Springer Handbook of Computational Intelligence

Kacprzyk
Pedrycz
Editors

Springer

66. Swarm Intelligence in Optimization and Robotics

Christian Blum, Roderich Groß

Swarm intelligence is an artificial intelligence discipline, which was created on the basis of the laws that govern the behavior of, for example, social insects, fish schools, and flocks of birds. The organization of these animal societies has always mesmerized humans. Therefore, it is surprising that it has only been in the second half of the last century that some of the most important principles of swarm intelligent behavior have been unraveled. A prime example is stigmergy, which refers to a self-organization of the animal society via changes applied to the environment.

In this chapter, we provide a concise introduction to swarm intelligence, with two main research lines in mind: optimization and robotics. Popular examples of optimization algorithms based on swarm intelligence principles are ant colony optimization and particle swarm optimization. On the other side, the field of robotics has adopted various swarm intelligent behaviors for problem solving and organizing groups of robots. This has resulted in a separate research field nowadays known as swarm robotics.

66.1 Overview

Swarm intelligence (SI) [66.1–3] is a subfield of the more general field of artificial intelligence [66.4]. The term *swarm intelligence* was introduced and used for the first time by *Beni* et al. [66.5–7] in the context of cellular robotic systems. Nowadays, SI research is generally concerned with the design of intelligent multiagent systems whose inspiration is taken from the collective behavior of social – or even eusocial – insects and other animal populations. Examples include ant colonies, bee hives, wasp colonies, frog populations, flocks of birds, and fish schools. Among these, social insects have always played a prominent role in the inspiration of SI techniques. Even though their intrinsic ways of functioning have fascinated researchers for many years, the mechanisms that govern their behavior remained unknown for a long time. In colonies of social insects, for example, single colony members are unsophisticated individuals, yet they are able to achieve complex tasks in cooperation. Essential colony behaviors emerge from relatively simple interactions between the colony's individual members.

An important aspect of any SI system is *self-organization* [66.8]. Originally, the term *self-organization* was introduced by the German philosopher *Immanuel Kant* [66.9] in an attempt to characterize what makes organisms so different from other objects. Nowadays, the term self-organization refers to a process where some form of global order or coordination emerges from rather simple interactions between low-level components of an initially unordered system. Self-organizing processes are neither directed nor controlled by any agent or component, neither from inside nor from outside the system. They are often triggered by random fluctuations that are amplified by *positive feedback* and

Fig. 66.1 Ants cooperate for retrieving a heavy prey (photo courtesy of M. J. Blesa)

possibly counterbalanced by *negative feedback*, which generally aids in stabilizing the system. The global properties exhibited by self-organizing systems are thus the result of this distributed interplay of their components. As such, self-organization is typically robust and able to survive and self-repair damage or perturbations. Historically, self-organization processes have been studied in physical, chemical, biological, social, and cognitive systems. Well known examples are crystallization, molecular self-assembly, and the way in which neural networks learn to recognize complex patterns.

During the last 50 years or so, biologists discovered that many aspects of the collective activities of social insects are self-organized as well, that is, they function without a central control. For example, the African weaver ant constructs nests by pulling leaves together. Where the gap between leaves exceeds the body length of an individual ant, multiple ants organize into pulling chains. Once the leaves are in contact, they are glued together using silk from larvae, which are carried to the site by other workers of the colony [66.10]. Other examples concern the recruitment of fellow colony members for prey retrieval (Fig. 66.1), the capabilities of termites and wasps to build sophisticated nests, or the ability of bees and ants to orient themselves in their environment. For more examples, we refer the interested reader to [66.1, 2].

In the meantime, some of the above mentioned behaviors have been used as inspiration for the resolution of technical problems, especially in the context of optimization and in robotics. This chapter is dedicated to reviewing some of the – in the opinion of the authors – most interesting algorithms/systems from these two fields.

66.2 SI in Optimization

The use of SI techniques for solving optimization problems has already a rather extensive history. SI techniques have been used for both solving combinatorial and continuous optimization problems in static and in distributed settings. Two of the most well-known SI techniques for solving optimization problems are ant colony optimization (ACO) and particle swarm optimization (PSO). More recently, other techniques such as the artificial bee colony algorithm have been developed. Apart from solving optimization problems, SI techniques are being used for management tasks, for example, in distributed settings or in online optimization. The following sections will give a brief overview of this application field of SI.

66.2.1 Ant Colony Optimization

ACO [66.11] is one of the earliest SI techniques for optimization. Dorigo and colleagues developed the first ACO algorithms in the early 1990s [66.12–14]. The development of these algorithms was inspired by the observation of ant colonies. Ants are social insects. They live in colonies and their behavior is governed by the goal of colony survival rather than being focused on the survival of individuals. The behavior that provided the inspiration for ACO is the ants' foraging behavior, and in particular, how ants of many species can find shortest paths between food sources and their nest. In order to search for food, ants initially explore the area around their nest by means of random walks. While moving, ants leave tiny drops of a pheromone substance on the ground. Ants are also able to scent these pheromones. When choosing their way, they are attracted by paths marked by strong pheromone concentrations. When having identified a food source, ants evaluate the quantity and the quality of the food and carry some of it back to their nest. During the return trip, the quantity of pheromone that ants leave on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. It has been shown in [66.15] that the indirect communication between the ants via pheromone trails – known as *stigmergy* [66.16] – enables them to find the shortest paths between their nest and food sources. Initially, ACO algorithms were developed with the aim of solving discrete optimization problems. It should be mentioned, however, that nowadays the class of ACO algorithms also comprise methods for the application to problems arising in networks, such as routing and load balancing [66.17], and for the application to continuous optimization problems [66.18].

ACO algorithms may be regarded from different perspectives. First of all, as mentioned above, they are SI techniques. However, seen from an operations research perspective, ACO algorithms belong to the class of metaheuristics [66.19–21]. The term *metaheuristic*, first introduced in [66.22], has been derived from the composition of two Greek words. *Heuristic* derives from the verb *heuriskein* ($\epsilon\upsilon\rho\iota\sigma\kappa\epsilon\iota\nu$) which means *to find*, while the prefix *meta* means *beyond, in an upper level*. Before this term was widely adopted, metaheuristics were often called *modern heuristics* [66.23]. In addition to ACO, other algorithms such as evolutionary computation, iterated local search, simulated annealing, and tabu search, are often regarded as metaheuristics. For books and surveys on metaheuristics, we refer the reader to [66.19–21, 23].

Algorithm 66.1 Ant colony optimization (ACO)

```
1: while termination conditions not met do
2:     ScheduleActivities
3:         AntBasedSolutionConstruction()
4:         PheromoneUpdate()
5:         DaemonActions(){optional}
6:     end ScheduleActivities
7: end while
```

From a technical perspective, ACO algorithms work as follows. Given a combinatorial optimization problem to be solved, first a finite set C of the so-called solution components, used for assembling solutions to the problem, must be defined. Second, a set $\mathcal{T}$ of *pheromone values* must be defined. This set of values is commonly called the *pheromone model*, which is – from a mathematical point of view – a parameterized probabilistic model. The pheromone model is one of the central components of any ACO algorithm. The pheromone values $\tau_i \in \mathcal{T}$ are commonly associated with solution components. The pheromone model is used to probabilistically generate solutions to the problem under consideration by assembling them from the set of solution components. In general, ACO algorithms attempt to solve an optimization problem by iterating the following two steps:

- Candidate solutions are constructed using a pheromone model, that is, a parameterized probability distribution over the search space.
- The candidate solutions are used to update the pheromone values in a way that is deemed to bias future sampling toward high-quality solutions.

The pheromone update aims to concentrate the search in regions of the search space containing high-quality solutions. In particular, the reinforcement of solution components depending on the solution quality is an important ingredient of ACO algorithms. It implicitly assumes that good solutions consist of good solution components. To learn which components contribute to good solutions can help assemble them into better solutions. The main steps of any ACO algorithm are shown in Algorithm 66.1. `DaemonActions` (see line 5 of Algorithm 66.1) may include, for example, the application of local search to solutions constructed in function `AntBasedSolutionConstruction()`.

The class of ACO algorithms comprises several variants. Among the most popular ones are $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [66.24] and ant colony system (ACS) [66.25]. For more comprehensive information, we refer the interested reader to [66.26].

66.2.2 Particle Swarm Optimization

PSO [66.2, 27] is an SI technique for optimization that is inspired by the collective behavior of flocks of birds and/or fish schools. The first PSO algorithm was introduced in 1995 by *Kennedy* and *Eberhart* [66.28] for the purpose of optimizing the weights of a neural network, that is, for continuous optimization. In the meantime, PSO has also been adapted for its application to discrete optimization problems [66.29].

In PSO, solutions to the problem under consideration are labeled *particles*. The algorithm works on a whole set of particles at the same time, the so-called *swarm*. Therefore, PSO can be seen as a population-based optimization technique. During the run time of the algorithm, particles move through the search space on the search for an optimal, or good enough, solution. Moreover, particles communicate their current positions to neighboring particles. The position of each particle is updated according to three terms: its so-called *velocity*, the difference between its current position and the best position it has found so far, and that from the best position found by its neighbors. This has the effect that, during the execution of the algorithm, the swarm increasingly focuses on areas of the search space containing high-quality solutions. The term *particle swarm* was chosen by Kennedy and Eberhart for the following reason. Their initial intention was to model the movements of flocks of birds and fish schools. As their model further evolved toward an algorithm for optimization, the visual plots produced from the results of the algorithm rather resembled swarms of mosquitoes. The term *particle* was used due to making use of the term velocity, and *particle* seemed to be the most appropriate term in this context.

PSO is closely related to *artificial life* models. Early works by *Reynolds* on the flocking model known as *boids* [66.30], and *Heppner* and *Grenander's* studies on rules governing large numbers of birds flocking synchronously [66.31], suggested that bird flocking is an emergent behavior resulting from local interactions between the birds. These studies laid the foundation for the development of PSO for solving optimization problems. PSO is – in some way – similar to *cellular automata* (CA), which are often used for generating astonishing self-replicating patterns based on simple local rules. CAs may be characterized by the following three main attributes:

1. Cells are updated in parallel.
2. The value of each new cell depends only on the old values of the cell and its neighbors.
3. There is no difference in rules for updating different cells [66.32].

These three attributes also hold for the particles in PSO.

Henceforth, $\boldsymbol{v}_i$ denotes the velocity of the ith particle in the swarm, $\boldsymbol{x}_i$ denotes its position, $\boldsymbol{p}_i$ denotes the *personal best* position, and $\boldsymbol{p}_g$ is the best position found by particles in its neighborhood. In the original PSO algorithm, $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$, for $i = 1, \ldots, n$, are updated according to the following two equations [66.28]:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + c_1 \boldsymbol{R}_1 \otimes (\boldsymbol{p}_i - \boldsymbol{x}_i) + c_2 \boldsymbol{R}_2 \otimes (\boldsymbol{p}_g - \boldsymbol{x}_i) \,, \tag{66.1}$$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i \,, \tag{66.2}$$

where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are independent functions returning a vector of values, generated uniformly at random, from the range $[0, 1]$. Moreover, c_1 and c_2 are the so-called acceleration coefficients. The symbol $\otimes$ refers to point-wise vector multiplication. As shown in (66.1), the velocity term $\boldsymbol{v}_i$ of a particle is composed of three components: the *momentum*, the *cognitive* and the *social* terms. The *momentum* term $\boldsymbol{v}_i$ carries the particle toward the previous direction; the *cognitive* term,

$$c_1 \boldsymbol{R}_1 \otimes (\boldsymbol{p}_i - \boldsymbol{x}_i) \,,$$

represents a force that pulls the particle toward its personal-best position; finally, the *social* part,

$$c_2 \boldsymbol{R}_2 \otimes (\boldsymbol{p}_g - \boldsymbol{x}_i) \,,$$

represents a force that influences the new direction toward the best position of neighboring particles. Various different neighborhood topologies may be used for this purpose. Examples include ring, star, and *von Neumann*. The use of rather small neighborhood topologies – such as the one induced by the *von Neumann* neighborhood – has generally been shown to lead to better results when rather complex problems are addressed, whereas larger neighborhoods generally lead to a better performance for simpler problems [66.33]. Algorithm 66.2 summarizes the basic PSO algorithm.

Algorithm 66.2 Particle swarm optimization (PSO)

```
1: Randomly generate an initial swarm
2: while termination conditions not met do
3:   for each particle i do
4:     if f(x_i) < f(p_i) then p_i ← x_i
5:     p_g = min(p_neighbors)
6:     Update velocity (66.1)
7:     Update position (66.2)
8:   end for
9: end while
```

The class of PSO algorithms is characterized by a multitude of different variants, rendering it impossible to mention all of them here. However, popular variants include the *Inertia Weight PSO* [66.34], *fully informed PSO* [66.33], and *adaptive hierarchical particle swarm optimizer* [66.35]. Moreover, *Frankenstein's PSO* [66.36] is a PSO variant that was created by analyzing the components of existing PSO variants and combining (some of) them in a beneficial way. For more information, the interested reader may consult [66.37].

66.2.3 Artificial Bee Colony Algorithm

The artificial bee colony (ABC) algorithm was first proposed by *Karaboga* and *Basturk* in 2005 [66.38, 39]. The inspiration for the ABC algorithm is to be found in the foraging behavior of honey bees, which essentially consists of three components: food source positions, amount of nectar and three types of honey bees, that is, *employed bees*, *onlookers*, and *scouts*. In short, the algorithm works as follows. Feasible solutions to the problem under consideration are modeled as *food source positions*. Moreover, the quality of a feasible solution is modeled as the *amount of nectar* present at the corresponding food source position. Each type of bee is responsible for one particular operation in the context of generating new candidate food source positions, that is, new candidate solutions. Specifically, employed bees will search in the vicinity of the food source position that is presently in their memory; meanwhile they pass information about good food source positions to onlooker bees. Onlooker bees tend to select good food source positions from those found by the employed bees, and then further search for better food source positions around the selected food source position. In case the employed bee and the onlookers associated with a food source position are not able to find a better food source position, their current food source position is abandoned and the employed bee associated with this food source becomes a scout bee that performs a search for discovering new food source positions. If a scout identifies a new food source position, it turns into an employed bee again.

Essentially, the difference between the ABC algorithm and other population-based optimization techniques is to be found in the specific way of managing the resources of the algorithm, as suggested by the foraging behavior of honey bees. Due to its simplicity and ease of implementation, the ABC algorithm has captured much attention recently. It should also be mentioned that, although the algorithm has initially been introduced for continuous optimization, in the meantime it has been adapted for its application to combinatorial optimization problems as well [66.40, 41]. For a recent survey, we refer to [66.42].

66.2.4 Other SI Techniques for Optimization and Management Tasks

In the following, we briefly mention other applications of SI techniques for optimization and management tasks, the latter especially for what concerns distributed environments. They are grouped with respect to their natural inspiration.

Division of Labor (Ants/Wasps)

In colonies of ants and wasps, for example, there are various tasks to be dealt with by the colony members. However, the urgency to engage in certain tasks may change over time. In 1984, *Wilson* [66.43] showed that the concept of *division of labor* in colonies of *Pheidole* genus ants allows the colony to adapt to these changing demands. Division of labor was later modeled in [66.44, 45] by means of response threshold models.

These models were later used in several technical applications. In the following, we mention a few of them. *Nouyan* et al. [66.46] consider static and dynamic task allocation problems in which trucks have to be painted in painting booths. Another application concerns media streaming in peer-to-peer networks [66.47]. A multiagent system for the scheduling of dynamic job shops with flexible routing and sequence-dependent setups is considered in [66.48]. *Merkle* et al. [66.49] made use of a response threshold model for self-organized task allocation in the context of computing systems with reconfigurable components. Finally, [66.50] present a system for task allocation in distributed environments.

Cemetery Formation (Ants)

The term *cemetery formation* refers to a behavior which has been observed in ant colonies of the species *Pheidole pallidula*, among others, which cluster the bodies of dead nest mates. This self-organized behavior has given rise to several applications, especially in the context of clustering and sorting. In 1991, a model for the clustering and sorting behavior of ants was published in [66.51]. Note that clustering refers in this context to the formation of piles, and sorting, on the other hand, refers to the spatial arrangement of objects according to their properties.

Mainly based on the model from [66.51], several algorithms for clustering and sorting were proposed in the literature. The first one was presented in [66.52], extending the original model to handle numerical data. More recent papers include [66.53] which deals with clustering and topographic mapping. Finally, the cemetery formation behavior of ants has also inspired an algorithm for dynamic load balancing [66.54].

Flashing in Fireflies

Fireflies are winged beetles that make use of bioluminescence to attract mates or prey. Moreover, tropical fireflies, in particular the ones from Southeast Asia, synchronize their light flashes in large groups of individuals. This is a self-organized phenomenon which is mathematically described by the so-called *phase-coupled oscillator* models [66.55]. The benefits of this self-synchronization are not yet fully understood. Current hypotheses consider diet, social interaction, and altitude.

The literature contains, at least, two types of technical applications that are inspired by different aspects of the flashing of fireflies. First, there are applications that require some type of self-synchronization. Examples include, but are not limited to, a synchronization protocol in sensor networks [66.56], the synchronization in overlay networks [66.57], and dynamic pricing in online markets [66.58]. Second, the literature offers the so-called firefly algorithm (FA) [66.59], which is inspired by the way in which fireflies attract mates or prey. This algorithm was initially introduced for continuous optimization. It has, however, been adapted for the application to combinatorial optimization as well [66.60].

Fish Schooling

A group of fish that have gathered are commonly called an *aggregation of fish*. Such a fish aggregation is called unstructured in the case in which the group consists of various species of fish having randomly gathered, for example, in the vicinity of a food source. If there is some social component to this gathering, the fish are said to be *shoaling*. Shoaling fish are aware of each other's presence, adjusting, for example, their swimming behavior to each other in order to stay together. However, their relation is rather loose. If, in contrast, an aggregation of fish is more tightly organized, for example, when all fish move at the same speed in the same direction, then the aggregation is said to be *schooling*. Schooling is a self-organized behavior that results from local interactions between the fish. This behavior comes with several advantages such as providing a means for social interactions, more successful foraging, and predator avoidance.

There are basically two different algorithms for optimization based on fish schooling to be found in the literature. The first algorithm is referred to as the *artificial fish swarm algorithm* (AFSA). It has, for example, been applied to the training of feed-forward neural networks [66.61], multiuser detection [66.62], image segmentation [66.63], and generally to continuous optimization [66.64]. The second algorithm is known as *fish school search* [66.65].

Self-Desynchronized Croaking (Japanese Tree Frogs)

Different biological studies – for example, [66.66] – have dealt with the croaking of Japanese tree frogs. The male individuals make use of their croaks in order to attract females. Moreover, females of this family of frogs can recognize the source of such a croak and are able to determine the current location of the corresponding male. However, this is only possible if no two frogs (that are close enough to the female) croak at the same time. In such a case, the female is not able to detect where the croaks came from. This is why, over time, male frogs evolved a self-organized way of desynchronizing their croaks. *Aihara* et al. [66.67] introduced a first formal model based on a set of pulse-coupled oscillators for capturing this behavior. So far, this model has only been applied to distributed graph coloring [66.68, 69]. However, the algorithm proposed in [66.69] is currently the state of the art for this problem.

Nest Building (Termites/Wasps)

Both termites and wasps build highly complex nests in cooperation. The construction of such nests is well beyond the capabilities of an individual insect. The nests of both termites and wasps have a very complex internal structure. Moreover, termite nests are extremely large in comparison to individual insects. Scientists studying the nest-building behavior came up with probabilistic models for describing (parts of) the behavior [66.70]. It is nowadays generally accepted that stigmergy plays a central role in nest building.

Models for nest building based on stigmergy have been used mainly in software tools for simulating the automated building of certain structures. Examples can be found in [66.71–74].

66.3 SI in Robotics: Swarm Robotics

Swarm robotics refers to the study and use of SI techniques for the coordination of groups of robots. The following sections provide a brief overview of this field, with a focus on swarm robotic systems and the tasks they accomplish.

66.3.1 Systems

In the late 1940s, *Walter* [66.75] built two autonomous robots called *Machina speculatrix*, or simply tortoise, which exhibited behaviors resembling those of simple animals. The robots had a driving/steering mechanism, a head light, a photoreceptor, and a bump sensor. They were designed to search for and approach light sources of moderate intensity. If a robot observed such a source, its head light was turned off, otherwise it was turned on. In an experiment, the robots were set up in a dark environment, where they approached each other exhibiting complex motion patterns. Such *mutual recognition* allowed *a population of machines* to form *a sort of community*, which broke up once an external light source was introduced [66.75, p. 129]. This two-robot system may be the first self-organizing multirobot system. Interestingly, even a single robot was reported to exhibit complex interactions when facing its mirror image – such a behavior, if *observed in an animal, might be accepted as evidence of some degree of self-awareness* [66.75, pp. 128–129].

In the 1950s, inspired by *von Neumann's* kinematic model of machine replication [66.76], the first physical models of self-replication were built. *Penrose* and *Penrose* [66.77] studied a system in which passive mechanical parts move on a linear track when the latter is subjected to side-to-side agitation. In their default position, the parts do not link under the influence of shaking alone. If a seed object composed of two complementary parts, one hooked up to the other, is added, it replicates by interacting with the other parts on the track. *Jacobson* [66.78] implemented a system in which self-propelled electromechanical parts move on a circular track with several branches. A seed object composed of two parts could trigger other parts to assemble into identical objects without human intervention.

In the late 1980s, studies of *Fukuda* and *Nakagawa* [66.79–81], *Beni* [66.5], and *Wang* and *Beni* [66.82] provided an enormous impetus for the field that developed into *swarm robotics*. *Fukuda* and *Nakagawa* proposed a novel type of robotic system called *dynamically reconfigurable robotic system (DRRS)*, which can *dynamically reorganize its shape and structure [...] for a given task and strategic purpose*. DRRS is made of *several cells* with built-in intelligence and the ability to autonomously connect to and detach from one another [66.81, pp. 55–56]. The authors also presented a first prototype of this system, the CEBOT Mark I [66.80]. At the same time, *Beni* introduced the term *cellular robotic system*, referring to a system that can *encode information as patterns of its own structural units* [66.5, p. 59]; the units would be structural elements, each with built-in intelligence, able to move in space and act asynchronously under distributed control. *Beni* and *Wang* also used the terms *swarm* and *swarm intelligence* in this context [66.83, 84].

Other early physical implementations of distributed robotic systems include the CEBOT Mark II [66.85], ACTRESS [66.86], and GOFER [66.87].

Hardware Architectures

Advances in technology, for example, in computers, manufacturing and mobile devices have made it affordable to study swarms of around 20–1000 physical robots [66.88] and up to around 1 000 000 robots in simulation [66.93–95]. At present, most swarm robotic systems consist of mobile robots that operate on the ground. An example is the Kilobot platform (Fig. 66.2a), which was designed to facilitate the fabrication and operation of thousands of robots – including their charging, programming and activation all at once [66.88]. Other state-of-the-art robotic systems include the r-one (Fig. 66.2b), which features, among others, a set of IR transmitters and receivers for communication and relative localization [66.89], and the Khepera I-IV [66.96] and e-puck [66.97], which feature a range of sensors including a camera. Increasingly, swarm robotic systems operate in spaces other than on the ground, such as underwater [66.90, 98] (Fig. 66.2c) or in the air [66.99, 100]. In some robotic systems, the swarms operate and collaborate across multiple spaces, such as on the ground and in the air [66.91, 101] (Fig. 66.2d,e).

According to their system architecture, most swarm robotic systems can be categorized into either *multirobot* systems or *modular reconfigurable robot* systems. Multirobot systems are composed of multiple distinct robots, which are typically mobile and able to perform (collectively) more than one task in parallel (Fig. 66.2a–c). Modular reconfigurable robot systems are composed of component modules that can be physically linked together to form a robot (Fig. 66.2f). A few *hybrid* systems exist, sharing properties of both multirobot and modular reconfigurable robot systems [66.91, 102–104] (Fig. 66.2d).

Of particular interest among systems of modular reconfigurable robots are those where the robots can build themselves [66.105, 106]. The term *self-reconfigurable* denotes the general ability of physical modules to reconfigure themselves, regardless of whether the process is centrally controlled, for example, by an external computer, or decentralized and autonomous. In the following, we use the term *self-assembly* to refer to processes by which pre-existing components (separate or distinct parts of a disordered structure) autonomously organize

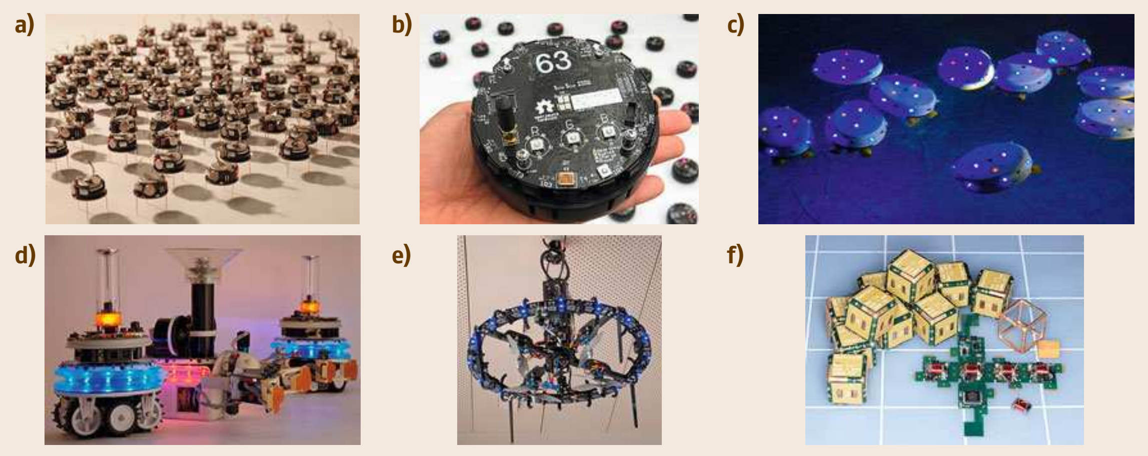

Fig. 66.2a–f Examples of swarm robotic systems: **(a)** Kilobots developed by Harvard University [66.88]; **(b)** r-one (after [66.89], photo courtesy of J. McLurkin, Rice University); **(c)** Lily developed in the CoCoRo project (after [66.90], photo courtesy of T. Schmickl, University of Graz); **(d,e)** a heterogeneous system studied in the Swarmanoid project (after [66.91], photo courtesy of M. Dorigo, Université Libre de Bruxelles); **(f)** Pebbles (after [66.92], photo courtesy of D. Rus, MIT)

into patterns or structures without external intervention. Self-assembly is responsible for the generation of much of the order in nature [66.107] and has widely been applied in the synthesis of products from molecular components. Increasingly, the potential of self-assembly processes involving larger components – up to the centimeter-scale – is being recognized [66.108]. In robotic systems, two distinct classes of self-assembling systems exist [66.109]: (i) systems in which the components that self-assemble are externally propelled, and (ii) systems in which the components that self-assemble are self-propelled.

Sensing and Communication

In most multirobot systems, robots interact with each other by using their sensors or some form of communication. *Dudek* et al. [66.110] presented a detailed taxonomy considering communication range, topology, and bandwidth. In the following, we adopt a simpler categorization proposed by *Cao* et al. [66.111]:

- *Interaction via environment* refers to the transfer of information that is mediated through the *memory* of the environment. In this case, robots leave persistent signs that stimulate the activity of other robots. This kind of indirect communication is also referred to as *stigmergy* [66.16]. Stigmergic communication is widely used in social insect societies, for example, during the construction of mounds by termites of *Macrotermes bellicosus* [66.8], and has been implemented in several swarm robotic systems [66.112–116].
- *Interaction via sensing refers to local interactions that occur between agents as a result of agents sensing one another, but without explicit communication* [66.111, p. 12]. We include in this category interactions where agents sense each other indirectly, that is, where the current presence or motion of another agent can be inferred from changes in the environment. Note that the boundary to stigmergic communication is blurred; for example, consider the situation where multiple agents push an object simultaneously [66.117–119].

In some social animals, the members of a group observe a common leader individual. Their actions can be highly dependent on the observed behavior of the leader, as, for instance, during an attack of the group [66.120]. In other animals, no recognizable leader individual exists; instead, individuals observe nearby group members. The latter situation is typical for swarm systems. It is reported, for instance, for animal groups that exhibit herding, flocking, and schooling behavior [66.8]. Note that where the groups are not homogeneous, even a minority of individuals may be able to influence the rest of the group [66.121].

In principle, interaction via sensing can be considered an implicit form of communication, in particular, as an observed agent can change action and thereby influence the behavior of its observers. *Arkin* [66.122] referred to the interaction via sensing category as *cooperation without communication*, and showed that it is sufficient to accomplish tasks, that require the cooperation of multiple robots. Other examples of swarm robotic studies using interaction via sensing include [66.123–126].

- *Interaction via communication* refers to interactions involving explicit communication. Thereby, information is either broadcast or transferred to specific teammates. Information transfer can take place through direct physical interactions, such as touch. This latter form of communication can also be referred to as *direct interaction* [66.127]. Explicit communication can improve the performance of a multirobot system. This is typically the case where the system benefits from robots being recruited to certain areas of the environment. *Balch* and *Arkin* [66.128] studied such an environment and showed that it can be sufficient for each robot to signal its overall state. The transfer of more elaborate information however would not result in any significant increase in task performance. Explicit communication is commonly used in modular reconfigurable robot systems, for example, to exchange information between inter-connected modules or to support the docking process of separate modules [66.129].

Control and Coordination

Over the last two decades, a range of design methods have been proposed for the control of swarm robotic systems. They can be broadly classified into behavior-based design methods and automated design methods [66.130].

In behavior-based design methods, the user approaches the problem in a bottom-up manner [66.131]. A repertoire of behaviors for individual robots is defined and often refined through a trial-and-error process. A common approach is the use of finite state machines. Each state defines a basic behavior. Transitions between states can be triggered by probability, external events, time-outs, and combinations of these [66.132–134]. A prominent example is the use of response threshold functions, for example,

$$1 - \exp^{-s_i/\theta_i} \quad \text{or} \quad \frac{s_i^2}{s_i^2 + \theta_i^2} ,$$

which define the probability for an individual to engage in task i based on the perceived task stimulus s_i and threshold θ_i. The particular threshold value θ_i can either be fixed for each individual from the outset [66.135] or learned during its lifetime [66.136, 137]. In both cases, the mechanism can facilitate the emergent allocation of tasks in groups of otherwise identical individuals (see also Sect. 66.2.4). In addition, *intentional* approaches to task allocation have been considered [66.138, 139]. These require the agents to cooperate explicitly with each other. For example, the decentralized ALLIANCE algorithm [66.140, 141] can be used for groups of heterogenous robots to perform tasks and subtasks, which may have ordering dependencies, in a fault-tolerant way. It assumes that the robots detect with some probability the effect of their own actions as well as the actions of other team members.

Virtual potential fields [66.142, 143], and physicomimetics [66.144], is another widely used behavior-based design method. The robots mimic a physical particle under the influence of a potential field. The latter guides the particle toward a point of minimal potential energy. While the goal point, which the robot shall reach, would exert an attractive force on the particle, any obstacle would exert a repulsive force. Other robots can exert forces on the particle as well. Using this concept, a wide repertoire of behaviors can be realized, such as the collective movement of robots arranged in particular formations [66.145], or the tracking of multiple moving targets [66.146]. The properties of the resulting swarm systems, for example, the cohesion of the swarm, can also be formally analyzed [66.147].

Other design methods include the Growing Point Language [66.148], the Origami Shape Language [66.149], and Proto [66.150]. These languages were developed in the context of Amorphous Computing [66.151], which considers systems of massively distributed, disordered, asynchronous, and locally interacting computational devices. The Proto language has been extended for use on mobile devices. This extension was validated with a swarm of 40 iRobot robots [66.152]. Some amorphous computing approaches allow users to specify desired global system properties in the language. A compiler then produces the local rule set for the agents to achieve these properties [66.149].

Automated design methods can be grouped into reinforcement learning and evolutionary robotics. In reinforcement learning [66.153], an agent interacts with its environment by choosing actions and receiving rewards. *Matarić* [66.154, 155] applied reinforcement learning in a swarm robotic context. The robots had to learn how to collaborate in a foraging task. The robots were provided with a set of hand-coded behaviors (as in a behavior-based approach) and were required to learn how *to correlate appropriate conditions for each of these behaviors in order to optimize the higher-level behavior* [66.155]. The difficulties of using reinforcement learning in a swarm robotic context are discussed in [66.130]. A recent survey of reinforcement learning in robotics is reported in [66.156].

Evolutionary robotics is an approach to designing robots, or aspects of them (e.g., morphology, control) using evolutionary algorithms [66.157, 158]. This approach can also be applied to the design of swarm robotic systems. In principle, evolution can bypass both the problem of decomposing a given task and the problem of identifying basic behaviors that achieve the subtasks [66.159, 160]. Early studies in evolutionary robotics developed collective behavior such as herding or flocking in simplistic simulation environments [66.161–163]. Simulation environments with physically embodied agents were considered in [66.159], where neural network controllers for aggregation were first evolved using a group of five robots in a simple simulation environment; the best of these controllers were subsequently validated using a more detailed simulation model of the robots. *Quinn* et al. [66.164] evolved neural network controllers for collective motion using a group of three simulated robots and subsequently tested the best-rated network in 100 trials with a group of three physical robots. *Watson* et al. [66.165] went a step further in that controllers for a simple phototaxis task were directly evolved on a group of eight physical robots. Working toward a distributed evolution of robot morphologies in hardware, *Griffith* et al. [66.166] demonstrated a system of template-replicating polymers, which were made of reconfigurable modules that slid passively on an air table and executed a finite state machine to control their connectivity. Recent work on evolutionary swarm robotics considers cultural evolution, for example, where behaviors that can be imitated (memes) are subject to an evolutionary process. In these, the robots engage as both teachers and learners to exchange memes [66.167].

Several design methods were developed specifically for, or mainly adopted in, the context of modular reconfigurable robot systems. One class of algorithms addresses the problem of how to adjust the relative positions of modules without changing their connection topology. *Yim* [66.168] proposed the use of *gait control tables* to produce a range of animal-like locomotion patterns, such as the walking gaits of hexapods. Each gait control table specifies for each control cycle and module a basic action to be performed. The controller is executed either from a central place or in a distributed fashion. In the latter case, the modules synchronize their actions using internal timers. *Shen* et al. [66.169] proposed *hormone-inspired* communication and control, in which artificial hormones help modules to synchronize actions and discover changes in their topology. For example, a set of independent caterpillar-like robots could be connected into a single entity, which would adapt its gait to the new topology. In a similar experiment, a connected entity was manually split into smaller entities that continued to move as independent caterpillars. *Støy* [66.170] proposed a *role-based control* algorithm to let modular robots display periodic locomotion patterns. A module's role specifies its actions and how to synchronize them with neighbor modules. For communication, a parent–child architecture is used; thus, modules need to be arranged in acyclic graphs. An extended version of the control algorithm can also cope with cycles.

Another class of algorithms addresses the problem of how to adjust the relative positions of modules by changing the connection topology [66.106]. One approach is to formulate the problem as a search problem. For example, in order to reconfigure a lattice-based robot from one topology to another, a graph search is performed, where the start node of the graph corresponds to the initial topology of the robot and the end node corresponds to the desired topology of the robot [66.171]. Due to the combinatorial explosion of possibilities, an exhaustive search of such graphs is impractical whenever the number of modules is not small. State-of-the-art approaches are thus heuristic and consider ways of reducing the problem complexity. For example, *Yoshida* et al. [66.172] proposed a two-level *motion planner*. A global planner ensures that the robot as a whole follows a predefined 3D trajectory. To do so, it specifies several candidate paths that bring individual modules from the tail to the head of the robot. A motion scheme selector chooses a feasible path for each module based on a rule database. Another example is to merge logically a group of nearby modules into *meta-modules*, which, typically, have more advanced locomotion abilities than the individual modules. The problem is then reduced to developing controllers for both meta-modules and modular robots composed of meta-modules [66.173]. In principle, modular robots can solve the search problem on the fly [66.174]. Other than by search, the reconfiguration problem can also be attempted by local movement strategies, for example, random walks [66.175, 176], cellular automata rules [66.177], gradient rules [66.178, 179], or combinations of these [66.180]. These approaches naturally lead to decentralized implementations, as is desired in swarm robotics.

66.3.2 Tasks

A range of capabilities have already been demonstrated with swarm robotic systems. In the following, a brief overview is given. More detailed information is provided in Chaps. 71–74 of Part F of this handbook. *Garnier* et al. [66.189] demonstrated how a group of 20 Alice robots *aggregate* in a homogeneous environment. The robots mimic the aggregation behavior of cockroaches, which are reported to join and leave clusters with probabilities that depend on the sizes of clusters [66.190]. Such probabilistic algorithms have the advantage that, as long as the environment is bounded, it is not required that the robots initially form a connected graph in terms of their sensing and/or communication. A deterministic algorithm for aggregation is considered in [66.181]. It requires robots to have one binary sensor, which informs them whether or not there is another robot in their line of sight. The robots do not need memory and do not need to perform arithmetic computation. They rotate on the spot when they perceive another robot, and move backward along a circular trajectory otherwise. This algorithm was validated with groups of 40 e-puck robots (Fig. 66.3a).

Werfel et al. [66.116] developed a system of robots that can simultaneously *construct* and navigate structures from a supply of building blocks (Fig. 66.3b). The robots are inspired by termites, which use stigmergic rules to construct sophisticated structures, in particular, the mounds they inhabit. Given a desired target structure, it is possible to generate automatically a set of rules to be uploaded onto each robot. Using only local information, these rules allow the robots to coordinate their activities in a way that avoids conflict. A group of three robots constructed several structures, one resembling a castle.

Halloy et al. [66.182] showed that hybrid societies comprising both cockroaches and robots can collectively *decide* to aggregate under either of two shelters (Fig. 66.3c) and that it is possible for the robots to influence the decision-making process. In general, such interactive robots could be used to study and control animal groups [66.182, 191], including livestock [66.192, 193], and to inform ecological conservation policy.

Following the pioneering simulation works on *boids* [66.30], *Turgut* et al. [66.183] demonstrated how a group of robots can *flock* through a real environment using simple rules. To align with each other, the robots used *virtual heading sensors*, each comprising a digital compass and a wireless communication module. Flocking was demonstrated with 9 Kobot robots in a bounded environment (Fig. 66.3d).

Krieger et al. [66.184] studied algorithms that allow a group of robots to *forage* (Fig. 66.3e). The robots rested in a central place, the nest. A robot would leave the nest if the total energy of the colony dropped below a threshold. Each robot had its own threshold, which effectively enabled the division of labor within the group. In addition, a robot would leave the nest when being recruited by another robot that had found a cluster of food. The pair of robots would then perform a tandem run to reach the cluster. The algorithms were tested on groups of up to 12 Khepera robots. The groups were reported to perform more efficiently when employing the division of labor and recruitment mechanisms than without such mechanisms.

Groß et al. demonstrated how a group of 16 s-bot robots *self-assemble* into a single composite entity [66.185]. The process was seeded by one of the robots activating its light emitting diode (LED) ring in red. Other robots activated their LED rings in blue. Once a robot would connect to the seed structure, it became red too, thereby attracting other robots to the structure as it grows (Fig. 66.3f). The problem of self-assembling into arbitrary morphologies of s-bot robots was considered in [66.194].

Holland and *Melhuish* [66.186] studied algorithms that allow groups of robots to *sort* (and cluster) objects of different types (Fig. 66.3g). Six robots were programmed using simple rules, which regulated the conditions under which objects of different types were picked up and deposited.

Following the pioneering work of *Kube* et al. [66.195, 196], *Chen* et al. [66.187] proposed an algorithm for a group of robots to *transport* objects larger than themselves toward a goal location (Fig. 66.3h). The robots were programmed to only push the object across the portion of its surface where the direct line of sight to the goal is occluded by the object. The algorithm was proven to work for objects of arbitrary convex shape and it was tested with 20 e-puck robots.

Ijspeert et al. [66.188] studied an algorithm that allows a group of robots to *pull sticks* out of the ground collaboratively (Fig. 66.3i). Upon encountering a stick,

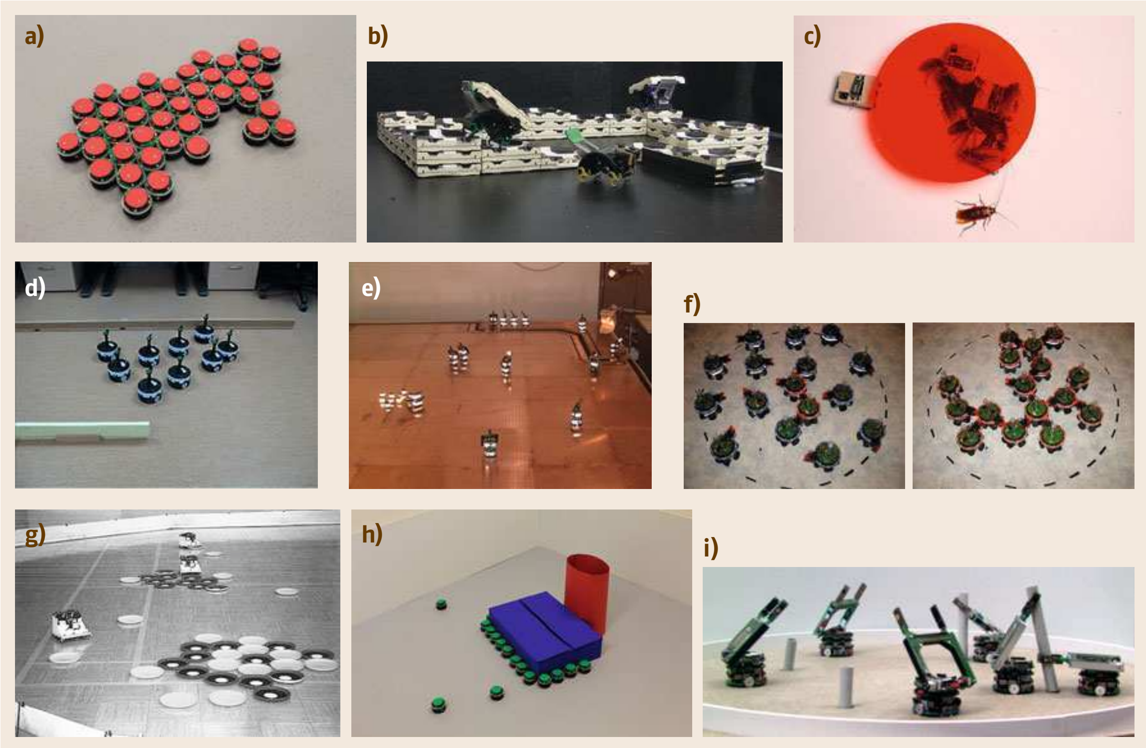

Fig. 66.3a–i Examples of capabilities demonstrated by swarm robotic systems: **(a)** aggregation (after [66.181]); **(b)** construction (after [66.116]; reprinted with permission from AAAS); **(c)** decision making (after [66.182]; photo courtesy of J. Halloy, Université Libre de Bruxelles); **(d)** flocking (after [66.183]; photo courtesy of E. Şahin, Middle East Technical University); **(e)** foraging (after [66.184]; photo courtesy of L. Keller, University of Lausanne); **(f)** self-assembly (after [66.185]); **(g)** sorting of objects (after [66.186]; photo courtesy of C. Melhuish, Bristol Robotics Laboratory); **(h)** transport of objects (after [66.187]); **(i)** pulling sticks out of the ground (after [66.188]; reprinted with permission from Springer)

a robot would only be able to pull it partially out of the ground. It would then wait for a second robot to arrive and pull the stick out completely. The optimal waiting time for the first robot was derived from an analytic model of the system. The algorithm was validated using a system of six Khepera robots.

66.4 Research Challenges

Research challenges concerning the use of swarm intelligence in optimization are mainly related to increasing their efficiency. More specifically, in addition to providing an innovative way of problem solving, swarm intelligence approaches must also be efficient concerning, for example, computation time in order to be able to compete with state-of-the-art optimization techniques. This may often be achieved by hybridizing swarm intelligence approaches with components taken from optimization algorithms in other fields such as, for example, operations research. The interested reader may find various references to such kind of techniques in [66.197].

With regard to swarm robotics, a major challenge is the transition from systems operating in structured indoor environments, as typically found in laboratories, to the more complex environments found in the real world. Over the next decades, swarms of robots are expected to have impact in a range of application scenarios, including cognitive factories, deep sea exploration, disaster management, precision farming, and space systems. Working toward more complex environments also concerns the ability of swarms of robots to interact safely with humans. Another challenge concerns the miniaturization of swarm robotic systems. Most of the current systems comprise of centimeter-sized robots. The swarm robotics approach, however, should be equally applicable to intelligent autonomous devices operating at scales from a millimeter down to a micrometer. This could have profound implications, for example, on advanced materials and healthcare technologies.

References

66.1 E. Bonabeau, M. Dorigo, G. Theraulaz: *Swarm Intelligence: From Natural to Artificial Systems* (Oxford Univ. Press, New York 1999)

66.2 J. Kennedy, R.C. Eberhart, Y. Shi: *Swarm Intelligence* (Morgan Kaufmann, San Francisco 2001)

66.3 C. Blum, D. Merkle (Eds.): *Swarm Intelligence: Introduction and Applications* (Springer, Berlin, Heidelberg 2008)

66.4 S.J. Russell, P. Norvig: *Artificial Intelligence. A Modern Approach* (Simon Schuster Co., Englewood Cliffs 1995)

66.5 G. Beni: The concept of cellular robotic systems, Proc. 3rd IEEE Int. Symp. Intell. Syst., Piscataway (1988) pp. 57–62

66.6 G. Beni, J. Wang: Swarm intelligence, Proc. 7th Annu. Meet. Robot. Soc. Japan, RSJ, Tokyo (1989) pp. 425–428

66.7 G. Beni, S. Hackwood: Stationary waves in cyclic swarms, Proc. 1992 IEEE Int. Symp. Intell. Control, Los Alamitos (1992) pp. 234–242

66.8 S. Camazine, J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau: *Self-Organization in Biological Systems* (Princeton Univ. Press, New Jersey 2001)

66.9 I. Kant: *Critique of Judgement* (Hackett, Indianapolis 1987), Translated by W. S. Pluhar

66.10 B. Hölldobler, E.O. Wilson (Eds.): *The Ants* (Springer, Berlin, Heidlberg 1990)

66.11 M. Dorigo, T. Stützle: *Ant Colony Optimization* (MIT, Cambridge 2004)

66.12 M. Dorigo: Optimization, Learning and Natural Algorithms, Ph.D. Thesis (Dipartimento di Elettronica, Politecnico di Milano, Italy 1992), in Italian

66.13 M. Dorigo, V. Maniezzo, A. Colorni: Positive feedback as a search strategy, Tech. Rep. 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991

66.14 M. Dorigo, V. Maniezzo, A. Colorni: Ant System: Optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B **26**(1), 29–41 (1996)

66.15 J.-L. Deneubourg, S. Aron, S. Goss, J.-M. Pasteels: The self-organizing exploratory pattern of the argentine ant, J. Insect Behav. **3**, 159–168 (1990)

66.16 P.-P. Grassé: La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs, Insectes Soc. **6**(1), 41–80 (1959), in French

66.17 G. Di Caro, M. Dorigo: AntNet: Distributed stigmergetic control for communications networks, J. Artif. Intell. Res. **9**, 317–365 (1998)

66.18 K. Socha: ACO for continuous and mixed-variable optimization, Lect. Notes Comput. Sci. **3172**, 25–36 (2004)

66.19 C. Blum, A. Roli: Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Comput. Surv. **35**(3), 268–308 (2003)

66.20 F. Glover, G. Kochenberger (Eds.): *Handbook of Metaheuristics* (Kluwer, Boston 2002)

66.21 H.H. Hoos, T. Stützle: *Stochastic Local Search: Foundations and Applications* (Elsevier, Amsterdam 2004)

66.22 F. Glover: Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. **13**, 533–549 (1986)

66.23 C.R. Reeves (Ed.): *Modern Heuristic Techniques for Combinatorial Problems* (Wiley, New York 1993)

66.24 T. Stützle, H.H. Hoos: $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System, Futur. Gener. Comput. Syst. **16**(8), 889–914 (2000)

66.25 M. Dorigo, L.M. Gambardella: Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)

66.26 M. Dorigo, T. Stützle: Ant colony optimization: Overview and recent advances. In: *Handbook of Metaheuristics*, ed. by M. Gendreau, J.-Y. Potrin (Springer, Berlin, Heidelberg 2010) pp. 227–264

66.27 M. Clerc (Ed.): *Particle Swarm Optimization* (ISTE, Newport Beach 2006)

66.28 J. Kennedy, R.C. Eberhart: Particle swarm optimization, Proc. 1995 IEEE Int. Conf. Neural Netw., Piscataway, Vol. 4 (1995) pp. 1942–1948

66.29 Q.-K. Pan, M. Fatih Tasgetiren, Y.-C. Liang: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, Comput. Oper. Res. **35**(9), 2807–2839 (2008)

66.30 C.W. Reynolds: Flocks, herds and schools: A distributed behavioral model, Comput. Graph. **21**(4), 25–34 (1987)

66.31 F. Heppner, U. Grenander: A stochastic nonlinear model for coordinated bird flocks. In: *The Ubiq-*

uity of Chaos, ed. by S. Krasner (AAAS, Washington DC 1990)
66.32 R. Rucker: *Seek!* (Four Walls Eight Windows, New York 1999)
66.33 R. Mendes, J. Kennedy, J. Neves: The fully informed particle swarm: Simpler, maybe better, IEEE Trans. Evol. Comput. **8**(3), 204–210 (2004)
66.34 Y. Shi, R. Eberhart: A modified particle swarm optimizer, Proc. 1998 IEEE World Congr. Comput. Intell. (1998) pp. 69–73
66.35 S. Janson, M. Middendorf: A hierarchical particle swarm optimizer and its adaptive variant, IEEE Trans. Syst. Man Cybern. Part B Cybern. **35**(6), 1272–1282 (2005)
66.36 M.A. de Montes Oca, T. Stützle, M. Birattari, M. Dorigo: Frankenstein's PSO: A composite particle swarm optimization algorithm, IEEE Trans. Evol. Comput. **13**(5), 1120–1132 (2009)
66.37 M. Dorigo, M.A. de Montes Oca, A. Engelbrecht: Particle swarm optimization, Scholarpedia **3**(11), 1486 (2008)
66.38 D. Karaboga, B. Basturk: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. **39**(3), 459–471 (2007)
66.39 D. Karaboga, B. Basturk: On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. **8**(1), 687–697 (2008)
66.40 Q.-K. Pan, M.F. Tasgetiren, P.N. Suganthan, T.J. Chua: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, Inf. Sci. **181**(12), 2455–2468 (2011)
66.41 F.J. Rodriguez, C. García-Martínez, C. Blum, M. Lozano: An artificial bee colony algorithm for the unrelated parallel machines scheduling problem, Lect. Notes Comput. Sci. **7492**, 143–152 (2012)
66.42 D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga: A comprehensive survey: Artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. **42**, 21–57 (2014)
66.43 E.O. Wilson: The relation between caste ratios and division of labour in the ant genus *phedoile*, Behav. Ecol. Sociobiol. **16**(1), 89–98 (1984)
66.44 G. Theraulaz, E. Bonabeau, J.-L. Deneubourg: Response threshold reinforcement and division of labour in insect societies, Proc. Biol. Sci. **265**(1393), 327–332 (1998)
66.45 E. Bonabeau, G. Theraulaz, J.-L. Deneubourg: Fixed response thresholds and the regulation of division of labor in social societies, Bull. Math. Biol. **60**, 753–807 (1998)
66.46 S. Nouyan, R. Ghizzioli, M. Birattari, M. Dorigo: An insect-based algorithm for the dynamic task allocation problem, Künstl. Intell. **4**, 25–31 (2005)
66.47 M. Sasabe, N. Wakamiya, M. Murata, H. Miyahara: Effective methods for scalable and continuous media streaming on peer-to-peer networks, Eur. Trans. Telecommun. **15**, 549–558 (2004)
66.48 X. Yu, B. Ram: Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups, Int. J. Prod. Res. **44**(22), 4793–4813 (2006)
66.49 D. Merkle, M. Middendorf, A. Scheidler: Self-organized task allocation for computing systems with reconfigurable components, Proc. 20th Int. Parallel Distrib. Proc. Symp., IPDPS 2006 (2006) p. 8
66.50 R. Klazar, A.P. Engelbrecht: Dynamic load balancing inspired by division of labour in ant colonies, Proc. 2011 IEEE Symp. Swarm Intell., SIS (2011) pp. 1–8
66.51 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, Proc. 1st Int. Conf. Simul. Adapt. Behav.: From Animals to Animats 1, SAB 91 (MIT, Cambridge 1991) pp. 356–365
66.52 E.D. Lumer, B. Faieta: Diversity and adaptation in populations of clustering ants, Proc. 3rd Int. Conf. Simul. Adapt. Behav.: From Animals to Animats 3, SAB 94, MIT Diversity, ed. by D. Cliff, P. Husbands, J.-A. Meyer, S.W. Wilson (1994) pp. 501–508
66.53 J. Handl, J. Knowles, M. Dorigo: Ant-based clustering and topographic mapping, Artif. Life **12**(1), 35–62 (2006)
66.54 R. Klazar, A.P. Engelbrecht: Dynamic load balancing inspired by cemetery formation in ant colonies, Lect. Notes Comput. Sci. **7461**, 236–243 (2012)
66.55 R.E. Mirollo, S.H. Strogatz: Synchronization of pulse-coupled biological oscillators, SIAM J. Appl. Math. **50**(6), 1645–1662 (1990)
66.56 Y.-W. Hong, A. Scaglione: A scalable synchronization protocol for large scale sensor networks and its applications, IEEE J. Sel. Areas Commun. **23**(5), 1085–1099 (2005)
66.57 O. Babaoglu, T. Binci, M. Jelasity, A. Montresor: Firefly-inspired heartbeat synchronization in overlay networks, Proc. SASO 2007 – 1st Int. Conf. Self-Adapt. Self-Organ. Syst. (2007) pp. 77–86
66.58 J. Jumadinova, P. Dasgupta: Firefly-inspired synchronization for improved dynamic pricing in online markets, Proc. SASO 2008 – 2nd IEEE Int. Conf. Self-Adapt. Self-Organ. Syst. (2008) pp. 403–412
66.59 X.S. Yang: *Nature Inspired Metaheuristic Algorithms* (Luniver, UK 2010)
66.60 G.K. Jati, S. Suyanto: Evolutionary discrete firefly algorithm for travelling salesman problem, Lect. Notes Comput. Sci. **6943**, 393–403 (2011)
66.61 C.-R. Wang, C.-L. Zhou, J.-W. Ma: An improved artificial fish-swarm algorithm and its application in feed-forward neural networks, Proc. 2005 Int. Conf. Mach. Learn. Cybern., Vol. 5 (2005) pp. 2890–2894
66.62 M. Jiang, Y. Wang, S. Pfletschinger, M.A. Lagunas, D. Yuan: Optimal multiuser detection with artificial fish swarm algorithm. In: *Advanced Intelli-*

gent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques, Communications in Computer and Information Science, Vol. 2, ed. by D.-S. Huang, L. Heutte, M. Loog (Springer, Berlin, Heidelberg 2007) pp. 1084–1093

66.63 M. Jiang, N.E. Mastorakis, D. Yuan, M.A. Lagunas: Image segmentation with improved artificial fish swarm algorithm, Proc. Eur. Comput. Conf., Lect. Notes Electr. Eng., Vol. 28, ed. by N. Mastorakis, V. Mladenov, V.T. Kontargyri (2009) pp. 133–138

66.64 A.M.A.C. Rocha, T.F.M.C. Martins, E.M.G.P. Fernandes: An augmented Lagrangian fish swarm based method for global optimization, J. Comput. Appl. Math. **235**(16), 4611–4620 (2011)

66.65 C.J.A.B. Filho, F.B. de Lima Neto, A.J.C.C. Lins, A.I.S. Nascimento, M.P. Lima: A novel search algorithm based on fish school behavior, Proc. SMC 2008 – IEEE Int. Conf. Syst. Man Cybern. (2008) pp. 2646–2651

66.66 K.D. Wells: The social behaviour of anuran amphibians, Anim. Behav. **25**, 666–693 (1977)

66.67 I. Aihara, H. Kitahata, K. Yoshikawa, K. Aihara: Mathematical modeling of frogs' calling behavior and its possible application to artificial life and robotics, Artif. Life Robot. **12**(1), 29–32 (2008)

66.68 S.A. Lee, R. Lister: Experiments in the dynamics of phase coupled oscillators when applied to graph coloring, ACSC 2008 – Proc. 31st Australas. Conf. Comput. Sci., Darlinghurst (2008) pp. 83–89

66.69 H. Hernández, C. Blum: Distributed graph coloring: An approach based on the calling behavior of Japanese tree frogs, Swarm Intell. **6**, 117–150 (2012)

66.70 E. Boneabeau, G. Theraulaz, J.-L. Deneubourg, N.-R. Franks, O. Rafelsberger, J.L. Joly, S. Blanco: A model for the emergence of pillars, walls and royal chambers in termite nests, Philos. Trans. R. Soc. **353**(1375), 1561–1576 (1997)

66.71 Z. Mason: Programming with stigmergy: Using swarms for construction, Proc. Artif. Life VIII – 8th Int. Conf. Artif. Life (2003) pp. 371–374

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

66.73 A. Grushin, J.A. Reggia: Stigmergic self-assembly of prespecified artificial structures in a constrained and continuous environment, J. Integr. Comput.-Aided Eng. **13**(4), 289–312 (2006)

66.74 E. Bonabeau, S. Guerin, D. Snyers, P. Kuntz, G. Theraulaz: Three-dimensional architectures grown by simple *stigmergic* agents, Biosystems **56**(1), 13–32 (2000)

66.75 W.G. Walter: *The Living Brain* (W. W. Norton, New York 1953)

66.76 J. von Neumann: The general and logical theory of automata, Cerebral Mechanisms in Behavior: The Hixon Symposium, ed. by L.A. Jeffress (Wiley, New York 1951) pp. 1–41

66.77 L.S. Penrose, R. Penrose: A self-reproducing analogue, Nature **179**(4571), 1183 (1957)

66.78 H. Jacobson: On models of reproduction, Am. Sci. **46**, 255–284 (1958)

66.79 T. Fukuda, S. Nakagawa: A dynamically reconfigurable robotic system (concept of a system and optimal configurations), Proc. 1987 IEEE Int. Conf. Ind. Electron. Control Instrum., Piscataway (1987) pp. 588–595

66.80 T. Fukuda, S. Nakagawa: Dynamically reconfigurable robotic system, Proc. 1988 IEEE Int. Conf. Robot. Autom., Piscataway, Vol. 3 (1988) pp. 1581–1586

66.81 T. Fukuda, S. Nakagawa: Approach to the dynamically reconfigurable robotic system, J. Intell. Robot. Syst. **1**(1), 55–72 (1988)

66.82 J. Wang, G. Beni: Pattern generation in cellular robotic systems, Proc. 3rd IEEE Int. Symp. Intell. Control (IEEE, Piscataway 1988) pp. 63–69

66.83 G. Beni, J. Wang: Swarm intelligence in cellular robotic systems, Proc. NATO Adv. Workshop Robot. Biol. Syst., Il Ciocco (1989) pp. 703–712

66.84 G. Beni, J. Wang: Swarm intelligence, Proc. 7th Annu. Meet. Robot. Soc. Japan, RSJ, Tokyo (1989) pp. 425–428, in Japanese

66.85 T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss: Self organizing robots based on cell structures – CEBOT, Proc. 1988 IEEE Int. Workshop Intell. Robot., Piscataway (1988) pp. 145–150

66.86 H. Asama, A. Matsumoto, Y. Ishida: Design of an autonomous and distributed robot system: ACTRESS, Proc. 1989 IEEE/RSJ Int. Workshop Intell. Robot. Syst., Piscataway (1989) pp. 283–290

66.87 P. Caloud, W. Choi, J.-C. Latombe, C. Le Pape, M. Yim: Indoor automation with many mobile robots, Proc. 1990 IEEE Int. Workshop Intell. Robot. Syst., Piscataway, Vol. 1 (1990) pp. 67–72

66.88 M. Rubenstein, A. Cornejo, R. Nagpal: Programmable self-assembly in a thousand-robot swarm, Science **345**, 795–799 (2014)

66.89 J. McLurkin, A.J. Lynch, S. Rixner, T.W. Barr, A. Chou, K. Foster, S. Bilstein: A low-cost multi-robot system for research, teaching, and outreach, Proc. 10th Int. Symp. Distrib. Auton. Robot. Syst. (DARS 2010), ed. by A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. Ani Hsieh, L.E. Parkes, K. Støy (2010) pp. 597–609

66.90 T. Schmickl, R. Thenius, C. Möslinger, J. Timmis, A. Tyrrell, M. Read, J. Hilder, J. Halloy, A. Campo, C. Stefanini, L. Manfredi, S. Orofino, S. Kernbach, T. Dipper, D. Sutantyo: Cocoro: The self-aware underwater swarm, 2011 Firth IEEE Conf. Self-Adapt. Self-Organ. Syst. Workshop, SASOW (2011) pp. 120–126

66.91 M. Dorigo, D. Floreano, L.M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A.L. Christensen, A. Decugniére, G.A. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi,

V. Longchamp, S. Magnenat, J. Martinez Gonzalez, N. Mathews, M.A. de Montes Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stuetzle, V. Trianni, E. Tuci, A.E. Turgut, F. Vaussard: Swarmanoid: A novel concept for the study of heterogeneous robotic swarms, IEEE Robot. Autom. Mag. **20**(4), 60–71 (2013)

66.92 K. Gilpin, A. Knaian, D. Rus: Robot pebbles: One centimeter modules for programmable matter through self-disassembly, IEEE Int. Conf. Robot. Auton., ICRA (2010) pp. 2485–2492

66.93 R. Vaughan: Massively multi-robot simulation in stage, Swarm Intell. **2**(2–4), 189–208 (2008)

66.94 R. Fitch, Z. Butler: Million module march: Scalable locomotion for large self-reconfiguring robots, Int. J. Robot. Res. **27**(3–4), 331–343 (2008)

66.95 M.P. Ashley-Rollman, P. Pillai, M.L. Goodstein: Simulating multi-million-robot ensembles, Proc. 2011 IEEE Int. Conf. Robotic. Autom. (2011) pp. 1006–1013

66.96 F. Mondada, E. Franzi, A. Guignard: The Development of Khepera, Exp. Mini-Robot Khepera, Proc. 1st Int. Khepera Workshop (HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut 1999) pp. 7–14

66.97 F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli: The e-puck, a robot designed for education in engineering, Proc. 9th Conf. Mobile Robot. Compet., ROBOTICA 2009, Castelo Branco (2009) pp. 59–65

66.98 N. Kottege, U.R. Zimmer: Underwater acoustic localization for small submersibles, J. Field Robot. **28**(1), 40–69 (2011)

66.99 J.F. Roberts, T. Stirling, J.-C. Zufferey, D. Floreano: 3-D relative positioning sensor for indoor flying robots, Auton. Robot. **33**(1–2), 5–20 (2012)

66.100 A. Kushleyev, D. Mellinger, C. Powers, V. Kumar: Towards a swarm of agile micro quadrotors, Auton. Robot. **35**(4), 287–300 (2013)

66.101 L. Chaimowicz, V. Kumar: Aerial shepherds: Coordination among UAVs and swarms of robots, 7th Int. Symp. Distrib. Auton. Robot. Syst. (2004) pp. 23–25

66.102 F. Mondada, L.M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, M. Dorigo: The cooperation of swarm-bots: Physical interactions in collective robotics, IEEE Robot. Autom. Mag. **12**(2), 21–28 (2005)

66.103 S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, P. Corradi, L. Ricotti: Symbiotic robot organisms: REPLICATOR and SYMBRION projects, Proc. 8th Workshop Perform. Metr. Intell. Syst. (2008) pp. 62–69

66.104 H. Wei, Y. Chen, J. Tan, T. Wang: Sambot: A self-assembly modular robot system, IEEE/ASME Trans. Mechatr. **16**(4), 745–757 (2011)

66.105 M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, G.S. Chirikjian: Modular self-reconfigurable robot systems, IEEE Robot. Autom. Mag. **14**(1), 43–52 (2007)

66.106 K. Støy, D. Brandt, D.J. Christensen: *Self-Reconfigurable Robots: An Introduction* (MIT, Cambridge 2010)

66.107 G.M. Whitesides, B. Grzybowski: Self-assembly at all scales, Science **295**(5564), 2418–2421 (2002)

66.108 G.M. Whitesides, M. Boncheva: Beyond molecules: Self-assembly of mesoscopic and macroscopic components, Proc. Natl. Acad. Sci. USA **99**(8), 4769–4774 (2002)

66.109 R. Groß, M. Dorigo: Self-assembly at the macroscopic scale, Proc. IEEE **96**(9), 1490–1508 (2008)

66.110 G. Dudek, M. Jenkin, E. Milios: A taxonomy of multirobot systems. In: *Robot teams: From Diversity to Polymorphism*, ed. by T. Balch, L.E. Parker (A. K. Peters, Natick 2002) pp. 3–22

66.111 Y.U. Cao, A.S. Fukunaga, A.B. Kahng: Cooperative mobile robotics: Antecedents and directions, Auton. Robot. **4**(1), 7–27 (1997)

66.112 S. Goss, J.-L. Deneubourg: Harvesting by a group of robots, Proc. 1st Eur. Conf. Artif. Life (MIT, Cambridge 1992) pp. 195–204

66.113 R. Beckers, O. Holland, J.-L. Deneubourg: From local actions to global tasks: Stigmergy and collective robotics, Proc. 4th Int. Workshop Synth. Simul. Living Syst. (Artificial Life IV) (MIT, Cambridge 1994) pp. 181–189

66.114 A. Martinoli: Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies, Ph.D. Thesis (EPFL, Lausanne, Switzerland 1999)

66.115 I.A. Wagner, Y. Altshuler, V. Yanovski, A.M. Bruckstein: Cooperative cleaners: A study in ant robotics, Int. J. Robot. Res. **27**(1), 127–151 (2008)

66.116 J. Werfel, K. Petersen, R. Nagpal: Designing collective behavior in a termite-inspired robot construction team, Science **343**(6172), 754–758 (2014)

66.117 D.J. Stilwell, J.S. Bay: Toward the development of a material transport system using swarms of ant-like robots, Proc. 1993 IEEE Int. Conf. Robot. Autom. 1 (1993) pp. 766–771

66.118 S. Sen, M. Sekaran, J. Hale: Learning to coordinate without sharing information, Proc. 12th Natl. Conf. Artif. Intell., AAAI'94, Menlo Park (1994) pp. 426–431

66.119 R. Groß, M. Dorigo: Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling, Adapt. Behav. **16**(5), 285–305 (2008)

66.120 M.R.A. Chance, J.J. Clifford: *Social Groups of Monkeys, Apes and Men* (E. P. Dutton, New York 1970)

66.121 I.D. Couzin, J. Krause, N.R. Franks, S.A. Levin: Effective leadership and decision-making in animal groups on the move, Nature **433**(7025), 513–516 (2005)

66.122 R.C. Arkin: Cooperation without communication: Multiagent schema-based robot navigation, J. Robot. Syst. **9**(3), 351–364 (1992)

66.123 M.J. Matarić: Designing emergent behaviors: From local interactions to collective intelligence, Proc. 2nd Int. Conf. Simul. Adapt. Behav. (MIT, Cambridge 1992) pp. 432–441

66.124 M.J. Matarić: Minimizing complexity in controlling a mobile robot population, Proc. 1992 IEEE Int. Conf. Robot. Autom., Piscataway (1992) pp. 830–835

66.125 Y. Kuniyoshi, N. Kita, S. Rougeaux, S. Sakane, M. Ishii, M. Kakikura: Cooperation by observation: The framework and basic task patterns, Proc. 1994 IEEE Int. Conf. Robot. Autom., Piscataway (1994) pp. 767–774

66.126 B.P. Gerkey, M.J. Matarić: Sold!: Auction methods for multirobot coordination, IEEE Trans. Robot. Autom. **18**(5), 758–768 (2002)

66.127 V. Trianni, M. Dorigo: Self-organisation and communication in groups of simulated and physical robots, Biol. Cybern. **95**(3), 213–231 (2006)

66.128 T. Balch, R.C. Arkin: Communication in reactive multiagent robotic systems, Auton. Robot. **1**(1), 27–52 (1994)

66.129 M. Rubenstein, K. Payne, P. Will, W.-M. Shen: Docking among independent and autonomous CONRO self-reconfigurable robots, Proc. 2004 IEEE Int. Conf. Robot. Autom. 3 (2004) pp. 2877–2882

66.130 M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo: Swarm robotics: A review from the swarm engineering perspective, Swarm Intell. **7**(1), 1–41 (2013)

66.131 R.A. Brooks: Intelligence without representation, Artif. Intell. **47**(1–3), 139–159 (1991)

66.132 S. Berman, Á. Halász, M.A. Hsieh, V. Kumar: Optimized stochastic policies for task allocation in swarms of robots, IEEE Trans. Robot. **25**(4), 927–937 (2009)

66.133 S. Nouyan, R. Groß, M. Bonani, F. Mondada, M. Dorigo: Teamwork in self-organized robot colonies, IEEE Trans. Evol. Comput. **13**(4), 695–711 (2009)

66.134 W. Liu, A.F.T. Winfield: Modeling and optimization of adaptive foraging in swarm robotic systems, Int. J. Robot. Res. **29**(14), 1743–1760 (2010)

66.135 E. Bonabeau, G. Theraulaz, J.-L. Deneubourg: Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies, Proc. R. Soc. B **263**(1376), 1565–1569 (1996)

66.136 G. Theraulaz, E. Bonabeau, J.-L. Deneubourg: Response threshold reinforcements and division of labour in insect societies, Proc. R. Soc. B **265**(1393), 327–332 (1998)

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

66.138 L.E. Parker, F. Tang: Building multirobot coalitions through automated task solution synthesis, Proc. IEEE **94**(7), 1289–1305 (2006)

66.139 G.A. Korsah, A. Stentz, M. Bernardine Dias: A comprehensive taxonomy for multi-robot task allocation, Int. J. Robot. Res. **32**(12), 1495–1512 (2013)

66.140 L.E. Parker: ALLIANCE: An architecture for fault-tolerant multi-robot cooperation, IEEE Trans. Robot. Autom. **14**(2), 220–240 (1998)

66.141 L.E. Parker: Adaptive heterogeneous multi-robot teams, Neurocomputing **28**(1–3), 75–92 (1999)

66.142 O. Khatib: Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. **5**(1), 90–98 (1986)

66.143 J.H. Reif, H. Wang: Social potential fields: A distributed behavioral control for autonomous robots, Robot. Auton. Syst. **27**(3), 171–194 (1999)

66.144 W.M. Spears, D.F. Spears (Eds.): *Physicomimetics: Physics-Based Swarm Intelligence* (Springer, Berlin, Heidelberg 2011)

66.145 W.M. Spears, D.F. Spears: Distributed, physics-based control of swarms of vehicles, Auton. Robot. **17**(2–3), 137–162 (2004)

66.146 A. Kolling, S. Carpin: Cooperative observation of multiple moving targets: An algorithm and its formalization, Int. J. Robot. Res. **26**(9), 935–953 (2007)

66.147 V. Gazi, K.M. Passino: Stability analysis of social foraging swarms, IEEE Trans. Syst. Man Cybern. Part B Cybern. **34**(1), 539–557 (2004)

66.148 D. Coore: Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer, Ph.D. Thesis (MIT, Cambridge 1999)

66.149 R. Nagpal: Programmable self-assembly using biologically-inspired multiagent control, Proc 1st Int. Joint Conf. Auton. Agents Multiagent Syst.: Part 1, AAMAS '02, New York (2002) pp. 418–425

66.150 J. Beal, J. Bachrach: Infrastructure for engineered emergence on sensor/actuator networks, IEEE Intell. Syst. **21**(2), 10–19 (2006)

66.151 H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T.F. Knight Jr., R. Nagpal, E. Rauch, G.J. Sussman, R. Weiss: Amorphous computing, Commun. ACM **43**(5), 74–82 (2000)

66.152 J. Bachrach, J. Beal, J. McLurkin: Composable continuous-space programs for robotic swarms, Neural Comput. Appl. **19**(6), 825–847 (2010)

66.153 R.S. Sutton, A.G. Barto: *Reinforcement Learning: An Introduction* (MIT, Cambridge 1998)

66.154 M.J. Matarić: Reward functions for accelerated learning, Proc. 11th Int. Conf. Mach. Learn. (Morgan Kaufmann, San Francisco 1994) pp. 181–189

66.155 M.J. Matarić: Learning social behavior, Robot. Auton. Syst. **20**(2–4), 191–204 (1997)

66.156 J. Kober, J.A. Bagnell, J. Peters: Reinforcement learning in robotics: A survey, Int. J. Robot. Res. **32**(11), 1238–1274 (2013)

66.157 I. Harvey, P. Husbands, D. Cliff, A. Thompson, N. Jakobi: Evolutionary robotics: The Sussex approach, Robot. Auton. Syst. **20**(2–4), 205–224 (1997)
66.158 S. Nolfi, D. Floreano: *Evolutionary Robotics – The Biology, Intelligence, and Technology of Self-Organizing Machines* (MIT, Cambridge 2000)
66.159 M. Dorigo, V. Trianni, E. Şahin, R. Groß, T.H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, L.M. Gambardella: Evolving self-organizing behaviors for a *swarm-bot*, Auton. Robot. **17**(2–3), 223–245 (2004)
66.160 V. Trianni: Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots. In: *Studies in Computational Intelligence*, Vol. 108, (Springer, Berlin, Heidelberg 2008)
66.161 C.W. Reynolds: An evolved, vision-based behavioral model of coordinated group motion, From Animals to Animats 2. Proc. 2nd Int. Conf. Simul. Adapt. Behav. (SAB92) (MIT, Cambridge 1993) pp. 384–392
66.162 G.M. Werner, M.G. Dyer: Evolution of herding behavior in artificial animals, From Animals to Animats 2. Proc. 2nd Int. Conf. Simul. Adapt. Behav. (SAB92) (MIT, Cambridge 1993) pp. 393–399
66.163 L. Spector, J. Klein, C. Perry, M. Feinstein: Emergence of collective behavior in evolving populations of flying agents, Genet. Program. Evol. Mach. **6**(1), 111–125 (2005)
66.164 M. Quinn, L. Smith, G. Mayley, P. Husbands: Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors, Philos. Trans. R. Soc. A **361**(1811), 2321–2343 (2003)
66.165 R.A. Watson, S.G. Ficici, J.B. Pollack: Embodied evolution: Distributing an evolutionary algorithm in a population of robots, Robot. Auton. Syst. **39**(1), 1–18 (2002)
66.166 S. Griffith, D. Goldwater, J.M. Jacobson: Self-replication from random parts, Nature **437**(7059), 636 (2005)
66.167 A.F.T. Winfield, M. Dincer Erbas: On embodied memetic evolution and the emergence of behavioural traditions in robots, Memet. Comput. **3**(4), 261–270 (2011)
66.168 M. Yim: Locomotion with a Unit-Modular Reconfigurable Robot, Ph.D. Thesis (Dept. Mech. Eng., Stanford Univ., Stanford 1994)
66.169 W.-M. Shen, B. Salemi, P. Will: Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots, IEEE Trans. Robot. Autom. **18**(5), 700–712 (2002)
66.170 K. Støy: Emergent Control of Self-Reconfigurable Robots, Ph.D. Thesis (The Maersk Mc-Kinney Moller Institute for Production Technology, Univ. Southern Denmark, Denmark 2004)
66.171 G. Chirikjian, A. Pamecha: Evaluating efficiency of self-reconfiguration in a class of modular robots, J. Robot. Syst. **13**(5), 317–338 (1996)
66.172 E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, S. Kokaji: A motion planning method for a self-reconfigurable modular robot, Proc. 2001 IEEE/RSJ Int. Conf. Intell. Robot. Syst., Piscataway, Vol. 1 (2001) pp. 590–597
66.173 D. Brandt, D.J. Christensen: A new meta-module for controlling large sheets of ATRON modules, Proc. 2007 IEEE Int. Workshop Intell. Robot. Syst. (2007) pp. 2375–2380
66.174 Z. Butler, D. Rus: Distributed planning and control for modular robots with unit-compressible modules, Int. J. Robotic. Res. **22**(9), 699–715 (2003)
66.175 S. Murata, H. Kurokawa, S. Kokaji: Self-assembling machine, Proc. 1994 IEEE Int. Conf. Robot. Autom. 1 (1994) pp. 441–448
66.176 M.D. Rosa, S. Goldstein, P. Lee, J. Campbell, P. Pillai: Scalable shape sculpturing via hole motion: Motion planning in lattice-constrained modular robots, Proc. 2006 IEEE Int. Conf. Robot. Autom. (2006) pp. 1462–1468
66.177 Z. Butler, K. Kotay, D. Rus, K. Tomita: Generic decentralized control for a class of self-reconfigurable robots, Proc. 2002 IEEE Int. Conf. Robot. Autom., Piscataway, Vol. 1 (2002) pp. 809–816
66.178 K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, I. Endo: Self-organizing collective robots with morphogenesis in a vertical plane, Proc. 1998 IEEE Int. Conf. Robot. Autom., Piscataway, Vol. 4 (1998) pp. 2858–2863
66.179 M. Yim, Y. Zhang, J. Lamping, E. Mao: Distributed control for 3D metamorphosis, Auton. Robot. **10**(1), 41–56 (2001)
66.180 K. Støy: Using cellular automata and gradients to control self-reconfiguration, Robot. Auton. Syst. **54**(2), 135–141 (2006)
66.181 M. Gauci, J. Chen, W. Li, T.J. Dodd, R. Groß: Self-organised aggregation without computation, Int. J. Robot. Res. **33**, 1145–1161 (2014)
66.182 J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tche, I. Sad, V. Durier, S. Canonge, J.M. Am, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, J.L. Deneubourg: Social integration of robots into groups of cockroaches to control self-organized choices, Science **318**(5853), 1155–1158 (2007)
66.183 A.E. Turgut, H. Çelikkanat, F. Gökçe, E. Şahin: Self-organized flocking in mobile robot swarms, Swarm Intell. **2**(2–4), 97–120 (2008)
66.184 M.J.B. Krieger, J.-B. Billeter, L. Keller: Ant-like task allocation and recruitment in cooperative robots, Nature **406**(6799), 992–995 (2000)
66.185 R. Groß, M. Bonani, F. Mondada, M. Dorigo: Autonomous self-assembly in swarm-bots, IEEE Trans. Robot. **22**(6), 1115–1130 (2006)

66.186 O. Holland, C. Melhuish: Stigmergy, self-organization, and sorting in collective robotics, Artif. Life **5**(2), 173–202 (1999)
66.187 J. Chen, M. Gauci, W. Li, A. Kolling, R. Groß: Occlusion-based cooperative transport with a swarm of miniature mobile robots, IEEE Trans. Robotic. (in press)
66.188 A.J. Ijspeert, A. Martinoli, A. Billard, L.M. Gambardella: Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment, Auton. Robot. **11**(2), 149–171 (2001)
66.189 S. Garnier, C. Jost, J. Gautrais, M. Asadpour, G. Caprari, R. Jeanson, A. Grimal, G. Theraulaz: The embodiment of cockroach aggregation behavior in a group of micro-robots, Artif. Life **14**(4), 387–408 (2008)
66.190 R. Jeanson, C. Rivault, J.-L. Deneubourg, S. Blanco, R. Fournier, C. Jost, G. Theraulaz: Self-organized aggregation in cockroaches, Anim. Behav. **69**, 169–180 (2005)
66.191 J. Krause, A.F.T. Winfield, J.-L. Deneubourg: Interactive robots in experimental biology, Trends Ecol. Evol. **26**(7), 369–375 (2011)
66.192 R. Vaughan, N. Sumpter, J. Henderson, A. Frost, S. Cameron: Experiments in automatic flock control, Robot. Auton. Syst. **31**(1/2), 109–117 (2000)
66.193 A. Gribovskiy, J. Halloy, J.-L. Deneubourg, H. Bleuler, F. Mondada: Towards mixed societies of chickens and robots, Proc. 2010 IEEE Int. Workshop Intell. Robot. Syst. (2010) pp. 4722–4728
66.194 A.L. Christensen, R. O'Grady, M. Dorigo: Swarmorph-script: A language for arbitrary morphology generation in self-assembling robots, Swarm Intell. **2**(2–4), 143–165 (2008)
66.195 C.R. Kube, H. Zhang: Collective robotics: From social insects to robots, Adapt. Behav. **2**(2), 189–218 (1993)
66.196 C.R. Kube, E. Bonabeau: Cooperative transport by ants and robots, Robot. Auton. Syst. **30**(1–2), 85–101 (2000)
66.197 C. Blum, J. Puchinger, G. Raidl, A. Roli: Hybrid metaheuristics in combinatorial optimization: A survey, Appl. Soft Comput. **11**(6), 4135–4151 (2011)