existing research works of the field, the reason behind the lack of exploitation of swarm robotic systems in real-world applications. For this purpose, we first review the different possibilities of study in SR: physical and simulated robotic platforms, development methodologies and the variety of basic tasks and collective behaviors. We then briefly describe some fields related do SR that have a big impact on the development of SR. After that, based on existing taxonomies found in literature, we categorize existing research works regarding SR in two large main groups: those that deal with SR design and those that deal with tasks as required in SR. The review of both existing robots and techniques in the literature show a diversity of approaches to discuss SR issues. Nonetheless, it is easily noticeable from these works that there is a clamant absence of solid real-world applications of SR. An analysis of the interests and bottlenecks of this field indicates that the number of research works is smaller than those in other related areas. This suggests that, even though with many research studies, the field of SR is not yet mature enough, mainly due the absence of a universal methodology and generic robots that can be used in any, or at least in many, applications. Thus, we emphasize, discuss and analyze the urgent need for standardization of many aspects in SR, including hardware and software, as to allow a possible flourishing of SR applicability to real-world applications. This standardization could accelerate a great deal the field of SR, thus facilitating the development of SR solutions for applications that impact our daily life.

1. Introduction

The *Multi-Robot Systems* (MRS) arose as an extension of the *mobile robots* research. The main idea behind MRS is that many robots can cooperate to complete a given task faster than a single mobile robot [1], actuating in different places at the same time. Such systems not only exploit an intrinsic parallelism, but they also can be fault tolerant due to the redundancy of the robots.

The term *swarm* was introduced by Beni [2] in his study of *Cellular Robotic Systems*. A robotic swarm must have specific characteristics, similar to those found in societies of insects, such as decentralized control, asynchronism, simple and quasi-identical members [3]. The concept of swarm as described by Beni [3] is robust enough to explain many behaviors of insect societies in the field of biology [4,5]. The general concept of swarms also led to the emergence of the *swarm optimization* meta-heuristics, such as the *Ant Colony Optimization* [6] and the *Particle Swarm Optimization* [7]. A well accepted definition of *Swarm Robotics* is given by Şahin [8]:

* Corresponding author.
E-mail address: nadia@eng.uerj.br (N. Nedjah).

https://doi.org/10.1016/j.swevo.2019.100565
Received 18 October 2017; Received in revised form 10 July 2019; Accepted 10 August 2019
Available online 15 August 2019

> "SWARM robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment."

Besides cooperation, three desired characteristics of SR are *robustness*, *flexibility* and *scalability*. A swarm system is said to be *robust* if it can continue to run properly even if one or more its members stop functioning due to internal failures, environmental changes, or even both. This is achieved by the redundancy of the individuals and the absence of leaders. A swarm system is said to *scalable* if it can work with different number of its members. So, an individual can work in the same mode within a swarm of either a large or small number of members without a significant impact on its performance. A swarm system is said to *flexible* when changes in terms of swarm behavior are possible. In a flexible system, the internal parameters of individuals can be adjusted with the occurrence of changes in the environment. All these three concepts are related to a single main idea, which dictates that a swarm system must be able to perform the proposed task under different conditions. Although these features are found in natural swarm systems, it is hard to be replicated, and may be even impossible in several platforms due to hardware limitations. For instance, Winfield and his colleagues [9,10] argued the possibility of SR achieving these three characteristics, which is taken as true by default in literature. The authors conclude that robustness and scalability can be improved by the study of fault tolerance in SR. The fault tolerance was also discussed in the work of Christensen et al. [11], where some capabilities needed to swarm face hardware faults are considered.

Furthermore, five more features are also recommended in SR, as described in the work of Şahin [8]. These are as follows:

1. A swarm is composed of *autonomous robots* that can physically move in an environment and interact with other physical objects.
2. The control, which is distributed in most applications, must support the existence of a *large number* of robots.
3. The swarm is either *homogeneous* or *heterogeneous*, that is, being composed by a number of few homogeneous groups.
4. Individual robots must cooperate to carry out a given task.
5. The sensing and communication capabilities are local and, thus the robots have no access to global information.

The main goal of SR, as well as any other robotic system, is to perform tasks to help humans. Robots may move in hazardous places, use different kind of sensors to monitor the environment, work a long period of time, or simply be more precise than humans due to their enhanced computational capabilities. It is safe to expect that in the near future, robots would be part of our daily life. For instance, nowadays, almost every home has a vacuum cleaner robot [12]. Two impressive examples of SR research are the *thousand-robot* swarm presented by Rubenstein et al. [13] and the construction team of *termite-inspired* robots designed by Werfel et al. [14]. In both works, there is a group of robots performing assembly tasks. In the first work, the robots spatially organize themselves to display, at swarm level, shapes predefined by the user. In the second work, the robots build predetermined complex structures using simple blocks. These two works result in the swarm performing a useful task in response to user input. However, these studies show only a narrow aspect of SR related to robotic control, and they do not address some main concerns about other challenges in the field, such as the robustness in face of dynamic tasks or adaption to changing environments.

The design of an entire robotic swarm is not an easy task. It can be divided in several levels, ranging from the design for physical implementation of individual robots, to the abstract and mathematical description of collective behaviors. It is noteworthy to point out that generally speaking, the material used to manufacture the swarm robot body is not an issue. Standard materials are mostly used unless it is a very special design for a particular objective [15]. Designing distributed SR controllers is an important role, setting how each robot should react to the environment to execute a given task. However, the implementation of such controllers in most cases is a complex process. Each robot must process the readings from many sensors, including readings with high noise, receive messages from neighboring robots, and then decide how to move or which messages must be sent. Soft computing approaches [16] are being used in several works to facilitate this design, achieve acceptable results when the analytical modeling is hard or impossible. An example is the use of *Evolutionary Computation* to iteratively adapt the settings of robot controller, usually neural network based [17]. Some approaches also rely on the swarm architecture to use other nature-inspired meta-heuristics, for instance, the robot clustering based on honeybees behavior [18] and the task partitioning based on particle swarm optimization [19]. There are also standard techniques for mobile robotics, such as behavioral robotics [20], the layered subsumption architecture [21,22] and motor schema [23], that are always at the basis of any design of an effective swarm robotic system.

The aim of this work is to review and expose the diversity of approaches on SR studies. At the same time, this work attempts to uncover the *ad hoc* nature of the existing works, and show that there is an absence of consensus at many levels. It is important to emphasize that this kind of analysis has never been done in previous reviews. A significant issue is the lack of common generic hardware platforms and controller design methodologies. Nonetheless, some effort are being expended to fill this gap [24]. In many cases, current solutions are problem-specific and not enough to solve a variety of required tasks. So, we attempt to encourage and emphasize the urgent need for standardization of many aspects in SR. This includes the robot hardware and control software. This could be the remedy to the flourishing of SR applicability to solve real-world applications effectively, efficiently but at the expense of a very low-cost. As far as we are concerned, this standardization could be one of the key points to accelerate considerably the field of SR, and help in the development of reasonable SR solutions for applications that impact our daily reality. It is noteworthy to point out that each section in the remaining of this paper includes several references to a variety SR research works, including the most recent or the most mentioned in the literature.

This rest of this paper is divided into seven sections. First, in Section 2, we describe some fields of study that share characteristics with SR. Then, in Section 3, we present the platforms used in SR research works. After that, in Section 4, we introduce and comment on some existing classifications, which were found in earlier SR reviews. Following these existing taxonomies of SR, relevant research works are classified in two main categories: the methodologies, which are addressed in Section 5, and the tasks, which are presented in Section 6. Subsequently, in Section 7 we present the promising research directions in SR. Later, in Section 8, we analyze the interests on SR in the recent years. Finally, in Section 9, we draw some conclusions of the proposed review.

2. Related work

Swarm robotics is a comprehensive field of study. In fact, we use the knowledge gained from several independent areas from hardware and software point of view. Some of these related areas that share characteristics with SR are presented in the rest of this section.

In recent years, several reviews on swarm robotics reviews have been published [24–29]. Each of these reviews addresses a particular aspect of swarm robotics. Some of these works provide taxonomies that are useful to categorize related works. Perhaps that Dorigo's et al. review [25] is the main door to enter swarm robotics research field for a new comer. It defines all aspect related to swarm robotics, starting from its origin, intrinsic feature and desirable properties, going through its potential applications and scientific implications, listing some of the main related ongoing research works and reporting on many aspects of design, analysis and emerging behaviors, till presenting some open issues about the use of swarm robotics in real-world applications.

One of the challenges facing the flourishing of swarm robotics is its applicability in the engineering field. Brambilla et al. [24] analyze the literature from the point of view of swarm engineering. They survey mainly the ideas, definitions and concepts that contribute to the usage of swarm robotics in the engineering field and thus could be useful to approach real-world applications. The authors also contribute two very exhaustive yet concise taxonomies: one for methods and the second for tasks that are applied in swarm robotics. Also, Garattoni and Birattari [26] survey existing research works on swarm robotics aiming at the definition of an engineering methodology for designing, analyzing and maintaining robotic swarms. They do so by presenting swarm robotics from an engineering perspective, and describing related works that contribute to the advancement of swarm robotics as an engineering field.

Another difficult challenge in swarm robotics is about system formal modeling, as to be able to effectively yet efficiently simulate robotic swarms. Hamann and Schmickl [27] as well as Hamann in Ref. [30] contributed interesting works, wherein they promote the existing research efforts on swarm systems and on the motivation of formal mathematical modeling of these kind of distributed and self-organizing systems. A special emphasis is expended to demonstrate how mathematical models of different kinds may increase our understanding of the self-regulation present within natural swarm systems such as social insect colonies. Also, they illustrate how mathematical models are used to handle and optimize the behavior of artificial swarm systems as used in swarm robotics. Common problems, such as modeling effort and the model itself, impact as generators of novel empirical experiments and that of empirical experimentation that yields meaningful model parameterizations. Another noticeable formal work on robotic swarm modeling has been contributed by Correll and Hamann [29]. They provide on widespread overview on probabilistic modeling of many aspects in swarming systems, such as population dynamics, collaboration and spatial distribution of the swarm members as well as for collective decision and optimization within the swarm. The work is concluded by launching some open challenges about combining non-spatial with spatial probabilistic modeling techniques to yield better fitted models for swarm robotics.

One of the important constituent parts of a swarming system in general and swarm robotics in particular is the exhibited collective behavior. Trianni and Campo [28] discuss in a didactic way some basic yet fundamental collective behaviors observed in the field of swarm robotics. Among others, it defines and relates research work on variants of the behavior of swarm aggregation, swarm member synchronization coordinated motion within a swarm, exploration and decision-making as performed by the swarm.

Our main purpose in this review consists of studying research works of the field of swarm robotics in an attempt to find out why until nowadays SR-based systems are still not yet part of our daily life via real-world applications. Thus, we would like through this review to informally show and emphasize that there is an urgent need for standardization of many aspects in SR, including hardware and software subsystems. Standardization could be valid solution to remedy to this shortage in SR applicability to real-world applications. This It could accelerate a great deal the field of SR, by eliminating the need to a new robot project with each new task form one side and new *ad-hoc* methodology to control the swarm collective behavior, thus facilitating the development of SR solutions for real-world applications. Note that standardization could also be a general problem in robotics, not only in swarm robotics. However, in the former case, it is mitigated as an application in this case usually requires a single industrial robot.

2.1. Networked devices

There are three ways, as described in the literature [31], that robots can use to communicate with each other in order to achieve some collective behavior. First, in *interaction* via *environment*, the robots can share information by changing characteristics of the environment, such as putting and reading marks on the ground. This way of communicating is referred to as *stigmergy*. Second, *interaction* via *sensing* is achieved by the use of robot's sensors and actuators, wherein the robots should be able to notice the presence of other robots, obstacles, or both. Third, in *interaction* via *communication*, there is the direct use of wireless messages, sent and received by neighboring robots. A swarm of robots using a communication strategy based on messages can be seen as a network, comprising processing elements able to communicate and exchange information. It is noteworthy to point out that when robots communicate with the neighboring robots, this is often referred to as local communication, while when a robot communicate with all the robots of the swarm, this is referred to as global communication or broadcasting.

Different communication standards are used to implement the wireless network infrastructure used in SR. These technologies are used to interconnect devices for different applications. For example, a group of networked devices can be designed to form a *mobile ad hoc network* (MANET) [32]. This kind of wireless network is characterized as autonomously self-organized networks without infrastructure support. In such systems, nodes move arbitrarily, and thus the network may experience rapid and unpredictable topology changes. This concept, well suitable for SR, is used in many systems of connected devices, such as *vehicular ad hoc networks* (VANETs) [33], *Smartphone ad hoc networks* (SPANs) [34], *Internet-based mobile ad hoc networks* (iMANETs) [35] and Military Tactical Networks [36]. Another system composed by interconnected devices are the *Wireless sensor networks* (WSNs) [37]. These systems are usually composed by low power devices, called *sensor nodes*, that can measure and gather information about the environment state (mechanical, thermal, chemical, optical, magnetic, and so forth) [38,39]. The nodes are usually equipped with one or more sensors, but have limited processing resources and low communication range. Potential applications of WSNs include the military tracking and surveillance [40], natural disaster monitoring [41] and hazardous environment exploration [42]. A detailed discussion on this issue can be found on [43]. Furthermore, robots in SR should be able to easily be added and/or removed, automatically, to provide for a robust operating environment. Hence, robot networks are becoming complex and dynamic [30,44]. When a robot moves, its neighborhood changes and its relationship to the environment also changes. As a consequence, the information it acquires and the actions it executes must change too. Not only is the network topology dynamic, but also the robot's behavior adapts to the actual topology changes. It is difficult to predict the performance of such dynamic robot networks [44]. Yet, it is this analysis problem, designers of robot networks must solve before deploying the robot network.

Many systems of networked devices have differences and similarities with the interconnected robots in SR. For instance, even though the nodes usually have mobility (as in the case of VANETs), they are unable to modify the environment around them due to lack of actuators, as it is the case in swarm robotics. Nevertheless, the similarities between the technologies, which are all composed by distributed devices using some communication architecture, allow methods developed for other wireless network based systems to be used in SR, such as communication protocols [45,46] and localization strategies [47,48].

2.2. Nature-inspired meta-heuristics

Although *swarm intelligence* has been originally described by Beni [3] as a feature of embodied systems, this concept was the first step towards the emergence of various successful methods. Nature-inspired algorithms rely on the use of multi-agent methodologies to propose numerical solutions of complex mathematical problems. The interaction of these agents follows rules that are based on communication protocols as it occurs in animals groups and insect societies [49]. In general, the operation of swarm intelligence based algorithms has many similarities to SR behaviors, mainly because both take inspiration from natural swarm systems. These techniques are also used in the design of robots and controllers. Some widespread meta-heuristics are briefly described in the following.

Evolutionary Computation [50,51] is a field of study characterized by algorithms that are inspired by biological evolution mechanisms, such as reproduction, mutation, recombination and selection. In general terms, evolutionary computation deals with the evolution of successive generations of a population of individuals. Each individual is a potential solution to a given problem. The quality of each individual determines whether it should reproduce, thus propagating their genes to the next generation of individuals. This process, often used to describe the biological evolution, is the ground of the *Evolutionary Algorithms*. A successful example is the *Genetic Algorithm* (GA), proposed by Holland [51,52]. In GA, the population is composed by candidate solutions of the problem being solved. Each solution has a set of properties organized in a binary encoding, called chromosomes or genotype. The fitness or quality of the solutions is defined by an objective function, which is problem-dependent. *Evolutionary Robotics* (ER) makes use of the concepts of Evolutionary Computation to design robots and/or robotic systems [17]. ER will be further detailed later in Section 5.3 as a design methodology.

Particle Swarm Optimization (PSO) is a meta-heuristic inspired by the coordinated flight of birds [7,51]. It aims at finding the best value of an objective function using multiple particles. Each particle moves within the search space. Each of its positions, which is potential solution of the optimization problem, is associated with a fitness value of the objective function. The particles iteratively update their speed and direction using the so far locally found best solution.

Another widely used meta-heuristic is *Ant Colony Optimization* (ACO) [6,51]. This method is based on the search of paths made by ants. Each agent, known as *artificial ant* travels the search space associated with the problem whose solution is to be found. So, in each cycle, artificial ants find paths, which are potential solutions of the problem, and use the knowledge to direct their future iterations as well as the solutions of other ants. Artificial ants use an artificial pheromone that is deposited along visited paths in the search space. The greater is the concentration of pheromone on a given path the better is the quality of the corresponding solution.

Motivated by the promising results of PSO and ACO, many other nature-inspired algorithms were idealized. Clerc proposed a parameter-free PSO, called *Tribes* [53]. Karaboga developed the *Artificial Bee Colony* (ABC) [54] for optimizing multi-variable and multi-modal numerical functions. *Firefly Algorithm* [55] is inspired by the social behavior of fireflies and the phenomenon of bioluminescent communication. *Cuckoo Search* [56] is another technique, based on the breeding behavior of certain species of cuckoos, such as the brood parasitism. *Backtracking Search Optimization Algorithm* (BSA) [57] is hybrid evolutionary algorithm that exploits some swarming principles for solving real-valued numerical optimization problems.

The computational techniques discussed in this section have common characteristics, such as *(i)* the use of multiple agents (distributed processing), which rate the quality of certain magnitudes, measured or calculated; and *(ii)* the comparison of such magnitudes obtained by different agents, using local or global communication (message passing, shared memory). These basic concepts allow the design of different techniques, and are exactly what gives the *swarm* characteristic of such algorithms and makes them suitable to solve specific swarm robotics related problems. The same concepts are inherited by MRS, going from multiple agents (software) to multiple devices (hardware).

2.3. Applicability of swarm and evolutionary meta-heuristics in swarm robotics

In the work of El Zoghby et al. [58], the main research directions in Swarm Intelligence implementation within a robot network through the cooperation among the robots are outlined. The authors show a clear illustration and characterization of the relationship between swarm intelligence and evolutionary computation with the cooperation among the robots forming a swarm. Actually, it is claimed that the convergence of robot cooperation and swarm intelligence has lead us towards the emergence of the field of Swarm Robotics. In the sense, Swarm Robotics is put at the intersection of bio-inspired systems, swarm intelligence and self-organized and distributed systems. From a historical point of view, the first experiments on multi-robot systems, which could be identified somehow as swarm robotic system, were conducted in late 1940s by Walter and his team [59]. They show some techniques that are reflected in today's reactive and bio-inspired robots. Swarm Robotics becomes an active field of research only in the 1990's after Beni's [2] introduction of the concept of swarm robotics by discussing cellular robotics systems. Later, Deneubourg et al. [60] introduced the concept of stigmergy in robots that mimic the ant's behavior. Since then, many research have been developed collective and self-organized systems and working on collective behaviors in swarm robotics inspired by the social organization in insects. Subsequently, it became notorious that the swarm and evolutionary meta-heuristics are well-suited for the design of SRs. Nonetheless, it is worth mentioning that these meta-heuristics usually come in different versions, wherein the communication between individuals is either global *vs.* local, such as the Global-Best PSO *vs.* Local-Best PSO, depending on the topology used and the execution is either synchronous *vs.* asynchronous [61–63]. Of course, for SR applications, one needs to use the most appropriate versions of these techniques, *i.e.* those that are based on local communication and asynchronous execution. There are many research works that solve swarm robotics related problems using these meta-heuristics. In the following, we introduce most relevant such works, wherein swarm and evolutionary meta-heuristics are applied to give rise to effective and efficient solutions to swarm robotics related problems.

In the work of Rezk et al. [58], obstacle avoidance, which is an extremely important task in swarm robotics is solved using a genetic algorithm. It teaches robots how to avoid obstacles in different environments. Also, i the work of Nitschke et al. [64], three cooperative co-evolution methods for automated controller design in simulated robot teams are validated. A collective Neuro-Evolution approach co-evolves multiple robot controllers using emergent behavioral specialization in order to increase collective behavior task performance.

In the work of Wang et al. [65], effective environment exploration of unknown environment, which is a precondition of environment mapping and carrying out other tasks for multi-robot system, is performed using variation of PSO. The authors proposed the robotic PSO (RDPSO) and the fractional order RDPSO (FORDPSO) to increase the algorithm performance and control its convergence rate. The results demonstrate the positive effect of the FORDPSO on the multi-robot environment exploration. Furthermore, In the work of Sá et al. [66], PSO-based algorithms are used to achieve self-localization in a robotic swarm.

In the work of Suárez et al. [67], the first practical implementation of the bat algorithm in swarm robotics. Their implementation tackled both the physical level, where they design and build a real robotic prototype, called microbat, and a computational level, where the develop a robotic simulation framework based on the swarm-based bat algorithm. An application of finding a target location within unknown static indoor 3D environments was used as testbed to evaluate the performance of the proposed swarm robotic system.

In the work of Rosalie et al. [68], a novel mobility models for multi-level swarms of collaborating unmanned aerial vehicles used for the coverage of a given area is proposed. These mobility models generate unpredictable, *i.e.* that cannot be foreseen at first glance, trajectories using a chaotic solution of a dynamical system. Note that such unpredictability of the offered routes is required in the military context, as explained in Ref. [67]. They detail how the chaotic properties are used to structure the exploration of an unknown area and enhance the exploration part exploiting ACO.

In general, the swarm and evolutionary approaches are applied *offline*: there is an initial *design stage*, when optimizations are performed, followed by an *operational stage* of the robots. An alternative is the usage of *online* approaches, when the optimization is also performed during the operation of the swarm. As seen in Eiben and Smith [50], EC is well recommended for dynamic optimization problems and online learning problems. In the work Mendonça et al. [19], PSO-based algorithms are exploited to achieve efficiently online dynamic task allocation in swarm of robots. Swarm and EC techniques can produce a diverse range of solutions in comparison to other methods, and are widely applicable to different problems. Optimization in robotics is a hard solution because the fitness function is usually noisy, costly and/or implicit. EC can work with these features, while other meta-heuristics, in general, require a well defined fitness functions. However, EC-based algorithms tend to be computationally expensive. This may turn their use in online impractical for some tasks. Needless to say that mobile robotics also offers some solutions to these problems.

3. Development platforms

Design in SR may be divided into two parts: the hardware design and the software design. The former deals with the physical constitution of the robots, like its size, weight, movement and sensing capabilities, and so forth. The latter aims at developing control strategies for the robots, that is, the algorithms that each individual runs internally. The two parts of the system must be designed in such a way that the desired swarm behavior would emerge from the individual behavior of robots, their interactions with each other and with the environment. The validation of the methods or behaviors described by swarms of robots can be performed in several ways. Such experiments include the use of either real or simulated robots.

3.1. Physical swarm robots

The more intuitive method for experimental validation in SR is the use of real robots. The use of physical mobile robots is an option for real-world applications. Several robot models are employed in current research works on SR. There are academic models specifically put together to engineer robots for research purpose only. In contrast, there are commercial models designed to engineer robots for general use.

Robot *S-bot*, shown in Fig. 1(a), is also called *swarm-bot* [69]. It is a mobile robot designed for the *Swarm-bots Project* lead by Dorigo and his research colleague at École Polytechnique Fédérale de Lausanne (EPFL). The main characteristic of an S-bot is its ability to connect and disconnect to and from other swarm members. This capability enables the swarm to perform tasks of self-organization and self-assembly [70]. The robots also have the sensing, processing and communication capabilities, needed to perform different kind of collective tasks, such as navigation and transportation. The extension of this work is the *Swarmanoid Project* [71]. Three different robots, shown in Fig. 1(d), are used to form a heterogeneous swarm. Mobile robots of the same kind of S-bot are called *Foot-bots*, and can move on the ground and explore the environment. The *Hand-bot* is a robot that cannot move, but climbs vertical structures. It can also manipulate objects with its grasping hands. The horizontal movement of a Hand-bot is achieved in association with the Foot-bots. The third robot, the *Eye-bot*, is a flying robot, which can explore and retrieve knowledge about the environment, providing this information to the robots on the ground.

The iRobot *SwarmBot*, shown in Fig. 1(c), is the individual member of the *iRobot Swarm Project*, which was developed by McLurkin [72]. This project aims at studying and developing distributed algorithms to efficiently control a swarm of hundreds of robots. This swarm was successful in performing robot dispersion in indoors using the neighbor-based gradient communication. Another multi-robot system designed by McLurkin is the *R-One* [73]. It is a low-cost robot design for large-scale multi-robot research and educational purposes to younger students. The R-One, shown in Fig. 1(j), has several devices, including a gyroscope, an accelerometer, two-wheel encoders, a global control radio, an infrared beacon for ground-truth localization, and implements infrared inter-robot communication. Its operation can also be expanded for transportation tasks using a *manipulator attachment* [74].

Robot *Khepera*, shown in Fig. 1(l), is a miniature mobile robot developed also at EPFL for use in research and education [82]. Three updated versions were released by K-Team [77,83,84]. Its most recent version, the Khepera IV, shown in Fig. 1(f)), includes twelve infrared sensors, five ultrasonic transceivers, an *inertial measurement unit* (IMU) featuring an accelerometer and a gyroscope, a frontal camera and two wheel encoders. It is also compatible with the external hardware modules made for the earlier versions of Khepera, such as a gripper manipulator and a laser range finder. As a commercial product, the Khepera is in constant evolution, not only to improve its performance including new resources, but also to update its architecture in each version aiming at achieving the best compatibility with recent operating systems, communication protocols and programming languages.

Robot *MarXbot*, shown in Fig. 1(b), is a miniature modular mobile robot that addresses many of needs of collective and swarm systems, such as large battery life, ability to perceive its neighboring robots and interact with them [75]. Note that the MarXbot is the plain version of the Footbot *i.e.* without camera. It exploits differential-drive treels, which is a contraction between *tr*acks and wh *eels*, to provide effective mobility in rough terrains. The MarXbot also allows long and continuous experiments as it implements an advanced energy management and a hotswap (on-the-fly) battery exchange mechanism. Furthermore, it can self-assemble with its peers thanks to a compliant attachment mechanism, thus allowing implementation of swarm robotic systems wherein self-assembly and self-organization is required. This swarm robot also includes a high-quality vision system formed by two high-resolution cameras controlled by a built-in ARM microprocessor.

Robot *Alice*, shown in Fig. 1(g), is another noticeable micro-robot, which was also developed by the EPFL for research work and education [78]. It is actually a very small robots of about 22 mm × 21 mm × 20 mm, allowing a maximum speed of 40 mm/s. It is equipped with two watch motors with wheels and tires, 4 infrared sensors and transmitters for communication and obstacle detection usage. The robots include a PIC microcontroller, accessing 8 K Flash EPROM memory and 368 bytes RAM. Among others, this robot is used in a very interesting experiment that physically mimics the ant foraging behavior [85]. For this experiment, the used robot is augmented with an add-on module, which is also equipped with two upwards photodiodes, of a two-fold purpose: it allows the robot to detect pheromone trails, which are represented by light paths; and it allows a tracking device to follow the robot during the pheromone “laying” operation. This work is actually a proof of concept for the ant foraging efficiency when implemented a robotic swarm. They were able to choose between two paths that link two locations.

Robot Droplet, shown in Fig. 1(h), is an experimental cheap platform for large-scale swarming research [79]. Both its hardware and software are open source. It has an omnidirectional motion, offering 6 linear directions and a turn-in-place possibility. One of main features consists of an unlimited running time without stopping to recharge as they get power from the floor through their legs. It is a very low cost platform as it relies on low-cost vibration motors (slip-stick locomotion). It can acquire various measurements in relation to neighboring robots, which allows it to swarm and collaborate perfectly in application where self-assembly and self-organization are required. It allows for directional communication with user-settable communication ranging from 1 cm to 1 m, bidirectional IR communication, range and bearing sensing, RGB color sensing and self-righting. The droplet project aims at creating a “liquid that thinks” with a horde of such swarming robots [86].

Many others commercial or open-source multi-robot platforms are available, such as robot *Thymio*, shown in Fig. 1(e), which was designed

(a) S-bot [69] (b) MarXbot [77] (c) SwarmBot [72]

(d) Swarmanoid [71] (e) Thymio [80] (f) Khepera IV [76]

(g) Alice [78] (h) Droplet [79] (i) Wanda [82]

(j) R-One [73] (k) Jasmine [81] (l) Khepera [75]

Fig. 1. Different models of mobile-robots used in SR research works.

to be a pedagogical robot for youngsters, robot *Jasmine*, shown in Fig. 1(k), robot *Wanda* [80], is shown in Fig. 1(i), the E-puck [87], the WolfBot [88], robot Pheeno [89], the Kilobot [90] and robot Mona [91]. Due to the miniature size of these robots, their production cost is usually lower than that of complex robots. This reduces the cost of maintaining a set of several robots during the design development stage. It is noteworthy to emphasize that there are many other swarm robots that are not shown here for space limitation only.

The use of simple robots is driven by their appropriateness to perform a range of tasks. A change in some aspects of the desired behavior, such as the inclusion of new features, can lead to hardware requirements that do not exist at the beginning of the project. In some instances such limitations can be overcome with the use of extension boards. For instance, the E-puck is a robot model that supports several extension boards, such as infrared communication [92], Pi-puck extension board, which is a board that allows an E-puck to interface with a Raspberry Pi, Fly-Vision Turret, which features three linear cameras with near-omnidirectional vision, among many other extensions. In other cases, to avoid possible existing limitations on available robots, some projects include developing their own robot models, as is the case of the S-bot for the Swarm-bot Project. However, there is no agreed upon answer to this question. For an interdisciplinary research center, a new mobile robot model can be designed and implemented from scratch. However, this approach is a challenge, which requires specialists in hardware and software integration [93]. On the other hand, if the researchers already have a specific model of a robot and prefer to use it, the work is constrained by the limitations and availability of the model source code. For researchers that want to implement a specific behavior in a commercial model, it may be necessary to verify extensively which hardware resources such a behavior may require. In order to help in this intermediary step of the design, besides testing the idealized controlling algorithm, simulators are also explored to experiment with different combinations of hardware that are necessary to test that required control algorithm. This issue is further discussed in the following section.

3.2. Multi-robot simulators

The use of simulators is another way to perform SR experiments. This is an intermediate step between the implementation at a high level of abstraction, such as using abstract mathematical models, and the validation using physical robots. Simulators can vary in many ways, ranging from multi-agent software for specific academic applications, to commercial simulators with advanced virtual environments. It is hard to find a common ground to compare different multi-robot simulators, which are actually developed with distinct project objective. Nonetheless, in the presentation of ARGoS, which is a simulator presented later in this section, Pinciroli et al. [94], attempt a comparison of some existing multi-robot simulators based on their flexibility and efficiency. Furthermore, for a broader discussion and more generic comparison of simulators and other tools used for robotics applications implementation, Kramer and Schultz [95] propose a useful survey.

The Player Project [96] is an open source simulator for robot, sensor and actuators based research. It includes a wide library of robots and other related devices. Multi-robot experiments can be performed in Stage, which is a two dimensional environment that is linked to Player, and can work with up to 1000 robots. An extension of this project is the three dimensional simulator Gazebo [97].

Enki [98] is a free, fast and two dimensional simulator that supports mobile robots, such as Khepera and s-bot. It provides collision avoidance between robots but limited support for physics simulation. The main characteristic of Enki is the simulation speed, which can simulate robot behavior many orders of magnitude faster than real-time.

Webots [99] is now an open-source three dimensional simulator for single and multi-robot experiments with a good support for physics and collision avoidance. It provides models of real robots and allows the design of robots using the available library of sensors and actuators.

The Virtual Robot Experimentation Platform, V-Rep [100], is a commercial three dimensional simulator with free license for educational use. The V-REP is based on distributed control architecture. That is, each object or model can be individually controlled via an embedded script, a plugin, a remote API client, or a custom solution. The simulator also supports controllers written in a range of programming languages. This makes V-REP very versatile and ideal for multi-robot applications.

ARGoS is an open source multi-physics robot simulator [101]. It can simulate complex experiments involving large-scale robot swarms of any kind, efficiently. Using ARGoS, it is possible to partition the simulated space into multiple sub-spaces, managed by different physics engines running in parallel. Also, ARGoS architecture is multi-threaded, thus designed to optimize the usage of modern multi-core CPUs, aiming at optimizing the overall execution time of the simulations. Last but not least, ARGoS architecture is highly modular, enabling easy addition of custom features and proper allocation of computational resources, which helps in sensing simulation [101].

The Urban Search and Rescue Simulation, USARSim [102] is another an open source high-fidelity robot simulator that can be used both for research and education. It offers many characteristics that differentiate it from most existing simulators. Most notably, it is the simulation engine used to run the virtual robots competition within the Robocup initiative [103]. It builds upon a widely used and affordable state of the art commercial game engine, and is highly configurable.

Another simulator used for several MRS and SR experiments is Roborobo [104], which is based on basic robotic hardware setup like that of the E-puck and Khepera. It can be used both to as robotic system and/or agent-based simulator. When used as the former, it offers real-world options and is easy to use, but it is considered a slow simulation frameworks. Moreover, in the latter case, it offers many imaginary options yet it is easy to use. It has been extensively used in various contexts, such as education and entertainment works. It is also used for evolutionary robotics and swarm robotics [104].

In order to extend the sensing capabilities of the Kilobot, the Kilogrid is used to virtualize sensors not present on the real robot. The Kilogrid is a modular and scalable virtualization environment designed for the study of collective behaviors in swarm robotics using the Kilobot robot [105]. It leverages on the infrared communication capabilities of the Kilobot to provide a reconfigurable environment, whereby sensors and actuators not physically available on the Kilobot can be virtualized. In addition, the Kilogrid eases the task of collecting data during an experiment, thus simplifying a great deal the design of the collective behaviors and their analysis. Furthermore, as the Kilobot has no other actuators than the two vibrating engines, the Kilogrid is also used to extend the actuation capabilities robot, by virtualizing them.

Another hardware/software platform for Kilobots is the Augmented Reality for Kilobots (ARK) [106]. It is composed of an overhead camera tracking system that provides real-time data on robot location and state, a modified overhead emitter, which broadcasts infrared (IR) signals to communicate to the Kilobots and a base control station to coordinate the system and simulate the virtual environments. This is a system that extends the Kilobot capabilities by allowing the robot to operate in augmented reality, allowing the robots access to customized information based on their location and state. In turn, the robots can modify their virtual environment that can then be sensed by other robots in the swarm. ARK can be used, as in the case of the Kilogrid, to extend the set of sensors and actuators of the Kilobot. Additionally, ARK eases and reduces the time required to operate large swarms by automating several necessary steps to set up a given experiment, such as positioning, motor calibration, and unique identifier assignment. Also, ARK provides some handy functionalities to log in and record experimental data for subsequent analysis.

Another interesting initiative of The Georgia Institute of Technology[1] that is worthy of mentioning here is the Robotarium project.[2] It provides a remotely accessible swarm robotics research platform that is freely accessible to anyone. Anyone with novel ideas that could advance the state-of-the-art of SR should be able to see their algorithms deployed on real robots, rather than purely simulated. In order to make this vision a reality, we have created a remote-access, robotics lab where anyone can upload and test their ideas on real robotic hardware.

It is also noteworthy to point out the novel technology, which is actually related to simulated sensing rather than simulators, provides real mobile robots with the capability of sensing an augmented reality environment through some virtual sensors [85]. Virtual sensors are a handy way to check on-the-fly the suitability and efficiency of a given solution without a physical and expensive implementation, which may become later obsolete. This is a very useful way of swarm robot prototyping, which allows researchers to assess the quality and impact of the added sensor before any financial investment can be made. In Ref. [85], to enable and evaluate virtual sensing in real robots, a hybrid platform composed of three main components is employed: a multi-camera based tracking system, a simulator and a robot swarm. The tracking device allows the simulator to acquire the localization and direction of each robot of the swarm in real-time. In turn, the simulator, based on

[1] https://www.gatech.edu/.
[2] https://www.robotarium.gatech.edu/.

the received data, computes the readings of the virtual sensors. It then delivers via WIFI the appropriate values, as they were yield by a physical built-in sensor of the robot. Note the ARGos is used as the simulator and the tracking system includes a 4×4 camera matrix, which is controlled by a 16 cores host, running a custom vision machine. As a proof of concept, a virtual pollutant sensor is implemented in a swarm of fifteen E-pucks. During the experiment, the robots move randomly within an arena and, as soon as robot senses the pollutant, it indicates it via its red LEDs.

3.3. Human/swarm interaction

When studying SR an issue, which eventually arises involves the interaction between humans and robots, that is, *Human-Robot Interaction* (HRI). Robots can be programmed and released in the environment to perform some task. However, it may be necessary to intervene in robot's work, without the interruption of the task. For instance, it must be necessary for the programmer to retrieve some internal information about the execution, such as program termination. In this case, there is a *robot to human* interaction. The reverse may also be needed. In *human to robot* interaction, the robot should be able to receive and understand external inputs provided by humans. This has a variety of uses. For instance, it helps to indicate that the swarm must change its behavior, or even to force an interruption of the task.

In many cases, robot to human interaction can be achieved via some robot's output devices. The simplest example is the use of a LED present in robot's body as a flag, indicating whether a specific condition is met. For instance, the Kilobot robot [90] uses a color LED as the only[3] communication way during task execution to indicate its internal states. McLurkin et al. [107] codified different patterns of blinking, intensity and color of SwarmBot LEDs to compose a *swarm language*, enabling what they call *hands-free operation*. They also expanded this idea to audio devices, wherein the robots generate sound patterns. The drawback of this "language" is the complexity of the light and sound patterns, which requires an external monitoring system.

Interaction in the reverse way, from human to robot, is a more complicated task, because robots must read and interpret external signals generated by the users. Robots must rely in their sensing capabilities, such as microphones and ambient light sensors. For instance, the E-puck [87] is an example of robot equipped with microphones, which enable it to detect sound sources. For practical use, the robot hardware must be fast enough to acquire and process the external signal, and then take the correct decision. Physical switches, present in robot's hardware, as in the case of the E-puck and SwarmBot, may also be used as input devices, but this goes against the *hands-free* idea. Communication interfaces, such as WIFI and radio, are very useful for both receiving and sending information, but these require an external communication system to manage the messages.

A most advanced interaction scheme uses video acquisition. Robots equipped with cameras can literally "see" the environment, and humans can use this capacity as a communication channel. In the work of Zhang and Vaughan [108], video processing is used to accomplish human to robot interaction. Humans can select one robot from a group just by looking at it. Each robot detects which human is trying to make the selection using face recognition algorithms. Proving the capacity of detection of a variety of faces, the authors intend to use similar techniques to allow the robot to interpret different facial expressions. In another work, Giusti et al. [109] achieved a *human-swarm* interaction by the cooperative sensing of hand gestures. Vision-based recognition is a hard task for a single robot, making the use of multiple robots mandatory. Each robot is supposed to capture hand gesture using a neural network classifier. The classification of each robot is then considered in a posterior consensus process. Vision-based human to robot interaction seems to be the most suitable and intuitive for general-purpose process, because there is no need for an external apparatus to send the signals to robots. The main problem in this strategy is the requirement of more processing power in each robot to handle and understand the video information.[4] In contrast with gesture based-interaction, Mondada et al. [110] propose a new implicit proximal communication technique to approach the problem of robot selection. They exploit electroencephalography signals to select the robot at which the operator is looking. This is achieved using steady-state visually evoked potential, which is a repeatable neural response to a regularly blinking visual stimulus that varies predictively based on the blinking frequency. Note that robot Thymio can also be controlled by a brain-computer interface. Le Goc et al. [111] introduce swarm user interfaces as a new class of human-computer interfaces comprised of many autonomous robots that handle both display and interaction. They describe the design of Zooids, which is an open-source open-hardware platform for developing tabletop swarm interfaces. The proposed platform consists of a collection of custom-designed wheeled micro-robots, a radio base-station, a high-speed Digital Light Processing light projector for optical tracking and a software framework for application development and control.

3.4. Range and bearing

For most tasks related to swarm robotics, such as target tracking [112], pattern formation [113], controlled navigation [114], it is necessary for robots to acquire their localization and also the whereabouts of their neighboring robots. This can be simply done on the global level, where a central entity finds out the positions of the robots of the swarm, and informs them about their locations and/or that of others in the swarm. However, in a distributed system, there is no such a single global entity. In general, the usage of a central entity degrades the scalability and robustness of the swarm robotic system. Global localization could still be accomplished using GPS techniques, but GPS is not always available, which would limit the environment, wherein the swarm could be effective. An alternative approach is to obtain the robots localizations in a distributed fashion. This consists of endowing the robots of some device [115] and/or algorithm [66,116] that are able to deduce the required localization locally between pairs of robots and generalize it to encompass the whole neighborhood or even the whole swarm.

In swarm robotics, the range of a robot with reference to another one, usually in its neighborhood, is defined by the distance between these two robots. Furthermore, the bearing of a robot with respect to another one in the vicinity refers to the angle between formed between them. Estimating the range and bearing of a robot to another or to an object is a necessary functionality for building a localization system to determine the robot's current position and direction. This is usually known as the pose of the robot. The most common application in robotics that depends heavily on robot's poses is the SLAM (Simultaneous Localization and Mapping), which allows the robots to self localize and yet build a map of an unknown environment [117].

There are many relevant research works that are concerned with determining the relative position of a robot and that of its neighbors. Pugh and Martinoli [118] characterize and improve an existing infrared relative localization/communication module used to find range and bearing between robots in small-scale multi-robot systems. Later, they used the proposed module to explore the problem of small scale multi-robot formation [119]. After that, Pugh et al. proposed an on-board robotic module that can determine relative positions among many miniature robots. The module uses high-frequency modulated infrared

[3] Kilobot has also a wired communication channel, but its use is not practical when robots are moving.

[4] A video illustrating the described behavior: https://bit.ly/2XoagV5.

Table 1
Sensing and communication of robotic platforms.

Robot	Sensors	Movement and Actuation	Communication
S-bot	IR, ambient light, accelerometer, encoders, omnidirectional camera, microphone, humidity, temperature	wheels, tracks, gripper	IR, WIFI
R-One	IR, ambient light, accelerometer, encoders, gyroscope, bump	wheels, gripper	IR, RF
Khepera IV	IR, ambient light, accelerometer, encoders, gyroscope, camera, microphone, ultrasonic	wheels, gripper	IR, WIFI, Bluetooth
E-puck	IR, accelerometer, camera, microphones	wheels	IR, bluetooth, serial
Wolfbot	IR, ambient light, accelerometer, magnetometer, camera, microphone	wheels	WIFI, zigbee
Pheeno	IR, accelerometer, encoders, magnetometer, camera	wheels, gripper	WIFI, serial, Bluetooth
Kilobot	IR, ambient light	vibration motors	IR
MarXbot	IR, camera, microphones, axis force, accelerometers, gyroscope	wheels, tracks, grippers, range & bearing, scanner	IR, RF, WIFI, Bluetooth
Alice	IR, camera tactile	wheels	IR, RF, zigbee
Droplet	IR, RGB color, range & bearing	legs	IR
Thymio	IR, Temperature, accelerometer, microphone	wheels	IR, WIFI, speaker
Jasmine	IR, proximity, touch, color and distance sensors, ambient light	stepper drives	IR, zigbee
Wanda	IR, range & bearing, RGB sensor	wheels, passive gripper	IR

Table 2
Illustrative application for the considered wheeled robotic platforms.

Robot	Application	illustrative video
S-bot	Swarmanoid Project	https://bit.ly/2I0enmv
R-One	Collective exploration	https://bit.ly/2Vo80wm
Khepera IV	Monitoring system	https://bit.ly/2UrNptR
E-puck	Child pulling	https://bit.ly/2OP9eOG
Wolfbot	Bot fight	https://bit.ly/2Uuurm9
Pheeno	Collective transport	https://bit.ly/2Kh8aEq
Kilobot	formation	https://bit.ly/2FTXB51
MarXbot	Swarmanoid Project	https://bit.ly/2I0enmv
Alice	Foraging	https://bit.ly/2Ufo1s3
Droplet	Smart liquid	https://bit.ly/2HXwo4X
Thymio	Brain-computer interface	https://bit.ly/2XoagV5
Jasmine	Pheromone robotics	https://bit.ly/2GpBJ2G
Wanda	Cleaning up	https://bit.ly/2V9TZWl

emissions to enable robots in the neighborhood to determine the range, bearing and receive message of the sender with a rapid update rate. They showed that the new module can maintain a precise multi-robot formations throughout difficult maneuvers. Kelly and Martinoli [120] developed an on-board localization system using infrared sensors for indoor applications. Roberts et al. [121] propose a three-dimensional relative positioning sensor for indoor flying robots was proposed, which is designed to enable the spatial coordination and goal-directed flight of inter-robot.

3.5. Platforms discussion

Various robotic platforms used in SR research share as many characteristics as they do not share. These differences eventually prevent the implementation of a variety of SR related tasks. In Table 1, some features of robots, widely used in SR studies, are shown. Moreover, Table 2 shows an illustrative application for each of the platforms listed in Table 1. Nonetheless, it is noteworthy to point out that these robots were not designed as swarm robots only. They can be used individually to automate tasks.

The S-bot and the MarXbot as well as Jasmine [81] for example, are one of the robots with the largest set of sensors, ranging from typical infrared (IR) to unusual humidity sensors. This robot is project-specific (from the Swarm-bots Project) and was intentionally designed with the hardware necessary in most swarm tasks. In the other extreme, the Kilobot and to some extent the Droplet has the simplest hardware specification. The former has single IR sensor is used to detect the distance to other robots. This limitation restrains the tasks that a swarm of Kilobots can perform, but it is also interesting as it allows the exploration of the collective work using very simple devices. IR sensors are present in all robots listed in Table 1. IR sensors are required in obstacle detection and during communication between robots. The actuation capabilities vary from a robot to another. For instance, robots Thymio [76] and R-One [73] are equipped with a grabber, which comes handy for applications that require grabbing objects. Most robots use wheels to move, except for the Kilobot, which has three slip-sticks and uses the so-called slip and stick technique to move. The s-bot also has a track system in addition to the normal wheels to increase the torque. The Wolfbot and the Droplets have a unique omnidirectional movement system composed of three wheels and sticks, respectively. This allows an increased mobility in comparison to other platforms, which are normally based on nonholonomic (Ackerman vehicle) systems [122]. In the case of the Droplet, it is noteworthy to point out that it only moves in a powered floor, which allows the swarm to be used in endless swarming experiments without stopping for battery re-charge.

Despite the advances on processing and sensing capabilities, there are still limitations in communication. The IR sensors in most robots can be used for inter-robot communication, but it is very limited in

Table 3
Characteristics of simulation platforms.

Simulator	Environment	Physics	Robots	License
Gazebo	3D	Yes	Robot library and customized robots	Open source
Enki	3D	No	Robot library and customized robots	Open Source
Webots	3D	Yes	Robot library and customized robots	Open Source
V-Rep	3D	Yes	Robot library and customized robots	Educational
ARGoS	3D	Yes	Foot-bot and customized robots	Open Source
USARSIM	3D	Yes	Robot library and customized robots	Open source
Roborobo	2D	No	E-puck and Khepera based	Open source

Table 4
Illustrative application for the considered simulation platforms.

Simulator	Application	Illustrative video
Gazebo	Foraging	https://bit.ly/2UhYjmE
Enki	Aggregation	https://bit.ly/2K6RrDK
Webots	Task allocation	https://bit.ly/2FZK398
V-Rep	Collective navigation	https://bit.ly/2IbLIdj
ARGoS	Wall building	https://bit.ly/2TZ767L
USARSIM	Environment mapping	https://bit.ly/2Vqag6c
Roborobo	Gaming	https://bit.ly/2UhZwuc

terms of amount of data that can be sent/received. This constraint is analogous to communication in real swarms, in which swarm members use only local communication. The robots in Table 1 also present different versions of media or long-range communication, but they are mostly directed to communication with a central host monitoring system, used to receive video streaming as captured by a robot's camera. An exception to this is the Wolfbot and Alice, which are equipped with a *zigbee* module that can establish a mesh network with other peers, allowing the broadcast of messages through the entire swarm. Even though this seems to go against the main point in SRs, hindering the scalability of the system, the inclusion of this kind of communication appears to be essential in future SR platforms. Taking into account that most robot communication in SR is directed equally to all the robots in the neighborhood, broadcast communication mitigates the overhead of point-to-point communication. It is noteworthy to point out that this does not increase the project cost, yet it enables a variety of tasks, which rely on information sharing.

As in the case of real robots, simulation platforms are also diverse in terms of resources. The main characteristic that differs between simulators is their abstraction level. Some of them simulate three dimensional environments with accurate physics. The others perform fast two dimensional simulations of robots interactions. The most known simulators are listed in Table 3. Again, it is worth mentioning that these simulators are not designed to be used for swarm robotic system solely. Also, the applications cited for simulators, as listed in Table 4, and these are only some illustrative examples from the many applications. These simulators have been extensively used in many different swarming tasks such as collective transport, collective motion and foraging.

4. Classification of SR research

There are many university and research institutions that are investing effort in advancing the state of the art in swarm robotics. The most famous and publicized, of course among many others, are listed in Table 5 together with the related research project or research group.

The research on SR is very broad, and is an extension of the study on multi-robot systems (MRS). Authors usually try to classify such works following some common characteristics, but there is no global consensus about this classification. In fact, this is an area in constant evolution, with new challenges arising every day. Several published taxonomies are discussed in this section. These classifications are included in review papers on SR and MRS that are the most referenced in literature.

One of the early works that proposed a taxonomy of the swarm research works was done by Dudek et al. [123]. Their classification focused on the robots and on the swarm itself, but not on the tasks or the steps related with the swarm design. In their research, the authors classified swarms considering the following aspects:

- *Swarm Size*: the number of robots defines the swarm characteristics.
- *Communication Range*: a robot can communicate with: (i) all other robots in the swarm, which is identified as broadcast communication; (ii) only with robots within a specific distance, which is identified as local communication; (iii) or the robot cannot communicate directly, but can only senses the presence or behavior of neighboring robots, which is identified as indirect communication.
- *Communication Topology*: robots can communicate with each other in a broadcast way, using addresses, or through some kind of hierarchy.
- *Communication Bandwidth*: the communication has a higher cost than robot movement.
- *Spatial Reconfiguration*: the robot topology can be fixed, can be rearranged or is dynamic, changing arbitrarily.
- *Unit Processing*: the way the robots process information.
- *Composition*: the swarm can be homogeneous, when all robots are of the same type, or heterogeneous, with robots of different kinds.

The concept of *cooperative behavior* was introduced in the works of Cao et al. [124] and Mataric [125], with a group of individuals working together to achieve some common goal. It contrasts with the concept of collective behavior in multiple robot systems, where several individual are dedicated to doing the same task, thus accelerating the process. The classification of swarm robotic related works is divided into five *Research Axes*: Group Architecture, Resource Conflict, Origin of Cooperation, Learning and Geometric Problems. The *Group Architecture* deals with the infrastructure of the swarm. The *Resource Conflicts* is the research axis related to the solution of problems that arise from the presence of multiple robots in the same environment. The *Origin of Cooperation* focuses on how cooperative behavior is actually motivated and achieved. *Learning* is based on the capacity of robots to change and optimize their own control parameters. *Geometric Problems* is the research axis that explores applications and robotic swarm behaviors that depend on robot position, distance and other spatial related issues.

In Iocchi et al. [126], another taxonomy based on four levels representing different aspects of robots is proposed. These levels are shown in Fig. 2. The first aspect is the *Cooperation Level*, followed by the *Knowledge Level*, wherein each robot is aware or not of the presence of other swarm members. For the systems with robot awareness, there exists a *Coordination Level*, which is divided into Strongly Coordinated, Weakly Coordinated and Not Coordinated. For the Strongly Coordinated, there is an *Organization Level*, divided into Strongly Centralized, Weakly Centralized and Distributed. All systems in this taxonomy can also be characterized as *Social Deliberative* or *Reactive*.

In Bayındır and Şahin [127], the taxonomy is divided into five main axes: Modeling, Behavior Design, Communication, Analytical Studies

Table 5
Some Universities and research institutions working in swarm robotics.

Institution	Group/Laboratory	Homepage	Main robot
Free University of Brussels	Swarm Intelligence and Robotics	http://fondation.ulb.ac.be/en/swarm-intelligence-robotics/	Foot-bot
Federal Polytechnic School of Lausanne	Autonomous Systems	http://mobots.epfl.ch/	E-puck
Harvard University	Self-Organizing Systems	https://ssr.seas.harvard.edu/	Kilobot
University of Colorado Boulder	Correll Robotics Lab	https://www.colorado.edu/cs/2018/08/02/droplets-swarm-robotics-platform	Droplet
Rice University	Multi-Robot Systems	http://mrsl.rice.edu/	R-One
Federal Polytechnic School of Lausanne	Autonomous Systems Lab	http://www.asl.ethz.ch/research.html	Alice
The University of Manchester	Robotics for Extreme Environments	https://www.eee.manchester.ac.uk/research/themes/robotics-for-extreme-environments	Mona
Federal Polytechnic School of Lausanne	Robotics Systems	https://www.thymio.org/	Thymio
Universities of Stuttgart & Karlsruhe	Open-source Micro-robotics	http://swarmrobot.org/	Jasmine
Karlsruhe Institute of Technology	Swarm Robotics, Modular Self-Reconfigurable Robotics, Simulation	https://www.ipr.kit.edu	Wanda
Massachusetts Institute of Technology	Computer Science and Artificial Intelligence	https://www.csail.mit.edu/	Particle

and Problems, as shown in Fig. 3. The *Modeling* axis is the study of how the swarm is organized and whether the swarm model is macroscopic or microscopic. The *Behavior Design* is related to the learning capabilities of robots. *Communication* defines whether the robots can communicate with each other directly, indirectly or by sensing. The axis *Analytical Studies* involves the mathematical and statistical studies of the swarm. The most characteristic research axis of this taxonomy is the *Problems* axis that includes the collective tasks that a swarm of robots will effectively perform. In Bayındır [128], there is a review of the different kind of collective tasks that can be performed by robotic swarms.

A more concise taxonomy was proposed by Brambilla et al. [24]. According to this taxonomy, the works on swarm robotics are divided in two classes: Methods and Collective Behaviors, as shown in Fig. 4. The former includes design and analytical methods while the latter is related to tasks and problems preformed by swarm members. The authors enumerate the main collective behaviors studied in literature and classify these behaviors in four categories: spatially organizing behaviors, navigation behaviors, collective decision-making and other collective behaviors.

The analysis of the reviews in SR is an interesting way to understand not only how this field has evolved over the years, but also how it is interpreted by researchers. In earlier works, there was a need to classify the systems based on some global characteristics, such as architecture organization of the swarm, as done by Dudek et al. [129] and Cao et al. [31]. This point of view changed in the work of Iocchi et al. [126], wherein the characteristics are dependent of each other, forming different levels. Even so, these taxonomies are more focused on how the systems are organized to perform a given task, but not on what are the different behaviors that the SR can perform. We noticed that most recent reviews classify SR works as task and behavior centered, where the task itself is evaluated, and robot and swarm centered, where the focus is the swarm hardware, the infrastructure, the design methods. In this review, we followed the classification found in Bayındır [127] and Brambilla et al. [24] and, which seems to be currently the most accepted in the literature.

Considering the many different classifications found in literature, in this review the works on SR are divided in two large groups: one is based on *methodologies* and the other one on *tasks*. Each one of these groups has its own subdivisions. These two classes of works will be addressed in the following sections. In fact, a class encompassing behaviors, tasks, problems or applications performed collectively by the swarm is present in most of the existing taxonomies, not only in the previously cited, but also in other reviews [130–133].

5. Design methodologies

SR is an interdisciplinary field of study, wherein the comprehended disciplines range from mathematical modeling of biological systems to the design and implementation of physical agents. Although the final interest in the use of robots should be the application, which is the task that robots must perform, other aspects are also studied. This includes works on how the robots interact, how the task can be performed considering the swarm as a single entity and how the task can be mathematically described. Some of these issues are addressed in this section.

5.1. Modeling

The design and analysis of SR systems are challenging tasks. Mathematical models of the swarm are necessary to evaluate several aspects, for instance, if a given task is feasible, how many robots are needed in the swarm to achieve a specific behavior, and so forth. Such models may be expressed in different levels of abstraction. Modeling of swarm systems is usually classified in two categories: microscopic and macroscopic [24,127,131,132]. The *microscopic modeling* approach takes into

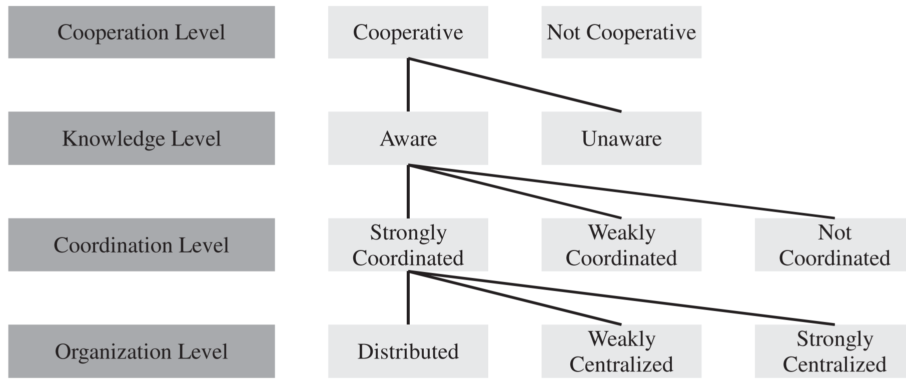

Fig. 2. Levels in the taxonomy proposed by Iocci et al. [126].

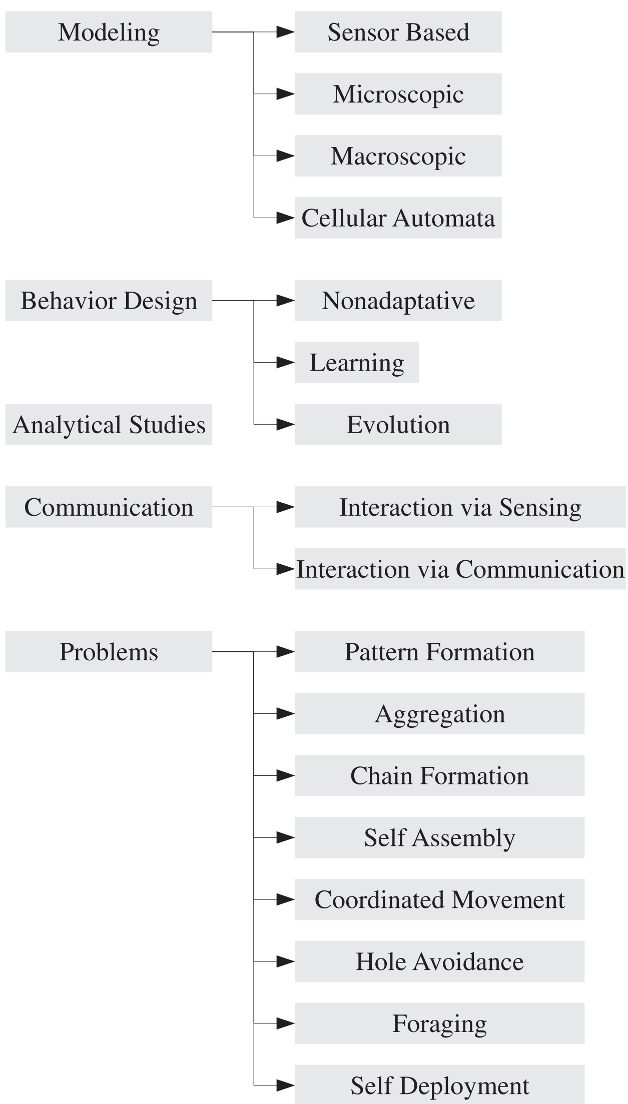

Fig. 3. Taxonomy proposed by Bayındır and Şahin [127].

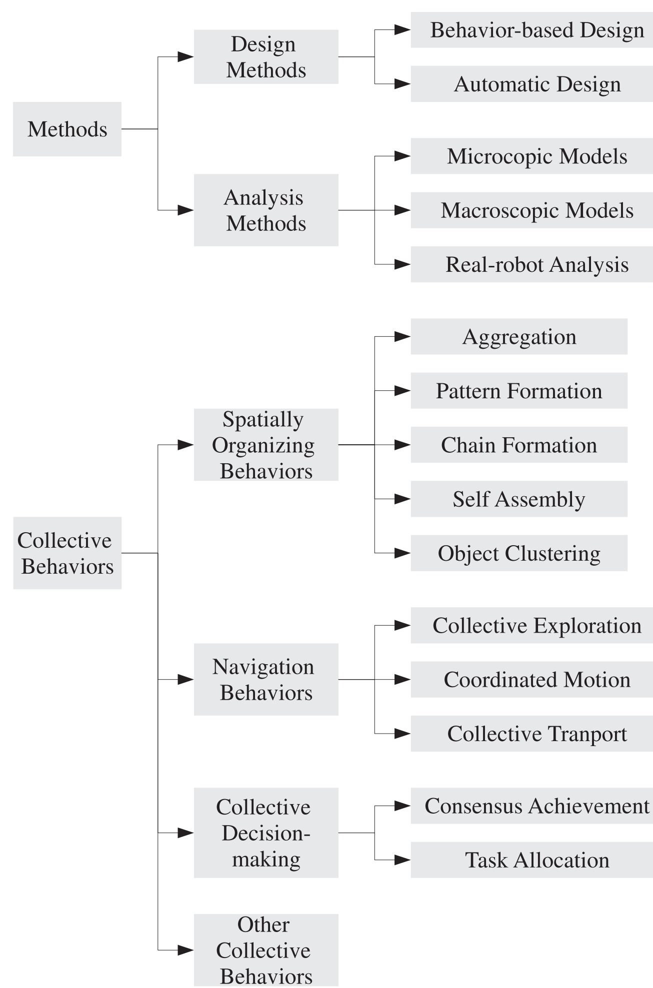

Fig. 4. Taxonomy proposed by Brambilla et al. [24].

account the interaction of swarm members while the *macroscopic modeling* approach concentrates on the entire swarm at a higher level of abstraction. These two views have some drawbacks. It is difficult to determine the local interactions of individuals that should lead to some predefined global behavior and vice-versa. As result of that, most analysis only consider either the microscopic or the macroscopic level of abstraction, but not both. Even though there are many interesting models, it is extremely difficult to find common ground criteria to rate them as they serve different modeling purposes and take care of different facets of the system. In the following, we describe some of these works.

In the work of Muniganti et al. [134], the authors report on the mathematical models that can be used to understand and estimate the overall mechanism that governs the swarm systems. These are mainly based on *probabilistic models*, *differential equations*, *dynamical systems* and the *automata theory*. Each one of these schemes can be more suitable to model a specific task.

Mermoud et al. [135] compare the strengths and weaknesses of microscopic and macroscopic model-based methodologies, called therein as *bottom-up* and *top-down*, for designing distributed controllers. The authors show that top-down design approaches are not generally suitable in every case. In contrast, bottom-up strategies seem to be more robust.

Lerman et al. [136] review the methods used for macroscopic modeling and analyzing collective behavior of swarm robotic systems. They show that the collective behavior of swarms is statistically predictable and has often a simple probabilistic description. They propose a class of mathematical models that describe the dynamics of collective behavior can be generated using the individual robot controller as modeling blueprint.

Hamann and Wörn [137] propose a model aiming at an explicit representation of space. They claim that, in general, the effectiveness of a swarm robotic scenario depends on spatial inhomogeneity. Statistical physics related methods are exploited to address spatiality. First, an abstract model of model of swarm motion based on the specification a single robot is derived. Then, the model is extended to a generic model framework regarding the robots interaction via communication. The generated models are validated using some simulation results, which show that qualitative correctness is easily achieved while quantitative correctness is disproportionately more difficult but still possible.

According to state-of-the-art applications modeling in SR, it can be noted that probabilistic models are suitable for and have been used in applications related to foraging, chain formation, stick-pulling, aggregation tasks, cooperation and transportation, clustering and sorting, assembling based on nest building of wasps and termites and flocking and schooling. Models based on automata theory have been most used and hence are suitable for applications related to collaborative mapping, unknown environment exploration, lattice formation and space related systems, such as pattern formations. Furthermore, differential and partial differential equations based models have been suitably used in pheromone controlled applications, trophallaxis and stigmergic systems as well as problems related to collective foraging and spatial systems, clustering and sorting, flocking and schooling, attraction, alignment and repulsion based systems. Last but not least, model based on dynamical systems are considered suitable for cohesion and stability analysis, beacon and odor localization, bifurcation models, attraction, alignment and repulsion systems as well as aggregation tasks and population dynamics of swarms [134].

5.2. Planning

The design of complex behaviors is necessary for the development of effective SR-based applications. So, a *planning* strategy aims at defining how a given task can be divided into a number of sub-tasks, providing a schedule for their interdependency and defining the possible duration of each sub-task, among other related issues.

A planning scheme for multi-robot cooperation is proposed in the MARTHA Project [138]. It uses local communication to continuously plan the operation of a fleet of autonomous robots in places such as airports and harbors. The work presents the *Plan-Merging Paradigm*, which basically addresses the issue of resource contention. When a robot wants to elaborate or execute a plan, it must first validate it with its neighboring robots in order to create a group plan.

In Ziparo et al. [139], the authors use Petri Nets to formulate such plans, called *Petri Net Plans* (PNPs), which allow for the design of intuitive and effective multi-robot behaviors. The goal of PNPs is to support developers in designing and implementing complex high-level robot and multi-robot behaviors, by providing a rich modeling language that offers several key features, such as concurrency, distributed execution and formal analysis, which are very often required in robotic systems.

A slightly different concept is the *Task Abstract Module* described in the work of Brutschy et al. [140]. Considering a complex task composed by multiple sub-tasks, the authors proposed and implemented a hardware device that emulates the start and conclusion of such sub-tasks. It enables the swarm of real robots to perform abstract tasks. It can be used in any other SR-based works, where the focus is on the interaction among robots rather than on the different kind of tasks to be performed by the swarm.

5.3. Evolutionary robotics

Evolutionary Robotics (ER) is a design methodology that makes use of the concepts of Evolutionary Computation to design robotic systems [17]. Generally speaking, in ER, the main objective is to design robot controllers automatically. In some cases, however, the execution of an evolutionary algorithm can also aim at improving, in an effective way, some robot characteristics. This includes the calibration of both sensors and actuators, the identification of other robots and obstacles in the environment. Furthermore, the usage of ER does not aim only at the optimization and adaptation of individual and collective robots behaviors, but also at the robot's physical design [141].

In the work of Duarte et al. [142], the ER techniques are used to evolve the controllers of a swarm of aquatic surface robots. The controller characteristics were evolved in simulations for a variety of tasks, such as dispersion and clustering. The controllers for simulated robots were then transferred to real robots, achieving similar performance in both the cases. The main difficulty reported in their work is possibly the accurate modeling of the environmental variables. Moreover, the researchers needed to design their own model of aquatic robot. It included wireless communication, a compass and a *Global Positioning System* (GPS) to assist during navigation. It is noteworthy to point out that this project consisted of a sophisticated hardware specification in comparison to other SR platforms.

Gomes et al. [143] used the so-called *novelty search* as an ER scheme for a swarm of simulated robots. In novelty search, solutions are rewarded based on their novelty, rather than their quality with respect to predefined objectives. The authors used this approach in a common aggregation task as well as in a resource-sharing task. Novelty search allowed for a better performance than fitness-based evolution, avoiding stagnation in local maxima.

Nitschke et al. [144] performed an extensive analysis of *Collective Neuro-Evolution* (CONE) in comparison to two other ER methods. Such techniques were employed during the evolution of multiple robot controllers for a collective construction task in a simulated swarm of Khepera robots. The concept of *behavioral specialization* is also described: the construction task is defined as a set of sub-tasks, and each robot is supposed to be able to specialize in one of such sub-tasks. The authors show that, in comparison to other ER techniques, the use of specialization in CONE yields a higher performance regarding the convergence time, required to achieve the expected collective behavior.

Trianni and López-Ibáñez [145] explored the advantages of multi-objective optimization over single-objective in evolutionary robotics. Using single robot and multiple robots, the authors demonstrate that multiple-objective optimization allows for the evolution of more varied behaviors and avoids convergence to local optima. However, the authors also show that the optimization approaches could be more or less suitable depending on the case study. Multi-objective optimization is a better option when there is not enough knowledge about the problem to be solved, while single-objective evolution proved to be more efficient when a reliable yet precise fitness function is available.

Trianni [146] presents a seminal work, which is a collection of very interesting research works on the exploitation of ER techniques for the design of self-organizing collective behaviors in both simulated and real environments. The works represent a showcase the usage of ER as an actual tool to achieve automatic synthesis of complex behaviors in a robotic system. Among the analyzed topics is how ER can be used to reach self-organization in swarm robotics, coordinated navigation, obstacle and hole avoidance, and collective decision-making mechanisms.

6. Collective behavior tasks

Several problems, tasks and behaviors are studied in SR research works. Some of the tasks, which are most discussed in literature are commented on this section. Basic behaviors are usually used to compose complex tasks, which are closer to real world problems. Such complex behaviors may be used as benchmarks to assess the performance of newly proposed methods. Most of these methods are nature-based, having as inspiration the collective behavior of animal and insects societies. Following the division described by Brambilla et al. [24], SR tasks deal, in general, with either (i) the spatial organization among the robots, among the robots and the environment or both; (ii) the motion of the robots in the environment, including the search, identification and movement of objects; and (iii) the collective decision-making process, also called the *consensus* problem. Tasks that fall into these three classes are described later in more details. The tasks, which appear more often in the literature, expose some trends in SR research works. Different implementations of these tasks are enumerated, including the most cited and most recent.

6.1. Spatial organization

The tasks that are dependent on the *spatial organization* of the swarm are characterized by the movement of the robots, usually increasing or reducing the distance between each other, hence resulting in the emergence of a particular property. Some of the most known spatial organizational tasks are *aggregation*, *dispersion*, *coverage* and *pattern formation*.

6.1.1. Aggregation

Self-aggregation is a task, where swarm members, initially dispersed in the environment, must aggregate to become closer to each other in the same spatial area. It is a simple collective behavior present in many insect societies [147]. In the context of SR, however, this task has no trivial solution because the robots must move to change their distance to neighboring robots, while avoiding the creation of multiple spatially separated groups. The aggregation is very useful for other tasks when a significant number of robots are needed, as in the case in collaborative transport [148,149] and pattern formation [150]. In Ref. [151], Arvin et al. propose a classification of the aggregation task as cue-based or self-organized. In a cue-based aggregation, there is a cue in the environment that points to the aggregation area, whereas in self-organized aggregation no cue is given. The authors proposed a novel fuzzy-based method for cue-based aggregation based on the BEECLUST algorithm [18].

Aggregation approaches were designed in several studies. These are usually inspired by insect aggregation strategies, such as the one used by honeybees [18,152] and that of cockroaches [153]. Honeybees present a *thermo-tactic* aggregation behavior, tending to stay still at places of high temperature. In a field with a temperature gradient, each bee stops together with other bees, but starts to move again after some time that is proportional to the local temperature. Cockroaches use a similar behavior, but they base their decision on the local light intensity instead of local temperature. They tend to aggregate in dark places. A cockroach wonders randomly, and stops after some time at a given point based on the luminosity in that place as well as the number of other individuals that are already still at that point. In both cases, there is external information that coordinates the behavior augmented by some inter-agent communication. For instance, the strategy developed by Schmickl and Hamann, called BEECLUST algorithm [18], makes use of environment light, instead of temperature, to allow for robots aggregation [152]. In the work of Garnier et al. [153], both simulated and real robot experiments use light and dark places to control the probability of aggregation. In the work [154], Arvin et al. present an implementation of a bio-inspired aggregation scenario using swarm Mona robots. In Ref. [155], Firat et al. exploits the aggregation behavior to help the swarm robots to self-organize to aggregate in one of the available sites.

6.1.2. Dispersion

A behavior that is opposite to self-aggregation is *dispersion* or *self-deployment*. In this case, the robots must deploy themselves in the environment, occupying a large area in a cooperative way. In other words, the dispersion of the swarm is usually needed when the explored area must be expanded without loss of connectivity. There are many approaches to this problem.

In McLurkin and Smith [72], the dispersion task is performed by a group of SwarmBots using only inter-robot communication. The network of robots forms a gradient, and the flood of messages guides the robots movement. A different approach used in the works of Ludwig and Gini [156], Ugur et al. [157] and Necsulescu and Schilling [158] consider the intensity of the received wireless signal to estimate the distance between robots. In both works, the robots try to maximize the covered area while preserving the connectivity within the swarm. Other approaches include the use of virtual forces [159], potential fields, landmarks [160], shared taboo lists [161] and concepts of iso-probability curves [162].

A particular case of self-deployment is the *area coverage* problem. In this task, the robots must spread out, covering equally a given area without a prior knowledge about the environment. It is useful in many real-world applications, such as the mapping of unknown regions and the use of the swarm as a sensor network. In many works [163–166], multi-robot coverage uses the communication of social insects as inspiration to regulate the mutual distance among the swarm members. In the works of Howard et al. [39] and Reif and Wang [167], the inter-robot distance is controlled by virtual forces, potential fields, or both.

6.1.3. Pattern formation

Pattern formation is a spatial organization problem, where the robots must re-arrange themselves to achieve some global spatial pattern. Through local interactions, the robots adjust their positions forming a regular and repetitive structure. Self-organized formations can be found in nature in different scenarios, such as honeybee nests [168], bacterial colonies [169] and crystalline structure of solids [170].

Physicomimetics Framework is proposed by Spears et al. [150] as a fully decentralized robot control inspired by physics. In this strategy, the robots can sense the relative position of their neighbors, and react to attractive and repulsive forces, resulting in lattice formations. In Flocchini et al. [171], the robots can describe any arbitrary pattern using two independent compasses as external reference. In Alonso-Mora et al. [172], a given pattern is divided uniformly using Voronoi partitioning, which indicates the final destination of each robot. In this case, there is global information that is shared among the members of the swarm. In Yamashita and Suzuki [173], robots perform a "look-compute-move" cycle to meet some geometric patterns. In these works, the authors show that some patterns are possible to achieve and other are impossible to be realized by oblivious (without memory) or non-oblivious (with memory) robots.

6.1.4. Self-assembly

An extreme example of self-organization is the *self-assembly*, where the robots not only stay close to each other, but they also are able to connect themselves, forming a single organism. This behavior is inspired by the *symbiotic organization* found in nature, where independent species evolve to live connected to each other, hence having a better chance of survival.

The *Symbrion Project* [174] put up a major effort in developing self-assembly related tasks (the name denotes “Symbiotic Evolutionary Robot Organisms”). In the Symbrion framework, modular robots can work as a typical swarm, moving and exploring the environment. The robots are also able to self-assemble to form a three dimensional organism. To implement this concept, it was necessary to develop three types of modular robots, each one with a different functionality, forming thus a heterogeneous swarm [175].

6.2. Collective motion

The individuals in SR are mobile devices, there is, wheeled, aerial or underwater robots which can move in the environment. Given this basic characteristic, different tasks based on the swarm motion can be considered, such as the movement of a single or multiple robots, the avoidance of obstacles, the identification of other robots or objects, among many others tasks.

The robot movement in these motion-related tasks differs from those regarding self-organization tasks. In collective motion tasks, the entire swarm is considered to go from one place of the environment to another. In contrast, in spatial organization tasks, each robot moves within the swarm aiming at some re-arrangement. However, there is no real displacement of the swarm. In this section, tasks based on swarm movement are presented and discussed, including those related to exploration, foraging, collective navigation and collective transport.

6.2.1. Exploration

The exploration of unknown environments is a fundamental problem in robotics. It is useful in many real-world applications, wherein the environment is hazardous to human life. In the single-robot case, the robot must wander in the environment, registering the information about obstacles found. One known example of this task is the *Simultaneous Localization And Mapping* (SLAM) [176], in which the robot builds a map of the region around itself. The robot must be able to localize itself inside the map for subsequent use in other application. When area exploration is performed by a swarm of robots, it is also called *collaborative exploration*. This instance has advantages over the single-robot case. Multiple robots can perform the entire task faster than a single one due to parallelism. Also, the redundancy of information can compensate for sensor uncertainty. Moreover, the presence of multiple agents can make the system more fault-tolerant. Usually, this kind of task has two general goals: optimize the robot’s paths and perform the fusion of information acquired by the robots, building a shared map. It is important to note that collaborative exploration is more complex than exploration done by a single robot due to the fact that the information collected by the many robots may be conflicting.

In Nieto-Granda et al. [177], different strategies are used to achieve coordination of the swarm robots. These strategies are referred to as *Reserves*, *Divide and Conquer*, and *Buddy System*. Other approaches include the use of frontier-based exploration strategies to create a grid map of the unknown environment, as described in the work of Faigl and Kulich [178].

6.2.2. Foraging

An example of one of the most complex swarm task is *foraging*. In this task, the robots must find and retrieve some object in the environment. This is directly inspired by the behavior of food foraging in ant colonies and other social insects. It is known that such insects can efficiently explore the environment and retrieve the food from its source to the nest, using the shortest paths available in the environment. Foraging tasks are usually identified as the main test for cooperative robotics [31]. In fact, a complete foraging task can be described as a sequence of three sub-tasks: (i) First, the robots wander in the environment, avoiding each other and avoiding objects that are of no interest. (ii) These robots must also be able to differentiate the objects of interest encountered in the environment from those that are not of interest. (iii) When robots find an object of interest, it must carry it to a specific place, analogical to insects moving food to the nest. Different approaches to implement the foraging task in SR can be found in Russel et al. [179], Liemhetcharat et al. [180], and Pitonakova et al. [181].

An extensive description of the foraging task can be found in the work of Winfield [182], which also defines this task as a benchmark problem for single and multiple robot systems. The author shows that the foraging, of animals or robots, can be molded as a repetitive process of four steps: searching, grabbing, homing and depositing. However, the author also points out that, although the principles of robot foraging are well understood, engineering their emergent collective behavior remains a challenge.

6.2.3. Flocking

Coordinated motion, can be described by the movement of a cluster of individuals towards a same common direction, in an aggregated fashion, avoiding any obstacle in their way. In order to achieve this, the location of the individuals may follow a particular formation or be dynamically reorganized. Moreover, the direction of the group may be pointed out by a global target or emerge from the interaction between the swarm members.

In Balch and Arkin [183], the different approaches of coordinated motion are classified by the reference used by each robot to keep its movement: *unit-center-referenced*, *leader-referenced* and *neighbor-referenced*. In Navarro and Matía [184], this classification is extended, considering also a *multi-neighbor-reference*.

The first artificial flocking was the computer graphic animation implemented by Reynolds [185]. In this work, three rules were used to describes the movement of a flock of birds: collision avoidance, also known as separation rule; velocity matching, also known as alignment rule; and flock centering, also known as cohesion rule. In Turgut et al. [186], these three rules of flocking in coordinated motion was implemented in a flock of real robots. In that swarm, the robots can manage collision and velocity with the use of infrared proximity sensors, while alignment is achieved via a compass-like sensor.

The flock-like motion is simulated by a network of connected robots using explicit messages in the work of Erfianto and Trilaksono [187]. The authors analyze the impact of the ratio of sent messages via broadcast and the connectivity between the robots on the flocking motion. In Ref. [188], the flocking motion of robots is achieved considering their topological separation read by an omnidirectional camera, instead of position and orientations. Moeslinger et al. [189] describe a low-end and easy to implement flocking algorithm, which works without communication, memory or global information. They actually eliminate the need for communication of traditional flocking algorithms. They prove that the proposed algorithm achieves emergent flocking properties.

6.2.4. Collective transport

Many species of social insects, particularly ants, have the capacity to collectively transport objects from a given site to their nest [147]. Those objects may be several times the size and weight of a single ant, which makes this task impossible for an individual. Even so, the group of insects can perform this task in an efficient way. *Collective transport* is a task of great interest in SR, where a group of physically limited robots interact with each other to move a large object to a given destination. In Kube and Bonabeau [148], the first formalized model of the ant’s behavior to transport objects is described and implemented in a swarm of mobile robots. In this task, the robots achieved coordination without direct communication. Nouyanwe et al. [149] investigate the conditions under which homogeneous entities could evolve to higher-order ones in swarm-based system, hence giving rise to an entity hierarchy. They prove, via experiments using a real swarm of S-bots, that the notion of teamwork emerges over time. Even though the robots are physically the same and are equally controlled by the same software, a very interesting collective behavior regarding a self-organization into a dynamical hierarchy of teamwork emerges spontaneously from the swarm members without any external interference. This is done by implementing a collective transport task, which consists of retrieving the prey to the nest by the swarm members. The constraints imposed as the core of the robots control allow for the emergence of the division of labor into three categories of tasks: path formation; recruitment; and actual retrieval. In turn, recruitment is further divided into two sub-tasks: path maintenance and path following & grasp. Retrieval is also subdivided into path decomposition and group transport sub-tasks. In turn, the group transport sub-task is further refined for transporting tasks when the path is visible *versus* invisible. The idealized controlling algorithm, which is the same for all the robots of the swarm, consists of a collection of behaviors: navigation, self-assembly, and group transport. It is note worthy to stress that the three behaviors are further refined in sub-behaviors. These separate behaviors are implemented using either the motor schema paradigm, neural networks, or simple hand-written commands [149]. Moreover, the implemented and individually tested behaviors are integrated according to the behavior-based approach, which uses a finite state machine, wherein states represent the possible robot behaviors while the transition between states represents triggering possible events, such as elapsed time in a given behavior, distance to prey to be transported, object grasped, among other events. During operation, the appropriate behavior is triggered as the corresponding events occurred.

Distributed planar manipulation [190,191] is an instance of collective transport, where the robots enclose an object of known shape and move it along a predefined or calculated global trajectory. Each robot has its direction and force applied to the object, which is specified in terms of the position and orientation of the object. This dependency on global information is a drawback, because the robots may require too much processing time to estimate the location of the object as well as other robots. Other methods were designed avoiding the use of global knowledge. For instance, in Chen et al. [192], a transport strategy is based on the occlusion of the goal. Nonetheless, there is still global information, which is the position of the goal. It is obtained by the robots using omnidirectional cameras. A decentralized strategy is investigated by Rubenstein et al. [193]. Using a physics-based model, this strategy proved to be successful in transporting a complex object to its destination. In this work, the robots know the target direction, but have no information about the object weight and shape. Furthermore, the robot did not know its own position nor the number and positions of the other helping robots. Tuci et al. [194] have published recently a thorough survey on cooperative object transport as implemented in a multi-robot system. They portrayed a wide range of transport, coordination and control strategies that have been proposed over time. The authors also discuss several open challenges and possible directions to improve collective transport in swarm robotics.

6.3. Decision making

The *collective decision-making* is a problem where the individuals must make a choice, given a set of options. The choice may be influenced by the other robots of the swarm. These dynamics may converge to a global opinion or give rise to the emergence of groups of individuals that came up with the same decision. This kind of task is essential in most agent-based systems. In a SR context, they are necessary due to the distributed nature of SR and the absence of global shared information.

6.3.1. Consensus

The convergence to a common choice among different alternatives is called *consensus* or *agreement*. Many studies investigate this task in the case of SR systems. In Valentini et al. [195], the positive feedback modulation to achieve the majority of the best option is explored. In this approach, each robot iteratively sends its decision to neighboring robots for a period of time proportional to the option quality. The next decision is controlled by a probabilistic finite-state machine augmented by a majority rule. In subsequent work [196], the authors investigate the impact of spatial density of robots in the decision making process in a swarm of a hundred Kilobots. In Ref. [197], Valentini et al. integrate the dilemma of accuracy *versus* speed to reach an adequate decision within a swarm of 100 Kilobots.

6.3.2. Task allocation

Task allocation is a decision process, where the robots select, from a list of options, which task each one will execute. This problem is also known as *division of labor*. The partitioning of the swarm may be self-organized, wherein the number of robots that should perform each task is defined by a global configuration. This process may also be self-regulated, if the number of robots dedicated to each task is automatically selected. Many task allocation schemes are based on *threshold methods* [198], when a given observed number should not exceed a predefined threshold, or *probabilistic methods*, when a robot changes its task based on some computed probabilities [199].

In Mathias et al. [19], a dynamic task allocation algorithm based on consensus by threshold is proposed and implemented on a swarm of up to 48 Elisa-3 robots [200]. The proposed scheme was successful to achieve the predefined allocation proportion. Furthermore, in Nedjah et al. [201], another dynamic task allocation scheme for SR is proposed. It is based on an optimization process of a defined cost function, which is optimized using the PSO algorithm. In this scheme, each robot of the swarm materializes a particle of the PSO. The number of dimension of the search space coincides with the number of tasks. Hence, a particle position is a potential task allocation for the swarm. When the PSO converges, all robots have the same position, that is, the task allocation that should be used by the swarm. In this case too, the proposed algorithm was implemented and validated using the same swarm of up to 48 Elisa-3 robots. Through local and global interactions, this distributed approach was efficient enough to achieve a predefined proportion of robots performing different tasks.

Two tasks, which are related to the division of labor, are *robot clustering* and *robot recruitment*. For robot clustering, the swarm members of the same cluster should be moved closer to each other and farther from members of other clusters. A clustering technique for many classes is proposed by Cruz et al. [202]. In this work, there is no movement of robots to perform the separation of classes. Instead, the information of the classes circulates through the swarm, based only in local communication and with the use of virtual tokens. When the process finally converges, the swarm is spatially partitioned into the predefined number of clusters. The recruitment of robots is investigated in the work of Silva Junior and Nedjah [114,203]. This process starts with one robot, which discovers when investigating the environment, that a given task must be performed. To achieve this goal, the robot, identified as the initiator, directly recruits other robots in the neighborhood to help in the task execution. If more initiators are present, the entire swarm will be divided into clusters, each one specialized in a different task. The authors show that the recruitment can be correctly performed by a wave algorithm, based mainly on the propagation of information with feedback. Using the experimental results obtained in Ref. [204], Ferrante et al. could explain the origin of division of labor and complex social traits in nature as well as provide novel methodological and experimental tools to synthesize controllers to rule swarm robotic systems.

6.3.3. Localization

Many applications of SR require that a robot knows its position, which may be either an absolute or relative position. The *localization problem* consists of inferring the position of a set of robots or sensors when no external reference, such as *Global Positioning System*, is available. Many of the localization methods depend on the ability of a node to measure its distance to some reference nodes, also known as anchors, whose positions are known.

A method for localization that uses a cluster matching technique is presented by Rashid et al. [205]. First, an infrared sensor scans the environment, assessing the coordinates of the neighboring robots, obtaining a first network topology with absolute positions. Then, each robot measures the distances and the angles to each of its neighbors, obtaining a second network topology, with relative positions. Finally, both network topologies are merged so that absolute positions are computed for a number of robots that is equal or higher than that of the nodes included in the first network topology. This is so because the first scan may have missed robots that were out of the sensor's range.

In Sá et al. [66], a new multi-hop method is proposed. First, the distances of the robot to reference nodes are estimated. Then, the positions of each robot using distance measurements are obtained. Finally, the position of each node is refined using the positioning and distance information informed by neighboring nodes. The method includes a new technique to assess the confidence that should be granted to a contribution received from a neighboring node, and hence incorporating it into the localization computation accordingly. Furthermore, targeting the accuracy of the localization result rather than the efficiency of the computational process, a new localization method based on Min-Max and PSO is proposed in Ref. [116].

7. Challenges and SR directions

Besides the many advances in SR, there are still many challenges in this field. There are many factors slowing down the advancement of Swarm Robotics. Among these, we can list: the lack of non-expensive robots that can be manufactured in a very large scale; the lack of reliable yet versatile communications; and the lack of adequate general-purpose distributed control algorithms. However, the situation is quickly changing towards embracing SR as a solution to many complex problems whose solution is naturally distributed. It is noteworthy to emphasize that the hereby given solutions and research directions are somehow subjective and open to discussion. Nonetheless, it is actually the purpose here, which is to provoke, stimulate and recruit new researchers to follow daring objectives dear to the swarm robotics community.

In their review, Tan et al. [206] propose several fundamental problems to solve in the near future before swarm robotics can really be adopted in applications solving problems of everyday life. There should be standard cooperative schemes, may be inspired from the behavior of natural swarms, that can be implemented via the limited sensing and computing and data storage abilities of a desired swarm level behavior. There should be a way to describe the swarm robotics system in a pure mathematical model, which can predict the system behaviors at both individual and group level. We should work towards the consolidation of a new and general strategy that can take full advantage of the characteristics of any robotic swarm. Finally, we should work towards the design of a swarm robot of really low cost yet easy to control their cooperation as to yield intelligent collective behaviors. In the remaining of this section, a thorough discussion on these challenges is introduced. An insight on possible future directions of SR is also offered.

A variety of efficient works in SR were published in the last decades, as described in Sections 5 and 6. The general impression that arises from this review is that the given solutions are *ad hoc*, *i.e.* the solution only addresses a single or just some tasks related to the management of robotic swarms. At some point, this is interesting because it allows researchers to explore all kinds of possibilities. Nonetheless, this has also an adverse effect: many new techniques are proposed, but some successful ones are not well exploited in real-world applications. Of course, there are few exceptions, such as in the case of the Swarmanoid Project [71], which proved that cooperation between heterogeneous robots is possible to accomplish a daily life task, the subCULTron [207], which demonstrated that swarm robotics can be a solution for environmental intelligent monitoring, the Swarm-Bot project [69], which showed many evidences that swarms can efficiently self-assemble into artifacts [70] and self-organize into well-divided teamwork [149] and the swarm of particle robots [208], which applies successfully statistical physics to engineer a very unusual yet effective robot swarm. Note that the last three projects will be explained in Section 7.2, which is deals with non-wheeled SR platforms and related challenges. However, in an a broader view, the absence of a clear and straight forward direction in SR research results is a huge obstacle facing the development of real-world applications based on the swarm robotic approach. One can wonder why there is still no attempts to transform these SR successful experiences into real-world applications, such as waste recycling, earthquake victims rescue, smart grids maintenance, among many other applications where SR would be a perfect solution.

7.1. Challenges for future platforms

In this rest of this section, we discuss some of the research challenges of the swarm robotics field. We divide this discussion regarding three aspects: the standardization and dedication of SR platforms as well as some challenges facing swarms of non-wheeled robots. Last but not least, we sketch some technical challenges of the field in view of its application as a viable solution of real-world and daily-life problems.

7.1.1. Standard SR platforms

The lack of common parameters can be found at different levels, starting from the robotic platforms. The robot models, shown in Section 3, exhibit many differences when they are compared. Practically speaking, each model has it own set of sensors, programming language, communication interfaces, and so on. This makes it difficult, if not impossible, to migrate a project from a platform to another: a task successfully implemented in a given model may be hard, or even impossible, to be adapted to another model, due to hardware and/or software incompatibilities. Furthermore, the exploitation of *heterogeneous swarms* is also hindered as each model takes advantage of a different communication infrastructure. A possible initiative to overcome these limitations is the development of a unified set of *SR standards* for future platforms. These standards could define minimal yet common resources for several SR aspects, including:

1. *Processing hardware*: Any platform must share, at least, a minimal specification of computational resources, such as processing frequency and memory size. Due to size, weight and power consumption, many platforms use very limited processors or a very small program memory, which restricts the implementation of controllers of more complex tasks. However, some of the current platforms, as R-One and Wolfbot, are based in modern ARM[5] architectures, indicating not only the need of more computational power, but also the choice of using well-known processors.
2. *Communication*: The same infrastructure must be employed in different platforms, attending minimal requirements of bandwidth and throughput. Many platforms use WIFI for communication with remote stations, but there is a lack of reliable, message-based, inter-robot communication infrastructures. Wolfbot is an example of robot that uses a dedicated *zigbee*[6] device for inter-robot communication. This infrastructure enables the robots to communicate with each other, an essential feature in tasks that require intense information sharing.
3. *Sensors*: Robots must have a minimal sensory setup. IR is present in most of robots, but other sensors, such as microphones and cameras are becoming increasingly popular. While IR and ambient light are useful in getting measures about the environment, cameras have been well used in human-robot interaction.

[5] http://www.arm.com.
[6] http://www.zigbee.org.

4. *Localization System*: A useful feature is the position awareness of the robots, knowing where they are in the environment. Localization can be achieved in different ways, ranging from odometry to triangulation techniques. These require different devices, such motor encoders (for odometry), compass sensors (for robot heading) and a Global Positioning System, GPS (for external operations in large terrains).
5. *Tasks*: Perhaps the most important feature, which summarizes the previous ones, is the definition of a minimal set of tasks that any platform must be able to perform. The features described above should be the minimal resources that allow the execution of a set of well-known basic tasks, including aggregation, navigation, consensus, and so on. The definition of a set of simple yet *robot-independent* tasks could be the ground for building more complex ones.

Note that the overall idea behind the adoption of the above steps is to make it easy to use swarm robotics to solve daily life problems. It is notable that unless we design swarm robots that are easy to program and versatile enough so that they can be re-utilized in several applications, this is going to be extremely difficult. For instance, some can advocate that natural swarms do not perform a minimal set of tasks, so why should a robot do. Actually, we do not know for a fact whether natural swarm members have a common set of tasks. However, what we are pointing out here is that if we could have swarm robots with a basic set of tasks that can be composed to achieve complex behavior, the work of the engineers that program the robots would be much easier. Of course, may be there are other ways to facilitate the spread of usage of swarm robotics in solving daily-life problems.

The current simulation tools can be very helpful in the validation of SR standards. Software, such as Webots [209] and V-Rep [100], allow not only the use of simulated robots,[7] but also they provide tools to virtually change their features, adding and/or removing devices *i.e.* sensors and actuators. On the other hand, each simulator following the SR standard must have at least a single robot model, as the basic model, which provides the requirements thus described. This simple simulated robot, compatible with any other model following a future standard, could be extended for advanced studies by an adaptation process, including and/or removing characteristics.

Perhaps it is still too early for unification in SR studies, but it is a process that has occurred in different fields: computational architectures, programming languages, communication protocols, and so forth. In fact, part of this idea is addressed by the *Robot Operating System* (ROS),[8] a framework for programming and reuse of robots software that is compatible with other simulation tools, such as Gazebo and V-Rep. At the core of the ROS framework is a runtime platform, based on which APIs, including abstract data structures, such as Swarm, Neighbor and Virtual Stigmergy, are provided to the user. Furthermore, a library of typical swarm algorithms to further facilitate application development is being engineered. Swarming with ROS allow the implementation of decentralized system based on simple local interactions [210]. This initiative is very extensive, covering all kinds of robots, including complex industrial complex machines. A subset of ROS features, focusing on mobile and distributed sensing and actuating, could be used as a starting point for the general specification of SRs. In this same direction, there is also OpenSwarm[9]: an embedded operating system for miniature robots. It is specifically designed to be is an easy-to-use event-driven preemptive operating system for swarm robots. It offers abstract hardware-independent functions to make user code more extendible, maintainable and portable. This kind of initiative to have platform-independent operating systems of swarm robots helps steering towards the standardization avenue.

[7] Some simulators also allow the remote access to real world robots, working as an extended controller.

[8] http://www.ros.org.

[9] http://openswarm.org.

It is noteworthy to emphasize that the platform standardization could boost a great deal the field of automatic design for the control software of swarm robotic systems. Needless to point out that automatic software design for swarm robotic systems has already accomplished some achievements. Nonetheless, many issues still need to be addressed by the SR community. Francesca and Birattari [93] survey such achievements and challenges of automatic design for robot swarms.

Unification in SR specifications can also help in the development of the emerging of new kinds of behaviors. In many works, as presented in this review, the controller design follows a microscopic modeling. This methodology may eventually result in many different controllers, where each one specialized in just a few behaviors. In this context, soft computing techniques, such as Evolutionary Computation [211] and Particle Swarm Optimization [212], are used to adjust the parameters of robot controllers based on neural networks or finite state machines. Those approaches are usually robot-centered. Many other works, enumerated in this review, follow a similar strategy: a specific intelligent technique is used to perform a single task or multiple tasks in a specific real or simulated robot. However, this strategy is not general enough in terms of tasks and platforms. Most of the efforts to develop a general description of SR come from the study of macroscopic models. For instance, in the work of Winfield et al. [213], a formal specification of SR behaviors based on temporal logic is proposed. Besides the high level of abstraction, the authors argue that this mathematical scheme would be valid in real-world problems.

7.1.2. Dedicated platforms

The standardization of future architectures can indubitably facilitate the learning and implementation of new behaviors in SR, in a similar way to the present use of simulators. This may represent an intermediary prototyping step of the entire SR project. A hypothetical platform that is generic and broad may eventually be expensive, having more computational resources than it is really necessary, to perform a specific task. To achieve a more optimized swarm in terms of hardware, the researcher must implement not only the software controller, but also design the physical robot altogether. As discussed by Bezzo et al. [214], two philosophies coexist in the robotics design communities. First, there are researchers with focus on expanding the robots capabilities and making them of more general purpose. Second, there are others that develop design systems for the design of user-dedicated robot models.

The design of robotic systems can be divided into three components: body, hardware and software. *Body design* is the conception and physical implementation of the robot structure. *Hardware design* is the specification of printed circuit boards (PCB) and components, such as processors, sensors and actuators. The *software design* is the development of the algorithms executed by the robots to finally achieve the expected behavior and thus the desirable results. Most of the works cited in this paper are focused only in the software component, the implementation of algorithms and their verification via simulators and/or commercial robots. Considering the manufacturing capabilities is also a different, feasible yet expensive approach.

The popularization of three-dimensional (3D) printing allow manufacturing rigid parts of any arbitrary shape [215]. Regarding electronic circuits design, two choices are generally considered: the adoption of a modular and reusable platform, or the design of dedicated PCBs. For example, Yu et al. [216] designed a swarm of simple mobile robots (microMVP) based on Sparkfun[10] prototype board with a 3D printed body. Pan et al. [217] proposed an underwater robot with a spherical body, also implemented using 3D printing technology. In their work, the authors used a *system-on-chip* (SoC) to embedded most of the circuits into a single *field programmable gate array* (FPGA). This is an interesting approach, because the robot has a very specific shape (a sphere), but

[10] www.sparkfun.com.

the electronic controller is based on a reconfigurable device (an FPGA). In both cases, software design is posterior to the physical implementation. The dedicated robot is first made using a 3D printed body, PCBs, actuators and sensors, and so forth. Only then the software is specified. It is noteworthy to emphasize that the project of the swarm is thus made on-the-fly for and dedicated to the project being designed. So there are no third-party restrictions.

Bezzo et al. [214] have a different philosophy for the future fabrication of dedicated robots, using integration and cogeneration of mechanical, electrical and software subsystems. Some approaches were developed in this direction. The ROSLab [218] is an example of modular programming environment for robotic applications based on a library of predefined features (ROS nodes). The user selects some features, such as sensors and actuators from a predefined library, defines the interconnection for these components, and the system automatically generates a skeleton code. In a posterior work [219], ROSLab was expanded to also generate the design of mechanical components. The same developers of ROSLab also introduced the EMLab [220], which generates the dedicated PCB for the desired robotic model. Of course, the current approaches are focused on single robots. However, future frameworks may introduce the SR concepts such as cooperation and collaboration. Furthermore, the use of EC together with other swarm intelligent techniques may be able to generate robots that are highly optimized for real-world SR applications.

7.2. Non-wheeled platforms

Most of the works described in this review are based on wheeled mobile-robots, or robots that are able to move over a smooth surface. We judge that there is no room in this review to cover properly this facet of swarm robotics, which in itself requires another dedicated work survey. In the design of some real-world applications, it may be needed to overcome this planar restriction to expand the mobility capabilities of the swarm. Future research works on SR may be directed by the study of new types of robotic devices, which will hopefully use the current knowledge developed for wheeled robots. The size of the robots is one of the promising research directions for robot manufacturing. Not only considering the development of new light yet more resistant materials, but also to consider the utilization of silicon integration, more processing capabilities, smaller yet smarter sensors, and so on. In fact, the miniaturization is a characteristic noted since the start of the MRS related studies.

A probable direction of SR research is in the field of flying or aerial robots as well as underwater robots. In both cases, such robots may inherit the well-grounded knowledge acquired from wheeled robots that move only on flat surfaces, to perform tasks in three-dimensional world. Nowadays, many works are focused in the design of *micro-aerial vehicles*, such as micro-helicopters [221], quadcopters [222] and winged robots [223]. Although there are many studies on these devices, only a small deal of those are approach the case of multiple robots or a swarm notion. For instance, in Jifeng et al. [224], a guidance system for multi Unmanned Aerial Vehicles (UAVs) is proposed. In their work, the components and functions of a simulated system for cooperative guidance is discussed. Moreover, a six-degree of liberty model of mission planning simulation system together with a cooperative guidance models and a fusion model of formation information are analyzed. Also, a situational awareness model, a data-link model and an airborne data link model are exploited. A more audacious application of SR is in *spatial exploration*. An effort to this goal can be seen in Truszkowski et al. [225] and Vassev et al. [226]. The *Autonomous Nano Technology Swarm* (ANTS) concept is an approach to asteroid belt resource exploration. By its virtue, ANTS provides extremely high autonomy, minimal communication requirements with Earth, and a set of very small explorers. These explorers, forming the swarm, are small, low-power, and low-weight spacecraft, capable of operating as fully autonomous and adaptable agents for years in space. These small robots are able to interact with each other, thus making self-organization possible based on the emergent behavior of their simple interactions.

Regarding underwater swarm robots, we mention two main projects with undergoing research, which are CoCoRo[11] and subCULTron.[12] The CoCoRo project explores and develops collective cognition in autonomous underwater robots. In this project, the researchers created robot swarms that are capable of collective cognition [227]. They work as a collective system of autonomous agents that can learn from past experience and their environment. The robot swarm mimics the behavior of fish schooling. Two kind of underwater autonomous robots were idealized: Jeff and Lily robots. As a proof of concept, twenty Jeff robots are put to float in a tank of water. The robots mission consists in finding debris, originating from a sunken object. Lily robots search just below the surface while Jeff robots search at the bottom of the pool. Magnets are placed around the target to mimic an electro-magnetic signal, emitted locally. The robots used their built-in compasses to locate the target. When a Jeff robot discovers the target it recruits other Jeff robots, which then gather around the found object, while Lily robots collected overhead. During field experiment, the robots are exposed to actual waves, currents and corrosive salt water. Despite the difficult conditions, the robot swarms were able to remain clustered around their base station as well as go on patrols and successfully return to base.

The subCULTron project aims at achieving long-term autonomy of an underwater swarm robot that can learn, self-regulate and self-sustain in a high-impact application area [207]. The heterogeneous system consists of 3 different types of robots: (i) on the sea-ground, artificial mussels, called aMussel, plays the role of a collective long-term memory of the swarm. They allow for information to stay beyond the runtime of other robots, thus continuing to learn from previous experiences. These mussels monitor the natural habitat, including biological agents like algae, bacterial incrustation and fish; (ii) on the water surface, artificial lily, called aPads, allows interfacing with the human society, delivering energy and information influx from ship traffic or satellite data; (iii) between those two layers of aMussels and aPads, there is the swarm of artificial fish, called aFish. They move, monitor and explore the environment and exchange info with the swarm of aMussels and aPads.

Another research direction that can drive non-wheeled platforms is the statistical physics inspiration. In the past three decades or so, biological systems characterized by a large number of interacting units have been studied using a statistical physics approach, applying it to study collective behaviors in swarm robotics, such as collective motion [228,229] and collective decision-making [230,231], among others. Statistical physics studies systems, wherein the large number of particles permits a successful application of a probabilistic approach to infer some rules about the collective behavior of the swarm of particles. In the statistical physics-based approach, correlation and fluctuation functions about the particle's behaviors play a central role in the description and control of the of collective phenomena [228,232]. When applied to swarms, it exploits the claimed strong correlation between the swarm members to implement the expected collective behavior [208,233]. Li et al. [208] apply the statistical mechanisms to control the overall behavior of a robotic system, wherein the robots themselves are a swarm of so-called loosely coupled particles. These particles are incapable of independent locomotion and do not possess any individual identity or addressable position. Despite the stochastic motion of the robot and lack of direct control of its forming particles, the authors demonstrate via a swarm of real particle robots, wherein a robot is composed of up to 24 particles, some collective behavior such as collective motion with obstacle avoidance, cooperative transport of a big object, and collective decision-making to determine the intensity of the light as well as the its phase.[13] Moreover, but this time via simulation

[11] Website of the project: http://zool33.uni-graz.at/artlife/cocoro.

[12] Website of the project: http://www.subcultron.eu/.

[13] A video illustrating the collective behaviors: https://bit.ly/2VTMy2F.

using particle robots with up to a 100,000 particles, they demonstrate that the robotic swarm is capable of robust collective behavior such as locomotion and phototaxis, which is a movement towards a light source.

7.3. Other challenges

Further to the SR platforms challenges, there are still many other technical challenges to make swarm robotic systems a robust and consolidated solution to real-world applications. These challenges have been assembled by experts in the robotic field in general and in SR in particular [234,235].

First of all, there is the energy and power bottleneck. Energy storage is a major a huge problem for mobile robotics. Even though the battery technology is trying to follow the diving progress, it is still a challenge for robots that need to be working (powered on) for decades. For this purpose, recharging systems such as automated battery-swapping stations have been implemented. These systems require that the robots interrupt their activity. Even if this interruption is for a short time, it usually influences the swarm overall behavior. In order to remedy to this situation, Arvin et al. [236] introduce a low-cost on-the-fly wireless charging system, composed of several charging cells. A prototype system with 12 charging cells and a small mobile robot called Mona has been developed. Moreover, renewable energy need to be considered for robots to overcome or at least mitigate shortages for this kind of very long robotic missions, and enabling the robot to be able to harvest and store energy for future use. In parallel to that there is an urgent need to minimize power consumption, which is only possible when the robot is well tailored to the specific mission to pursue.

Secondly, there is the fabrication material bottleneck. It is believed that, in the near future, robots need to be made of materials that have some functionality embedded therein, such as artificial smart material and smart muscles. Such robots would behave like a living systems exhibiting some characteristics such as embedded sensing and actuating operations [237–239] and even self-regeneration [240]. Functional materials allowing sensing, movement, energy harvesting and storage allow the emerging of tiny (nano) yet efficient robots [240]. There has been some progress in manufacturing artificial muscles, but their robustness, efficiency and energy/power density need to be improved. Moreover, embedding living cells into robots can overcome challenges of powering small robots, as well self-healing and sensing [240].

Thirdly, there is a fault-tolerance bottleneck. Swarm robotic system can come handy in exploring places where humans cannot or do not want go, such as the deep sea, space or disaster zones, among others, often highly disordered and hostile environments. Nowadays, there are many application of collective exploration performed by a swarm robotic system. However, all existing implementations are not fault-tolerant. So a major challenge in this sense is to engineer robot swarms that have the ability to adapt to new scenarios, by learning from them. This learning process should allow the swarm to recover from failures and even make new discoveries. In this case, the robot should be able to self-reconfigure their embedded hardware and software to adapt to the new scenarios. Example of such cognitive robot swarms are those used in the CoCoRo [227] and subCULTron projects [207].

Last but not least, a major challenge ahead the robotics community is related to the integration of control and decision making. It is important to pay attention to applications, wherein simultaneous decision making and control of the swarm robots, which are executing a cooperative task, take place. Such an integrated approach takes care simultaneously of the kinematics based representations of the overall scenario as well as the dynamics of the individual robot [241].

8. Discussion on SR research interests

In order to investigate the general interest on topics related to SR, the most commonly used web-based search engines were used [242].

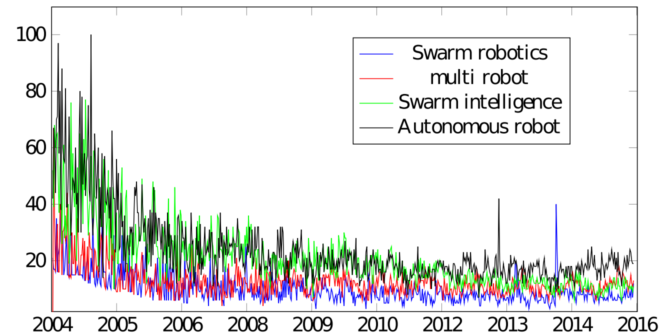

Fig. 5. Interest in terms of number of hits over the time on topics related to Swarm Robotics.

These provide the number of Internet searches of a specific term per month since the year of 2004. It also allows the user to compare the volume of searches between two or more terms. We compared the evolution of the interest on 4 topics: *swarm robotics*, *multi robot*, *swarm intelligence* and *autonomous robots*. It is noteworthy to point out that these are the keywords that we could trace as topics available in *Google trends* that are related to swarm robotics. Even though we are convinced that this purely quantitative analysis has a limited use only, it allows us to establish a kind of comparison of researcher's interests and research trends in the field of swarm robotics. Fig. 5 shows the evolution of the interest on these four topics. Some general comments can be made from the analysis present in Fig. 5:

- The interest on *swarm intelligence* and *autonomous robots* were high in the middle of year 2000, in comparison with *swarm robotics* and *multi robots*. These two topics are broad, and include many other sub-topics. On other hand, swarm robotics and multi robots are very narrow when viewed as fields of study. In fact, these are sub-topics of swarm intelligence and robotics [128].
- The interest on swarm intelligence and autonomous robots decreased, and became nearly steady by year 2010. This behavior can be interpreted as the maturation of these two topics. First, a great number of studies were performed in these fields, explaining most of their characteristics. Now, these well-known themes are part of other research fields, but not the main focus. For example, nature-inspired meta-heuristics are recurrently used as optimization processes in engineering problems, but are not necessarily the main contribution of such research works.
- The terms *swarm robotics* and *multi robots* show a steady number of searches since recording began. Also, the interest on multi robots is more intense thanks to *swarm robotics*. It can be explained by the difficulty in differentiating these two terms. Any group of mobile robots can be described as a multi-robot system, but not as a swarm. The characteristics of swarm, such as coordination and cooperation, may turn this field of study too specific and abstract. As a result, most of the works including real-world implementation are considered works on multi-robot systems. Even so, the number of searches on these topics did not decrease, indicating a constant interest in both swarm and multi-robot systems.

The search for the term *swarm robotics* in the Google Scholar[14] returns 4,950 entries for the same period between 2004 and 2016. It is a low number of academic publications, in comparison to 21,400 entries for *multi robot*, 38,200 for *swarm intelligence* and 17,700 for *autonomous robot*. This is a hint that the field of SR is yet to see better days with more interest by the scientific community.

The main question that arises from this analysis of interest is: why swarm intelligence receives much more attention than SR, when both

[14] http://scholar.google.com/.

had their origins in early nineties? A plausible answer could be simply that swarm intelligence algorithms have easier implementations than SR systems. Swarm intelligence techniques are idealized to perform optimizations and find optimal solutions of complex cost functions. This numerical characteristic makes such algorithms applicable to a variety of engineering problems, relying mainly on computational power. The search for increased performance for such techniques attracted a lot of attention in the last decades. Swarm algorithms, as well as evolutionary computing, are well-grounded tools to resolve different kinds of computational problems. The decrease of interest on swarm intelligence over time shown in Fig. 5 have two main explanations. Perhaps there are already too many techniques that can perform optimization efficiently, or maybe inventing new algorithms that would outperform the existing ones is a very hard task. Furthermore, following some existing definitions, swarm robotics is a sub-area of swarm intelligence. For instance, Dorigo et al. in Ref. [25] state that the design of robot swarms is guided by swarm intelligence principles. Also, in Wikipedia [243], one can find a statement like "The Swarm Robotics approach emerged from the field of artificial swarm intelligence". Furthermore, Brambilla et al. in Ref. [24] write about "robotics systems that exhibit swarm intelligence features". This may be one of the reasons why swarm intelligence is a more diffused as terminology and as discipline. On the other hand, SR research follows a different process. It usually requires an explicit group of agents to execute tasks, composed by real or simulated robots. Large experiments may be prohibitively expensive, when considering the resources needed to acquire a swarm of hundreds of physical robots. However, we strongly believe that this problem could be mitigated if all robots available in the research laboratory can be used independently of their distinct models. In turn, this can only be possible when swarm robots have a standard set of sensors, actuators and processors. Note that experiments with a large simulated swarm are computationally expensive, and may be even impractical due to simulator limitations.

Even though SR may have still not achieved the level of maturity already asserted for swarm intelligence, there are a lot of research works in robot design, behavior modeling and tasks execution among others. Nonetheless, these works implemented in SR are still not for final real-world applications. This "lack of closure" is probably the most significant problem in present SR research works. When the SR stops to be yet another proof of concept and starts to become a "final product", then this field may receive much more attention, something like that experienced by swarm intelligence during the nineteen nineties. This said however, we feel that projects like the Robotorium [244], ARK [106] and Kilogrid [105] can help a great deal in advancing the state-of-the-art of the applicability of Swarm Robotics in our daily life tasks.

9. Conclusions

Swarm Robotics is a multi-disciplinary field of study, which has as main inspiration the interaction among members of physical and biological systems, in general and of animal and insect societies, in particular. Redundancy of robots enables the realization of tasks that are hard or impossible to be done by a single robot, while making the entire system more robust. New technologies yield new designs of more reliable robot models, in comparison with old models used until the nineteen nineties. Such advances include efficient power consumption, low noise sensors, increased range of communication, robot miniaturization and the adoption of recent programming languages in the development environments.

This paper reviews the many directions of SR research. It overviews research on physical robots *versus* development of simulation environments. It also revises and discusses works on the design of behaviors and collective behavior tasks, many of them using soft computing techniques.

Although some related areas, such as swarm intelligence and wireless sensor networks, are already mature, there is still a gap between the academic research on SR and real-world applications. Even with recent advances, SR is still a hard field of study. One significant obstacle is the lack of portability, that is, a behavior designed for a specific robot must be completely redesigned to work in a different model. The huge amount of time wasted to adapt the same task to different models is regrettable, and the useful know-how ends up being restrained to abstract models of swarms and tasks. Particular aspects of a given robot are too specific to be inherited by other models, except maybe in the case of new versions of the same robot, or robots made by the same manufacturer. Perhaps the adoption common hardware and software platforms to control the robots would help in the emergence of universal standards for SR. A different point of view is the print-on-demand robots, with intelligent frameworks designing mechanical, electric and software aspects of entire swarm. Actual 3D printing technologies makes this fabrication concept possible, although still at an early stage. Standardizing SR features, either in general-purpose platforms, or in 3D printed dedicated robots, may open a new perspective in SR research in a near future.

Nowadays, swarms of Unmanned Aerial Vehicle, drones and underwater robots and related applications are becoming more and more common and popular in swarm robotics. This kind of swarms and related applications, approaches and methodologies need a dedicated survey. There are also swarms formed of micro/nano robots that deserves another dedicated survey of related applications, methodologies and used approaches. We intend in the near future to invest time and effort to review research works versed in these kinds of robotics swarms.

Acknowledgment

We are very much grateful to the editor and to all the reviewers for their constructive criticisms, advices and suggestions, helping us in further improving the content of the present review. We are also grateful to FAPERJ (*Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro* www.faperj.br) and CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico* www.cnpq.br) and CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior* www.capes.gov.br) for their continuous financial support.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.swevo.2019.100565.

References

[1] Didier Guzzoni, Cheyer Adam, Luc Julia, Konolige Kurt, Many robots make short work: report of the SRI international mobile robot team, AI Mag. 18 (1) (1997) 55–64.
[2] Gerardo Beni, The concept of cellular robotic system, in: IEEE International Symposium on Intelligent Control, 1988. Proceedings, IEEE, 1988, pp. 57–62.
[3] Gerardo Beni, From swarm intelligence to swarm robotics, in: Swarm Robotics, Springer, 2005, pp. 1–9.
[4] Jean-Louis Deneubourg, Theraulaz Guy, Ralph Beckers, P. Bourgine, E. Varela, Swarm made architectures, in: 1st European Conference on Artificial Life, MIT Press, 1992, pp. 123–133.
[5] Theraulaz Guy, Jean-Louis Deneubourg, in: R.A. Gardner, A.B. Chiarelli, B.T. Gardner, F.X. Ploojd (Eds.), Swarm Intelligence in Social Insects and the Emergence of Cultural Swarm Patterns. *The Ethological Roots of Culture*, Kluwer Academic Publishers, Dordrecht, 1994, pp. 1–19.
[6] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B Cybern. 26 (1) (1996) 29–41.
[7] James Kennedy, Russ Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, vol. 4, Nov 1995, pp. 1942–1948. vol. 4.
[8] Erol Şahin, Swarm robotics: from sources of inspiration to domains of application, in: Swarm Robotics, Springer, 2004, pp. 10–20.
[9] Alan Winfield, Julien Nembrini, Safety in numbers: fault-tolerance in robot swarms, Int. J. Model. Identif. Contr. 1 (1) (2006) 30–37.

[10] Jan Dyre Bjerknes, Alan Winfield, On fault tolerance and scalability of swarm robotic systems, in: Distributed Autonomous Robotic Systems, Springer, 2013, pp. 431–444.
[11] Anders Christensen, Rehan OGrady, Marco Dorigo, From fireflies to fault-tolerant swarms of robots, IEEE Trans. Evol. Comput. 13 (4) (2009) 754–766.
[12] Colin Angle, Robotics Business Review: 20% of All Vacuums Are Robots, 2016. [Online accessed 8-July-2019].
[13] Michael Rubenstein, Alejandro Cornejo, Radhika Nagpal, Programmable self-assembly in a thousand-robot swarm, Science 345 (6198) (2014) 795–799.
[14] Justin Werfel, Kirstin Petersen, Radhika Nagpal, Designing collective behavior in a termite-inspired robot construction team, Science 343 (6172) (2014) 754–758.
[15] Fredericka Berger, Robots: what They Are, what They Do, Greenwillow, 1992.
[16] Toshio Fukuda, Soft Computing for Intelligent Robotic Systems, vol. 21, Springer, 2013.
[17] Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, Agoston Eiben, Evolutionary robotics: what, why, and where to, Front. Robot. AI 2 (4) (2015).
[18] Thomas Schmickl, Heiko Hamann, BEECLUST: a swarm algorithm derived from honeybees, in: Xiao Yang (Ed.), Bio-inspired Computing and Communication Networks, CRC Press, Boca Raton, FL, USA, 2011, pp. 96–135.
[19] Rafael Mendonça, Nadia Nedjah, Luiza Mourelle, Efficient distributed algorithm of dynamic task assignment for swarm robotics, Neurocomputing 172 (2016) 345–355.
[20] C. Ronald, Arkin. *Behavior-Based Robotics*, The MIT Press, Cambridge, Massachusets, USA, 2000.
[21] Rodney A. Brooks, Elephants Don't Play Chess. Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, The MIT Press, Cambridge, Massachusets, USA, 1990.
[22] Rodney Brooks, A robust layered control system for a mobile robot, IEEE J. Robot. Autom. 2 (1986) 14–23.
[23] C. Ronald, Arkin. Motor schema — based mobile robot navigation, Int. J. Robot. Res. 8 (4) (1989) 92–112.
[24] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, Marco Dorigo, Swarm robotics: a review from the swarm engineering perspective, Swarm Intell. 7 (1) (2013) 1–41.
[25] Marco Dorigo, Mauro Birattari, Manuele Brambilla, Swarm robotics, Scholarpedia 9 (1) (2014) 1463.
[26] L. Garattoni, M. Birattari, Swarm robotics, in: J.G. Webster (Ed.), Wiley Encyclopedia of Electrical and Electronics Engineering, John Wiley & Sons, Hoboken, NJ, 2016.
[27] H. Hamann, T. Schmickl, Modelling the swarm: analysing biological and engineered swarm system, Math. Comput. Model. Dyn. Syst. 18 (1) (2012) 1–12.
[28] V. Trianni, A. Campo, Fundamental collective behaviours in swarm robotics, in: J. Kacprzyk, W. Pedrycz (Eds.), Handbook of Computational Intelligence, Springer, 2015, pp. 1377–1394.
[29] N. Correll, H. Hamann, Probabilistic modeling of swarming systems, in: J. Kacprzyk, W. Pedrycz (Eds.), Handbook of Computational Intelligence, Springer, 2015, pp. 1423–1431.
[30] Heiko Hamann, Swarm Robotics: A Formal Approach, Springer International Publishing, Cham, Switzerland, 2018.
[31] Y Uny Cao, Alex Fukunaga, Andrew Kahng, Cooperative mobile robotics: antecedents and directions, Aut. Robots 4 (1) (1997) 7–27.
[32] Jonathan Loo, Jaime Mauri, Jesús Ortiz, Mobile Ad Hoc Networks: Current Status and Future Trends, CRC Press, 2016.
[33] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, Aamir Hassan, Vehicular ad hoc networks (VANETS): status, results, and challenges, Telecommun. Syst. 50 (4) (2012) 217–241.
[34] Gianluca Aloi, Marco Di Felice, Valeria Loscri, Pasquale Pace, Giuseppe Ruggeri, Spontaneous smartphone networks as a user-centric solution for the future Internet, IEEE Commun. Mag. 52 (12) (2014) 26–33.
[35] M. Corson, J. Macker, G. Cirincione, Internet-based mobile ad hoc networking, IEEE Int. Computi. 3 (4) (1999) 63–70.
[36] Jack Burbank, Philip Chimento, Brian Haberman, William Kasch, Key challenges of military tactical networking and the elusive promise of MANET technology, IEEE Commun. Mag. 44 (11) (2006) 39–45.
[37] Shuang-Hua Yang, Wireless Sensor Networks: Principles, Design and Applications, Springer, 2014.
[38] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, Wireless sensor network survey, Comput. Netw. 52 (12) (2008) 2292–2330.
[39] Andrew Howard, Maja Matarić, Gaurav Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: Hajime Asama, Tamio Arai, Toshio Fukuda, Tsutomu Hasegawa (Eds.), Distributed Autonomous Robotic Systems, vol. 5, Springer, Japan, 2002, pp. 299–308.
[40] Milica Đurišić, Zhilbert Tafa, Goran Dimić, Veljko Milutinović, A survey of military applications of wireless sensor networks, in: 2012 Mediterranean Conference on Embedded Computing, IEEE, 2012, pp. 196–199.
[41] Dan Chen, Zhixin Liu, Lizhe Wang, Minggang Dou, Jingying Chen, Hui Li, Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems, Mob. Netw. Appl. 18 (5) (2013) 651–663.
[42] Chol-U. Jee, Jung-Hoon Lee, Ryum-Duck Oh, System management of human object on the industrial safety, Int. J. Smart Home 10 (5) (2016) 171–182.
[43] V. Trianni A. Reina, Deployment and redeployment of wireless sensor networks: a swarm robotics perspective, in: N. Mitton, D. Simplot-Ryl (Eds.), Wireless Sensor and Robot Networks - from Topology Control to Communication Aspects, World Scientific, 2014, pp. 143–162.
[44] Othon Michail, G. Paul, Spirakis. Elements of the theory of dynamic networks, Commun. ACM 61 (2018) 72.
[45] Faruk Bagci, Ala' Khalifeh, Cluster communication protocol for wireless sensor networks, Int. J. Sens. Netw. 20 (2) (2016) 84–91.
[46] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, Gregory Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Commun. 7 (5) (2000) 16–27.
[47] Baoxian Zhang, Zhenzhen Jiao, Li Cheng, Yao Zheng, Athanasios Vasilakos, Efficient location-based topology control algorithms for wireless ad hoc and sensor networks, Wirel. Commun. Mob. Comput. 16 (14) (2016) 1943–1955 Wiley Online Library.
[48] Harsha Chenji, Stoleru Radu, Toward accurate mobile sensor network localization in noisy environments, IEEE Trans. Mobile Comput. 12 (6) (2013) 1094–1106.
[49] Eric Bonabeau, Marco Dorigo, Theraulaz Guy, Swarm Intelligence: from Natural to Artificial Systems, Oxford University Press, Inc., New York, NY, USA, 1999.
[50] A. Eiben, J. Smith, Introduction to Evolutionary Computing, SpringerVerlag, 2003.
[51] Dan Simon, Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence, John Wiley & Sons, Hoboken, New Jersey, 2013.
[52] John Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Univ. Michigan Press, 1975.
[53] Maurice Clerc, TRIBES — un exemple dŠoptimisation par essaim particulaire sans parametres de contrôle, in: Optimisation Par Essaim Particulaire (OEP 2003), Paris, France, 2003, p. 64.
[54] Dervis Karaboga, Bahriye Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459–471.
[55] Szymon Łukasik, Sławomir Żak, Firefly algorithm for continuous constrained optimization tasks, in: International Conference on Computational Collective Intelligence, Springer, 2009, pp. 97–106.
[56] Xin-She Yang, Suash Deb, Cuckoo search via lévy flights, in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE, 2009, pp. 210–214.
[57] Pinar Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.
[58] Nicole El Zoghby, Valeria Loscri, Enrico Natalizio, Véronique Cherfaoui, Robot cooperation and swarm intelligence, in: Wireless Sensor and Robot Networks: from Topology Control to Communication Aspects, World Scientific Publishing Company, 2014, pp. 168–201.
[59] Owen E. Holland, G.W. Walter, The pioneer of real artificial life, in: Christoper Langton (Ed.), Proceedings of the 5th International Workshop on Artificial Life, MIT Press, Cambridge, 1997, pp. 34–44.
[60] 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, MIT Press, Cambridge, MA, USA, 1990, pp. 356–363.
[61] Luiza de Macedo Mourelle, Rogério de Moraes Calazan, Nadia Nedjah, Particle, dimension and cooperation-oriented pso parallelization strategies for efficient high-dimension problem optimizations on graphics processing units, Comput. J. 59 (6) (06 2016) 810–835.
[62] J. Pugh, A. Martinoli, Inspiring and modeling multi-robot search with particle swarm optimization, in: IEEE Swarm Intelligence Symposium, IEEE Computer Society, Washington, DC, USA, 2007, pp. 332–339.
[63] Songdong Xue, Li Jin, Jian-Chao Zeng, Xiaojuan He, Guoyou Zhang, Synchronous and Asynchronous Communication Modes for Swarm Robotic Search, vol. 12, 2011, pp. 311–328 (chapter 14).
[64] G.S. Nitschke, M.C. Schut, A.E. Eiben, Evolving behavioral specialization in robot teams to solve a collective construction task, Swarm Evol. Comput. 2 (2012) 25–38.
[65] Dongshu Wang, Haitao Wang, Lei Liu, Unknown environment exploration of multi-robot system with the FORDPSO, Swarm Evol. Comput. 26 (2016) 157–174.
[66] Sá Alan, Nadia Nedjah, Luiza Mourelle, Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms, Neurocomputing 172 (2016) 322–336.
[67] Patricia Suárez, Andrés Iglesias, Akemi Gálvez, Make robots be bats: specializing robotic swarms to the bat algorithm, Swarm Evol. Comput. 44 (2019) 113–129.
[68] Martin Rosalie, Grégoire Danoy, Serge Chaumette, Bouvry Pascal, Chaos-enhanced mobility models for multilevel swarms of UAVs, Swarm Evol. Comput. 41 (2018) 36–48.
[69] Marco Dorigo, Elio Tuci, Roderich Groß, Vito Trianni, ThomasHalva Labella, Shervin Nouyan, Christos Ampatzis, Jean-Louis Deneubourg, Gianluca Baldassarre, Stefano Nolfi, Francesco Mondada, Dario Floreano, LucaMaria Gambardella, The SWARM-BOTS project, in: Erol Şahin, William M. Spears (Eds.), Swarm Robotics, Volume 3342 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 31–44.
[70] R. Gross, M. Bonani, F. Mondada, M. Dorigo, Autonomous self-assembly in swarm-bots, IEEE Trans. Robot. 22 (6) (2006) 1115–1130.
[71] Marco Dorigo, Dario Floreano, Luca Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, et al., Swarmanoid: a novel concept for the study of heterogeneous robotic swarms, IEEE Robot. Autom. Mag. 20 (4) (2013) 60–71.

[72] McLurkin James, Jennifer Smith, Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots, in: Rachid Alami, Raja Chatila, Hajime Asama (Eds.), Distributed Autonomous Robotic Systems, vol. 6, Springer Japan, 2007, pp. 399–408.
[73] McLurkin James, Andrew Lynch, Rixner Scott, Thomas Barr, Alvin Chou, Kathleen Foster, Siegfried Bilstein, A low-cost multi-robot system for research, teaching, and outreach, in: Distributed Autonomous Robotic Systems, Springer, 2013, pp. 597–609.
[74] McLurkin James, McMullen Adam, Nick Robbins, Golnaz Habibi, Aaron Becker, Alvin Chou, Hao Li, Michael John, Nnena Okeke, Jarogniew Rykowski, et al., A robot system design for low-cost multi-robot manipulation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 912–918.
[75] V. Longchamp, M. Bonani, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, F. Mondada, The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010.
[76] M. BenAri, First Steps in Robotics with the Thymio Robot and the ASEBA/VPL Environment, 2013.
[77] Jorge Soares, Iñaki Navarro, Alcherio Martinoli, The khepera iv mobile robot: performance evaluation, sensory data and software toolbox, in: Robot 2015: Second Iberian Robotics Conference, Springer, 2016, pp. 767–781.
[78] G. Caprari, T. Estier, R. Siegwart, Fascination of down scaling - alice the sugar cube robot, J. Micromechatron. 1 (3) (2002) 177–189.
[79] J. Klingner, A. Kanakia, N. Farrow, D. Reishus, N. Correll, A stick-slip omnidirectional powertrain for low-cost swarm robotics: mechanism, calibration, and control, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 846–851.
[80] Kettler Alexander, Marc Szymanski, Wörn Heinz, The wanda robot and its development system for swarm algorithms, in: Ulrich Rückert, Sitte Joaquin, Werner Felix (Eds.), Advances in Autonomous Mini Robots, Springer, Berlin, Heidelberg, 2012, pp. 133–146.
[81] Serge Kernbach, Swarmrobot.org — Open-Hardware Microrobotic Project for Large-Scale Artificial Swarms, CoRR, abs/1110.5762, 2011.
[82] Francesco Mondada, Edoardo Franzi, Andre Guignard, The development of khepera, in: Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop, Number LSRO-CONF-2006-060 in HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut 64, 1999, pp. 7–14.
[83] Judith Challinger, Efficient use of robots in the undergraduate curriculum, in: ACM SIGCSE Bulletin, vol. 37, ACM, 2005, pp. 436–440.
[84] Jim Pugh, Xavier Raemy, Cédric Favre, Riccardo Falconi, Alcherio Martinoli, A fast onboard relative positioning module for multirobot systems, Mech., IEEE/ASME Trans. 14 (2) (2009) 151–162.
[85] S. Garnier, F. Tache, M. Combe, A. Grimal, G. Theraulaz, Alice in pheromone land: an experimental setup for the study of ant-like robots, in: Swarm Intelligence Symposium, IEEE Computer Society, Washington, DC, USA, 2007, pp. 37–44.
[86] Rebecca Boyle, Ping pong ball-sized robots can swarm together to form a smart liquid, Popul. Sci. (12) (2012).
[87] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Klaptocz Adam, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, Alcherio Martinoli, The e-puck, a robot designed for education in engineering, in: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.
[88] Betthauser Joseph, Daniel Benavides, Jeff Schornick, O'Hara Neal, Jatin Patel, Jeremy Cole, Edgar Lobaton, WolfBot: a distributed mobile sensing platform for research and education, in: American Society for Engineering Education, 2014 Zone 1 Conference of the, IEEE, 2014, pp. 1–8.
[89] Sean Wilson, Ruben Gameros, Michael Sheely, Matthew Lin, Kathryn Dover, Robert Gevorkyan, Matt Haberland, Andrea Louise Bertozzi, Spring Berman, Pheeno, a versatile swarm robotic research and education platform, IEEE Robot. Automat. Lett. 1 (2) (July 2016) 884–891.
[90] Michael Rubenstein, Christian Ahler, Nick Hoff, Adrian Cabrera, Radhika Nagpal, Kilobot: a low cost robot with scalable operations designed for collective behaviors, Robot. Auton. Syst. 62 (7) (2014) 966–975.
[91] Farshad Arvin, Jose Luis Espinosa Mendoza, Benjamin Bird, Andrew West, Simon Watson, Lennox Barry, Mona: an affordable mobile robot for swarm robotic applications, in: Proceedinsg of the UKRAS Conference on Robotics and Autonomous Systems, 2017, pp. 49–52.
[92] A. Gutierrez Martín, A. Campo, M. Dorigo, J. Donate, Open e-puck range & bearing miniaturized board for local communication in swarm robotics. International Conference on Robotics and Automation, IEEE, 2009.
[93] Gianpiero Francesca, Mauro Birattari, Automatic design of robot swarms: achievements and challenges, Front. Robot. AI 3 (29) (2016).
[94] Carlo Pinciroli, Vito Trianni, Rehan OÕGrady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, Marco Dorigo, ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems, Swarm Intell. 6 (2012) 271–295.
[95] James Kramer, Matthias Scheutz, Development environments for autonomous mobile robots: a survey, Aut. Robots 22 (2007) 101–132.
[96] Brian Gerkey, Richard Vaughan, Andrew Howard, Nate Koenig, The Player/stage Project, 2003. Available at: http://playerstage.sourceforge.net.
[97] Nathan Koenig, Andrew Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, IEEE, 2004, pp. 2149–2154.
[98] S. Magnenat, M. Waibel, A. Beyeler, Enki: the Fast 2D Robot Simulator, 2007. Available at: http://home.gna.org/enki/.
[99] Olivier Michel, Webots: Professional Mobile Robot Simulation, 2004. arXiv preprint cs/0412052.
[100] Eric Rohmer, Surya Singh, Marc Freese, V-REP: a versatile and scalable robot simulation framework, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Nov 2013, pp. 1321–1326.
[101] Carlo Pinciroli, Vito Trianni, Rehan OÕGrady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al., ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems, Swarm Intell. 6 (4) (2012) 271–295.
[102] Stefano Carpin, Mike Lewis, Jijun Wang, Balakirsky Stephen, Chris Scrapper, USARSim: a robot simulator for research and education, in: IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 1400–1405.
[103] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, RoboCup: the robot world cup initiative, in: Proceedings of the First International Conference on Autonomous Agents, ACM, 1997, pp. 340–347.
[104] Nicolas Bredeche, Jean-Marc Montanier, Berend Weel, Evert Haasdijk, RoboRobo! a Fast Robot Simulator for Swarm and Collective Robotics, 2013. arXiv preprint arXiv:1304.2888.
[105] Gabriele Valentini, Antoun Anthony, Marco Trabattoni, Bernat Wiandt, Yasumasa Tamura, Etienne Hocquard, Vito Trianni, Marco Dorigo, Kilogrid: a novel experimental environment for the kilobot robot, Swarm Intell. 12 (2018) 245–266 12.
[106] Andreagiovanni Reina, Alex J. Cope, Eleftherios Nikolaidis, A. James, R. Marshall, Chelsea Sabo, Ark: augmented reality for kilobots, IEEE Robot. Automat. Lett. 2 (2017) 1755–1761.
[107] McLurkin James, Jennifer Smith, James Frankel, David Sotkowitz, David Blau, Brian Schmidt, Speaking swarmish: human-robot interface design for large swarms of autonomous mobile robots, in: AAAI Spring Symposium: to Boldly Go where No Human-Robot Team Has Gone before, 2006, pp. 72–75.
[108] Lingkang Zhang, Richard Vaughan, Optimal gaze-based robot selection in multi-human multi-robot interaction, in: 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2016, pp. 645–646.
[109] A. Giusti, J. Nagi, L. Gambardella, S. Bonardi, G.A. Di Caro, Human-swarm interaction through distributed cooperative gesture recognition, in: Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session), 2012, p. 401. Boston, MA, USA, March 5-8.
[110] Luca Mondada, Mohammad Ehsanul Karim, Francesco Mondada, Electroencephalography as implicit communication channel for proximal interaction between humans and robot swarms, Swarm Intell. 10 (4) (Dec 2016) 247–265.
[111] Mathieu Le Goc, Lawrence H. Kim, Parsaei Ali, Jean-Daniel Fekete, Pierre Dragicevic, Sean Follmer Zooids, Building blocks for swarm user interfaces, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, ACM, New York, NY, USA, 2016, pp. 97–109.
[112] Adrian N. Bishop, Pubudu Pathirana, Andrey Savkin, Target Tracking with Range and Bearing Measurements via Robust Linear Filtering, 2008, pp. 131–135.
[113] Nicholi Shiell, Andrew Vardy, A bearing-only pattern formation algorithm for swarm robotics, in: Marco Dorigo, Mauro Birattari, Xiaodong Li, Manuel López-Ibáñez, Kazuhiro Ohkura, Carlo Pinciroli, Thomas Stützle (Eds.), Swarm Intelligence, Springer International Publishing, Cham, 2016, pp. 3–14.
[114] Luneque Silva Junior, Nadia Nedjah, Wave algorithm applied to collective navigation of robotic swarms, Appl. Soft Comput. 57 (2017) 698–707.
[115] Álvaro Gutiérrez, Alexandre Campo, Marco Dorigo, Jesus Donate, Félix Monasterio-Huelin, Luis Magdalena, Open e-puck range & bearing miniaturized board for local communication in swarm robotics, in: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, IEEE Press, Piscataway, NJ, USA, 2009, pp. 1745–1750.
[116] Sá Alan, Nadia Nedjah, Luiza Mourelle, Distributed efficient localization in swarm robotics using min–max and particle swarm optimization, Expert Syst. Appl. 50 (2016) 55–65.
[117] Cadena Cesar, Luca Carlone, Carrillo Henry, Yasir Latif, Davide Scaramuzza, Neira Jose, Ian Reid, John J. Leonard, Past, present, and future of simultaneous localization and mapping: toward the robust-perception age, Trans. Rob. 32 (6) (December 2016) 1309–1332.
[118] Jim Pugh, Alcherio Martinoli, Relative localization and communication module for small-scale multi-robot systems, in: IEEE International Conference on Robotics and Automation, 2006, pp. 188–193.
[119] Jim Pugh, Xavier Raemy, Cedric Favre, Riccardo Falconi, Alcherio Martinoli, A fast on-board relative positioning module for multi-robot systems, IEEE/ASME Trans. Mechatron., Focused Sect. Mechatron. Multi Robot Syst. 14 (2009) 151–162.
[120] I.D. Kelly, A. Martinoli, A scalable, on-board localisation and communication system for indoor multi-robot experiments, Sens. Rev. 24 (2) (2004) 167–180.
[121] James F. Roberts, Timothy Stirling, Jean-Christophe Zufferey, Dario Floreano, 3-D relative positioning sensor for indoor flying robots, Aut. Robots 33 (1) (Aug 2012) 5–20.
[122] Jawhar Ghommam, Hasan Mehrjerdi, Maarouf Saad, Leader-follower formation control of nonholonomic robots with fuzzy logic based approach for obstacle avoidance, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 2340–2345.

[123] Gregory Dudek, Michael Jenkin, Evangelosand Milios, David Wilkes, A taxonomy for swarm robots, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 1993, pp. 441–447. part I.
[124] Y Uni Cao, Alex Fukunaga, Andrew Kahng, Frank Meng, Cooperative mobile robotics: antecedents and directions, in: Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots, Proceedings. 1995 IEEE/RSJ International Conference on, vol. 1, 1995, pp. 226–234.
[125] Maja J. Mataric, IEEE Trans. Robot. Autom. 8 (1992) 304–312.
[126] Luca Iocchi, Daniele Nardi, Massimiliano Salerno, Reactivity and deliberation: a survey on multi-robot systems, in: Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Volume 2103 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 9–32.
[127] Levent Bayındır, Erol Şahin, A review of studies in swarm robotics, Turkish J. Electr. Eng. 15 (2) (2007) 115–147.
[128] Levent Bayındır, A review of swarm robotics tasks, Neurocomputing 172 (2016) 292–321.
[129] Gregory Dudek, Michael Jenkin, Evangelosand Milios, David Wilkes, A taxonomy for multi-agent robotics, Aut. Robots 3 (4) (1996) 375–397.
[130] Yogeswaran Mohan, S. Ponnambalam, An extensive review of research in swarm robotics, in: World Congress on Nature & Biologically Inspired Computing, IEEE, 2009, pp. 140–145.
[131] Iñaki Navarro, Fernando Matía, An introduction to swarm robotics, ISRN Robot. 2013 (2013) 1–10.
[132] Ying Tan, Zhong-yang Zheng, Research advance in swarm robotics, Defence Technol. 9 (1) (2013) 18–39.
[133] Jan Carlo Barca, Y. Ahmet Sekercioglu, Swarm robotics reviewed, Robotica 31 (03) (2013) 345–359.
[134] Purushotham Muniganti, Pujol Albert, A survey on mathematical models of swarm robotics, in: Workshop of Physical Agents, 2010, pp. 29–30.
[135] Grégory Mermoud, Utkarsh Upadhyay, William Evans, Alcherio Martinoli, Top-down vs. bottom-up model-based methodologies for distributed control: a comparative experimental study, in: Experimental Robotics, Springer, 2014, pp. 615–629.
[136] Kristina Lerman, Alcherio Martinoli, Aram Galstyan, A review of probabilistic macroscopic models for swarm robotic systems, in: Erol Şahin, William M. Spears (Eds.), Swarm Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 143–152.
[137] H. Hamann, H. Wörn, A framework of space-time continuous models for algorithm design in swarm robotics, Swarm Intell. 2 (2) (2008) 209–239.
[138] Rachid Alami, Sara Fleury, Matthieu Herrb, Félix Ingrand, Frédéric Robert, Multi-robot cooperation in the MARTHA project, IEEE Robot. Autom. Mag. 5 (1) (1998) 36–47.
[139] Vittorio Ziparo, Luca Iocchi, Pedro Lima, Daniele Nardi, Pier Palamara, Petri net plans, Aut. Agents Multi-Agent Syst. 23 (3) (2011) 344–383.
[140] Arne Brutschy, Lorenzo Garattoni, Manuele Brambilla, Gianpiero Francesca, Giovanni Pini, Marco Dorigo, Mauro Birattari, The TAM: abstracting complex tasks in swarm robotics research, Swarm Intell. 9 (1) (2015) 1–22.
[141] Josh Bongard, Evolutionary robotics, Commun. ACM 56 (8) (2013) 74–83.
[142] Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Oliveira, Anders Christensen, Evolution of collective behaviors for a real swarm of aquatic surface robots, PLoS One 11 (3) (2016) e0151834.
[143] Jorge Gomes, Paulo Urbano, Anders Lyhne Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intell. 7 (2–3) (2013) 115–144.
[144] Geoff Nitschke, Martijn Schut, A. Eiben, Evolving behavioral specialization in robot teams to solve a collective construction task, Swarm Evolut. Comput. 2 (2012) 25–38.
[145] Vito Trianni, Manuel López-Ibáñez, Advantages of task-specific multi-objective optimisation in evolutionary robotics, PLoS One 10 (8) (2015) e0136406.
[146] V. Trianni, Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots, Springer Verlag, Berlin, Germany, 2008.
[147] Eric Bonabeau, Theraulaz Guy, Jean-Louls Deneubourg, Serge Aron, Camazine Scott, Self-organization in social insects, Trends Ecol. Evol. 12 (5) (1997) 188–193.
[148] C. Kube, Eric Bonabeau, Cooperative transport by ants and robots, Robot. Auton. Syst. 30 (1) (2000) 85–101.
[149] S. Nouyan, R. Gross, M. Bonani, M. Dorigo F. Mondada, Teamwork in self-organized robot colonies, IEEE Trans. Evol. Comput. 13 (4) (2009) 695–711.
[150] William Spears, Diana Spears, Jerry Hamann, Rodney Heil, Distributed, physics-based control of swarms of vehicles, Aut. Robots 17 (2–3) (2004) 137–162.
[151] Farshad Arvin, Ali Emre Turgut, Farhad Bazyari, Kutluk Arikan, Nicola Bellotto, Shigang Yue, Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method, Adapt. Behav. 22 (05 2014) 189–206.
[152] Farshad Arvin, Ali Emre Turgut, Tomás Krajník, Salar Rahimi, Ilkin Ege Okay, Shigang Yue, Simon Watson, Lennox Barry, ϕ clust: pheromone-based aggregation for robotic swarms, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018, 2018, pp. 4288–4294.
[153] Simon Garnier, Jacques Gautrais, Masoud Asadpour, Christian Jost, Theraulaz Guy, Self-organized aggregation triggers collective decision making in a group of cockroach-like robots, Adapt. Behav. 17 (2) (2009) 109–133.
[154] Sarika Ramroop, Farshad Arvin, Simon Watson, Joaquin Carrasco, Lennox Barry, A bio-inspired aggregation with robot swarm using real and simulated mobile robots, in: Maria Elena Giannaccini, Manuel Giuliani, Tareq Assaf (Eds.), Proceedings 19th Annual Conference, TAROS 2018: towards Autonomous Robotic Systems, Lecture Notes in Computer Science, Springer-Verlag, 2018, pp. 317–329.
[155] Ziya Firat, Eliseo Ferrante, Nicolas Cambier, Elio Tuci, Self-organised aggregation in swarms of robots with informed robots, in: Theory and Practice of Natural Computing - 7th International Conference, TPNC 2018, Dublin, Ireland, December 12-14, 2018, Proceedings, 2018, pp. 49–60.
[156] Luke Ludwig, Maria Gini, Robotic swarm dispersion using wireless intensity signals, in: Distributed Autonomous Robotic Systems, vol. 7, Springer, 2006, pp. 135–144.
[157] Emre Ugur, Turgut Ali, Erol Şahin, Dispersion of a swarm of robots based on realistic wireless intensity signals, in: 22nd International Symposium on Computer and Information Sciences, IEEE, 2007, pp. 1–6.
[158] Philip Necsulescu, Klaus Schilling, Automation of a multiple robot self-organizing multi-hop mobile ad-hoc network (MANET) using signal strength, in: IEEE International Conference Instrumentation and Measurement Technology, IEEE, 2015, pp. 505–510.
[159] Gamal Sallam, Uthman Baroudi, COVER: a cooperative virtual force robot deployment technique, in: Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015 IEEE International Conference on, IEEE, 2015, pp. 1308–1315.
[160] Uthman Baroudi, Gamal Sallam, Mohammed Al-Shaboti, Mohamed Younis, GPS-free robots deployment technique for rescue operation based on landmark's criticality, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International, IEEE, 2015, pp. 367–372.
[161] Mihai Andries, François Charpillet, Multi-robot taboo-list exploration of unknown structured environments, in: Intelligent Robots and Systems IROS, 2015 IEEE/RSJ International Conference on, IEEE, 2015, pp. 5195–5201.
[162] Ashish Macwan, Julio Vilela, Goldie Nejat, Beno Benhabib, Multi-robot deployment for wilderness search and rescue, Int. J. Robot. Autom. 31 (1) (2016).
[163] Bijan Ranjbar-Sahraei, Gerhard Weiss, Nakisaee Ali, A multi-robot coverage approach based on stigmergic communication, in: Multiagent System Technologies, Springer, 2012, pp. 126–138.
[164] Ipek Caliskanelli, Bastian Broecker, Karl Tuyls, Multi-robot coverage: a bee pheromone signalling approach, in: Artificial Life and Intelligent Agents, Springer, 2014, pp. 124–140.
[165] Bastian Broecker, Ipek Caliskanelli, Karl Tuyls, I Sklar Elizabeth, Daniel Hennes, Hybrid insect-inspired multi-robot coverage in complex environments, in: Towards Autonomous Robotic Systems, Springer, 2015, pp. 56–68.
[166] Bastian Broecker, Ipek Caliskanelli, Karl Tuyls, Elizabeth Sklar, Daniel Hennes, Social insect-inspired multi-robot coverage, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1775–1776.
[167] John Reif, Hongyan Wang, Social potential fields: a distributed behavioral control for autonomous robots, Robot. Auton. Syst. 27 (3) (1999) 171–194.
[168] Camazine Scott, Sneyd James, A model of collective nectar source selection by honey bees: self-organization through simple rules, J. Theor. Biol. 149 (4) (1991) 547–571.
[169] Eshel Ben-Jacob, Bacterial self-organization: co-enhancement of complexification and adaptability in a dynamic environment, Phil. Trans. R. Soc. Lond. A: Math., Phys. Eng. Sci. 361 (1807) (2003) 1283–1312.
[170] G. Springholz, V. Holy, M. Pinczolits, G. Bauer, Self-organized growth of three-dimensional quantum-dot crystals with FCC-like stacking and a tunable lattice constant, Science 282 (5389) (1998) 734–737.
[171] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, Peter Widmayer, Arbitrary pattern formation by asynchronous, anonymous, oblivious robots, Theor. Comput. Sci. 407 (1) (2008) 412–447.
[172] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Siegwart Roland, Beardsley Paul, Multi-robot system for artistic pattern formation, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 4512–4517.
[173] Masafumi Yamashita, Ichiro Suzuki, Characterizing geometric patterns formable by oblivious anonymous mobile robots, Theor. Comput. Sci. 411 (26) (2010) 2433–2453.
[174] Levi Paul, Serge Kernbach, Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution, vol. 7, Springer Science & Business Media, 2010.
[175] Wenguo Liu, Alan Winfield, Self-assembly in heterogeneous modular robots, in: Distributed Autonomous Robotic Systems, Springer, 2014, pp. 219–232.
[176] Josep Aulinas, Yvan Petillot, Joaquim Salvi, Xavier Lladó, The SLAM problem: a survey, in: CCIA, Citeseer, 2008, pp. 363–371.
[177] Carlos Nieto-Granda, John Rogers, Henrik Christensen, Coordination strategies for multi-robot exploration and mapping, Int. J. Robot. Res. (2014) (paper = 0278364913515309) SAGE Publications.

[178] Faigl Jan, Miroslav Kulich, On benchmarking of frontier-based multi-robot exploration strategies, in: European Conference on Mobile Robots, IEEE, 2015, pp. 1–8.
[179] Katherine Russell, Michael Schader, Kevin Andrea, Sean Luke, Swarm robot foraging with wireless sensor motes, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 287–295.
[180] Somchaya Liemhetcharat, Rui Yan, Keng Peng Tee, Matthew Lee, Multi-robot item delivery and foraging: two sides of a coin, Robotics 4 (3) (2015) 365–397.
[181] Lenka Pitonakova, Richard Crowder, Seth Bullock, Information flow principles for plasticity in foraging robot swarms, Swarm Intell. 10 (1) (2016) 33–63.
[182] Alan Winfield, Foraging robots, in: Encyclopedia of Complexity and Systems Science, Springer, 2009, pp. 3682–3700.
[183] Tucker Balch, Ronald Arkin, Behavior-based formation control for multirobot teams, IEEE Trans. Robot. Automat. 14 (6) (1998) 926–939.
[184] Iñaki Navarro, Fernando Matía, A survey of collective movement of mobile robots, Int. J. Adv. Robot. Syst. 10 (73) (2013).
[185] Craig Reynolds, Flocks, herds and schools: a distributed behavioral model, in: ACM SIGGRAPH Computer Graphics, vol. 21, ACM, 1987, pp. 25–34.
[186] Turgut Ali, Hande Çelikkanat, Fatih Gökçe, Erol Şahin, Self-organized flocking in mobile robot swarms, Swarm Intell. 2 (2–4) (2008) 97–120.
[187] Bayu Erfianto, Bambang Trilaksono, Simulation of swarm robot flocking assisted by explicit communication, J. Unmanned Syst. Technol. 1 (2) (2013) 49–57.
[188] Toshiyuki Yasuda, Akitoshi Adachi, Kazuhiro Ohkura, Self-organized flocking of a mobile robot swarm by topological distance-based interactions, in: IEEE/SICE International Symposium on System Integration, IEEE, 2014, pp. 106–111.
[189] Christoph Moeslinger, Thomas Schmickl, Karl Crailsheim, A minimalist flocking algorithm for swarm robots, in: Kampis George, István Karsai, Eörs Szathmáry (Eds.), Advances in Artificial Life. Darwin Meets von Neumann, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 375–382.
[190] ZhiDong Wang, Vijay Kumar, Object closure and manipulation by multiple cooperating mobile robots, in: IEEE International Conference on Robotics and Automation, vol. 1, IEEE, 2002, pp. 394–399.
[191] Guilherme Pereira, Vijay Kumar, John Spletzer, Camillo Taylor, Mario Campos, Cooperative transport of planar objects by multiple mobile robots using object closure, in: Experimental Robotics VIII, Springer, 2003, pp. 287–296.
[192] Jianing Chen, Melvin Gauci, Wei Li, Andreas Kolling, and Roderich Gros. Occlusion-based cooperative transport with a swarm of miniature mobile robots, IEEE Trans. Robot. 31 (2) (2015) 307–321.
[193] Michael Rubenstein, Adrian Cabrera, Justin Werfel, Golnaz Habibi, McLurkin James, Radhika Nagpal, Collective transport of complex objects by simple robots: theory and experiments, in: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 47–54.
[194] Elio Tuci, H. Muhanad, Mohammed Alkilabi, Otar Akanyeti, Cooperative object transport in multi-robot systems: a review of the state-of-the-art, Front. Robot. AI (2018) 2018.
[195] Gabriele Valentini, Heiko Hamann, Marco Dorigo, Efficient decision-making in a self-organizing robot swarm: on the speed versus accuracy trade-off, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1305–1314.
[196] Gabriele Valentini, Heiko Hamann, Marco Dorigo, Self-organized collective decision-making in a 100-robot swarm, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[197] Gabriele Valentini, Eliseo Ferrante, Heiko Hamann, Marco Dorigo, Collective decision with 100 kilobots: speed versus accuracy in binary discrimination problems, Aut. Agents Multi-Agent Syst. 30 (3) (2016) 553–580.
[198] Michael Krieger, Jean-Bernard 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.
[199] Chris Jones, Maja Matarić, Adaptive division of labor in large-scale minimalist multi-robot systems, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, Oct 2003, pp. 1969–1974. vol. 2.
[200] G. Caprari, Elisa-3 Robot, 2016. Available at: http://www.gctronic.com/doc/index.php/Elisa-3.
[201] Nadia Nedjah, Rafael Mendonça, Luiza Mourelle, PSO-based distributed algorithm for dynamic task allocation in a robotic swarm, Procedia Comput. Sci. 51 (2015) 326–335.
[202] Nicolás Cruz, Nadia Nedjah, Luiza Mourelle, Efficient spacial clustering in swarm robotics, in: Computational Science and its Applications, Springer, 2015, pp. 14–27.
[203] Luneque Silva Jr., Nadia Nedjah, Wave algorithm for recruitment in swarm robotics, in: International Conference on Computational Science and its Applications, Springer, 2015, pp. 3–13.
[204] Eliseo Ferrante, Ali Emre Turgut, A. Edgar, Duéñez-Guzmán, Marco Dorigo, Wenseleers Tom, Evolution of self-organized task specialization in robot swarms, PLoS Comput. Biol. 11 (8) (2015).
[205] Abdulmuttalib Rashid, Mattia Frasca, Abduladhem Ali, Alessandro Rizzo, Luigi Fortuna, Multi-robot localization and orientation estimation using robotic cluster matching algorithm, Robot. Auton. Syst. 63 (2015) 108–121.
[206] Ying Tan, yang Zheng Zhong, Research advance in swarm robotics, Defence Technol. 9 (1) (2013) 18–39.
[207] Ronald Thenius, Daniel Moser, Joshua Cherian Varughese, Serge Kernbach, Igor Kuksin, Olga Kernbach, Elena Kuksina, Nikola Mišković, Stjepan Bogdan, Tamara Petrović, Anja Babić, Frédéric Boyer, Lebastard Vincent, Stéphane Bazeille, Graziano William Ferrari, Elisa Donati, Riccardo Pelliccia, Donato Romano, Godfried Jansen Van Vuuren, Cesare Stefanini, Matteo Morgantin, Alexandre Campo, Thomas Schmickl, subCULTron — cultural development as a tool in underwater robotics, in: Peter R. Lewis, Christopher J. Headleand, Steve Battle, Panagiotis D. Ritsos (Eds.), Artificial Life and Intelligent Agents, Springer International Publishing, Cham, 2018, pp. 27–41.
[208] Shuguang Li, Richa Batra, David Brown, Hyun-Dong Chang, Nikhil Ranganathan, Chuck Hoberman, Daniela Rus, Hod Lipson, Particle robotics based on statistical mechanics of loosely coupled components, Nature 567 (7748) (2019) 361–365.
[209] Olivier Michel, Webots: symbiosis between virtual and real mobile robots, in: Virtual Worlds, Springer, 1998, pp. 254–263.
[210] David St-Onge, Vivek Varadharajan, Giovanni Beltrame, ROS and Buzz: consensus-based behaviors for heterogeneous teams, in: International Conference on Intelligent Robots and Systems, IEEE, 2018, arXiv:1710.08843.
[211] Jean-Marc Montanier, Carrignon Simon, Nicolas Bredeche, Behavioral specialization in embodied evolutionary robotics: why so difficult? Front. Robot. AI 3 (2016) 38.
[212] Ezequiel Di Mario, Iñaki Navarro, Alcherio Martinoli, Distributed learning of cooperative robotic behaviors using particle swarm optimization, in: Experimental Robotics, Springer, 2016, pp. 591–604.
[213] Alan Winfield, Sa Jin, Mari-Carmen Fernández-Gago, Clare Dixon, Michael Fisher, On formal specification of emergent behaviours in swarm robotic systems, Int. J. Adv. Robot. Syst. 2 (4) (2005) 363–370.
[214] N. Bezzo, A. Mehta, C. Onal, M. Tolley, Robot makers: the future of digital rapid design and fabrication of robots, IEEE Robot. Autom. Mag. 22 (4) (Dec 2015) 27–36.
[215] Barry Berman, 3-D printing: the new industrial revolution, Bus. Horiz. 55 (2) (2012) 155–162.
[216] Jingjin Yu, Shuai Han, Daniela Rus, A Portable, 3d-Printing Enabled Multi-Vehicle Platform for Robotics Research and Education, 2016. arXiv preprint arXiv:1609.04745.
[217] S. Pan, S. Guo, L. Shi, Y. He, Z. Wang, Q. Huang, A spherical robot based on all programmable SoC and 3-D printing, in: 2014 IEEE International Conference on Mechatronics and Automation, Aug 2014, pp. 150–155.
[218] N. Bezzo, J. Park, A. King, P. Gebhard, R. Ivanov, I. Lee, Demo abstract: ROSLab; a modular programming environment for robotic applications, in: ACM/IEEE International Conference on Cyber-Physical Systems, April 2014, 214–214.
[219] Ankur Mehta, Nicola Bezzo, Peter Gebhard, Byoungkwon An, Vijay Kumar, Insup Lee, Daniela Rus, A Design Environment for the Rapid Specification and Fabrication of Printable Robots, Springer International Publishing, Cham, 2016, pp. 435–449.
[220] Nicola Bezzo, Peter Gebhard, Insup Lee, Matthew Piccoli, Vijay Kumar, Mark Yim, Rapid co-design of electro-mechanical specifications for robotic systems, in: ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2015, V009T07A009–V009T07A009.
[221] Renzo De Nardi, Owen Holland, UltraSwarm: a further step towards a flock of miniature helicopters, in: Swarm Robotics, Springer, 2006, pp. 116–128.
[222] Timothy Stirling, Dario Floreano, Energy-time efficiency in aerial swarm deployment, in: Distributed Autonomous Robotic Systems, Springer, 2013, pp. 5–18.
[223] Robert Wood, Fly, robot, fly, Spectrum 45 (3) (2008) 25–29.
[224] Jifeng Ma, Chunmei Zhou, Shuo Zhang, Wenjie Dong, Chunxia Zhang, Jinyong Lin, The design of simulation system for Multi-UAV cooperative guidance, in: Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, IEEE, 2015, pp. 1250–1254.
[225] Truszkowski Walt, Mike Hinchey, James Rash, Christopher Rouff, NASA's swarm missions: the challenge of building autonomous software, IT Profess. 6 (5) (2004) 47–52.
[226] Emil Vassev, Mike Hinchey, Joey Paquet, Towards an assl specification model for nasa swarm-based exploration missions, in: Proceedings of the 2008 ACM Symposium on Applied Computing, ACM, 2008, pp. 1652–1657.
[227] Thomas Schmickl, Ronald Thenius, Christoph Moslinger, Jon Timmis, Andy Tyrrell, Mark Read, James Hilder, Jose Halloy, Alexandre Campo, Cesare Stefanini, Luigi Manfredi, Stefano Orofino, Serge Kernbach, Tobias Dipper, Donny Sutantyo, CoCoRo — the Self-Aware Underwater Swarm, vol. 10, 2011, pp. 120–126.
[228] Tamás Vicsek, András Czirók, Illés J. Farkas, Dirk Helbing, Application of statistical mechanics to collective motion in biology, Phys. A Stat. Mech. Appl. 274 (1) (1999) 182–189.
[229] Andrea Cavagna, Irene Giardina, S. Tomas, Grigera. The physics of flocking: correlation as a compass from experiments to theory, Phys. Rep. 728 (2018) 1–62 (The physics of flocking: Correlation as a compass from experiments to theory).
[230] Thomas Bose, Andreagiovanni Reina, James Marshall, Collective decision-making, Current Opinion Behav. Sci. 16 (2017) 30–34.
[231] Tian Shen, Emergent Collective Decision-Making: Control, Model and Behavior, PhD thesis, 2015.
[232] Andrea Cavagna, Irene Giardina, S. Tomas, Grigera. The physics of flocking: correlation as a compass from experiments to theory, Phys. Rep. 728 (1) (2018) 1–62.

[233] Heiko Hamann, Wörn Heinz, A framework of space–time continuous models for algorithm design in swarm robotics, Swarm Intell. 2 (2) (Dec 2008) 209–239.
[234] Edd Gent, The 10 Grand Challenging Facing Robotics in the Next Decade, February 2018, https://singularityhub.com/2018/02/06/.
[235] Jason Ernest, Challenges Holding Back Swarm Robotics, 2015, https://www.roboticsbusinessreview.com/rbr/.
[236] Farshad Arvin, Simon Watson, Ali Emre Turgut, Jose Luis Espinosa Mendoza, Tomas Krajnik, Lennox Barry, Perpetual robot swarm: long-term autonomy of mobile robots using on-the-fly inductive charging, J. Intell. Robot. Syst. 92 (3–4) (2017) 395–412.
[237] Alessandro Massaro, Fabrizio Spano, Paolo Cazzato, Carola La Tegola, Roberto Cingolani, Athanassia Athanassiou, Robot tactile sensing: gold nanocomposites as highly sensitive real-time optical pressure sensors, IEEE Robot. Autom. Mag. 20 (2) (2013) 82–90.
[238] Alessandro Massaro, Fabrizio Spano, Diego Caratelli, Alexander G. Yarovoy, Roberto Cingolani, Athanassia Athanassiou, New approaches of nanocomposite materials for electromagnetic sensors and robotics, Int. J. Meas. Technol. Instrum. Eng. 1 (1) (2011) 55–72.
[239] Alessandro Massaro, Fabrizio Spano, Aimé Lay-Ekuakille, Paolo Cazzato, Roberto Cingolani, Athanassia Athanassiou, Design and characterization of a nanocomposite pressure sensor implemented in a tactile robotic system, IEEE Trans. Instrum. Measur. 60 (8) (2011) 2967–2975.
[240] Micky Rakotondrabe (Ed.), Smart Materials-Based Actuators at the Micro/Nano-Scale Characterization, Control, and Applications, springer, 2013.
[241] Titas Bera, Rajarshi Bardhan, Sundaram Suresh, An Integrated Decision and Control Theoretic Solution to Multi-Agent Co-operative Search Problems, ArXiv, (1704.07158), 2017.
[242] Hyunyoung Choi, Hal Varian, Predicting the present with Google trends, Econ. Rec. 88 (s1) (2012) 2–9.
[243] Wikipedia contributors, Swarm Robotics, 2019. [Online; accessed 10-April-2019].
[244] D. Pickem, P. Glotfelter, L. Wang, A. Mote, E. Feron Ames, M. Egerstedt, The Robotarium: a remotely accessible swarm robotics research testbed, in: IEEE International Conference on Robotics and Automation, IEEE, Singapore, 2017, pp. 1699–1706.