Bio-inspired multi-robot systems

13

*B. Ranjbar-Sahraei, K. Tuyls, I. Caliskanelli, B. Broeker, D. Claes,
S. Alers, G. Weiss*

13.1 Introduction

Recent years have seen an increasing interest in nature-inspired modeling for solving complex computational problems in disciplines at the intersection of computer science, robotics, and economics. An interesting natural phenomenon is the coordination behavior that can be observed in colonies of social insects such as ants and bees. For instance, recent work shows a strong potential in creating artificial systems that mimic insect behavior for solving complex coordination tasks such as, e.g., routing on the internet, mobile *ad hoc* network routing, robotic tasks (Lemmens and Tuyls, 2012; Dressler and Akan, 2010; Floreano and Mattiussi, 2008). These insects have evolved over a long period of time and display remarkable behaviors that are highly suitable for addressing the complex tasks that they are facing. Swarm optimization algorithms, like ant colony optimization (ACO) (Dorigo et al., 2006b), rely on pheromone trails to mediate (indirect) communication between agents. Such insect-inspired multi-agent research has also opened the possibility of applying some of these techniques to robotic systems, i.e., swarm robotics (Dorigo and Roosevelt, 2004; Şahin, 2005). Swarm robotic systems are motivated by a wide range of application areas, such as for instance surveillance and patrolling, where mobile guarding robots are considered as an alternative and improvement over fixed security cameras and even humans. Other application areas include exploration and identification of hazardous environments (e.g., nuclear plants and fire detection), mobile sensor networks, wireless sensor and robot networks, space exploration, etc.

Though easy to simulate, artificial pheromones are hard to bring into real life robotic applications as pheromones need to be deployed and sensed by robots while they decay over time. Recently, non-pheromone-based algorithms were developed as well (Lemmens, 2011). Such algorithms are inspired by the foraging and nest-site selection behavior of (mainly) bees. Generally speaking, bees explore the environment in search for high-quality food sources, and once returned to the hive, they start to dance in order to communicate the location of the source. Using this dance, bees recruit other colony members for a specific food source. The more bees adopt a certain transportation path, the more bees will eventually perform the same dance. Since few dances will not attract enough bees, the best transportation path will eventually prevail.

We draw inspiration from these insect behaviors with the goal to create emergent intelligent systems for distributed coordination that can be deployed in real-world settings on physical platforms. One of the physical platforms we will consider in this chapter is the e-puck robot that has moderate resources such as limited sensing, computation, and actuation capabilities (Mondada et al., 2009). Another platform we will

Biomimetic Technologies. http://dx.doi.org/10.1016/B978-0-08-100249-0.00013-6

consider is the Turtlebot robot that has more extensive resources, contains advanced cameras and carries a general purpose computer (Willow Garage, 2014). We will overview some of our multi-robot swarm experiments in this chapter using simulations and the real robotic platforms, e-puck and Turtlebot robots. Finally, we will describe how the future trend of swarm robotics is moving toward heterogeneity among robots in a swarm.

The remainder of this chapter is structured as follows. Section 13.2 provides the required background on biological inspirations used in this chapter and reviews the related work in the areas of coordination in multi-robot systems. Section 13.3 describes the ant-inspired coordination principle, whereas Section 13.4 shows bee-inspired coordination principles and is split into two parts: (1) Section 13.4.1 covers the foraging behavior of honeybees and (2) Section 13.4.2 exploits pheromone signaling process in honeybees. Future trends of bio-inspired coordination on multi-robot platforms are presented in Section 13.5. The main conclusion of this research is presented in Section 13.6.

13.2 Background

This section provides the required background knowledge to understand the remainder of this chapter. It is divided into two main parts: Section 13.2.1 explains underlying biological inspirations of ant and bee colonies in detail, whereas Section 13.2.2 discusses related work on coordination and coverage in multi-robot and distributed network systems (e.g., sensor nets) based on bio-inspired principles.

13.2.1 Biological inspirations

This section provides the required background knowledge of three different biological inspirations used in this chapter. We start by introducing stigmergic behavior of ants, continue with foraging behaviors of honeybees, and end with the pheromone signaling mechanism for queen bee selection within honeybee colonies.

13.2.1.1 Stigmergic behavior of ants

Most of the research in swarm intelligence revolves around the behavior of ants (Dorigo and Stützle, 2004; Dorigo and Blumb, 2005; Dorigo et al., 2006b). The principle is simple yet elegant: ants deposit a pheromone trail on the path they take during travel. Using this trail, they are able to navigate toward their nest or food and communicate with their peers. More specifically, ants employ an indirect recruitment strategy by accumulating pheromone trails. When a trail gets strong enough, other ants are attracted to it and will follow this trail toward a food destination. The more ants follow a trail, the more pheromone is accumulated and, in turn, the trail becomes more attractive for being followed. This is known as the autocatalytic process. Since long paths take more time to traverse, it will require more ants to sustain a long path. As a consequence, short paths will eventually prevail (see Figure 13.1a). We explain the details of stigmergic coordination algorithm in Section 13.3.

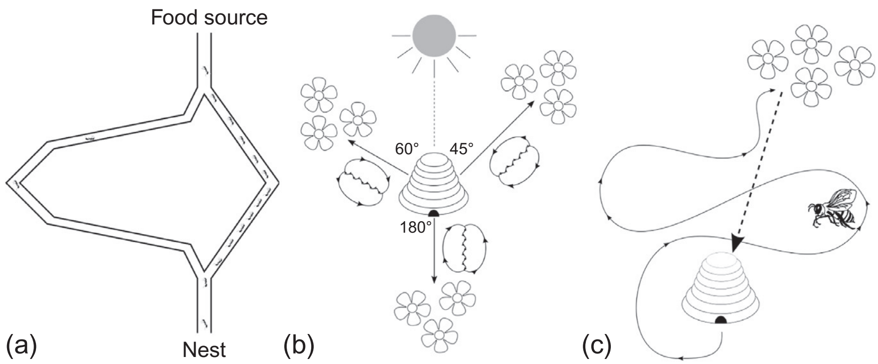

Figure 13.1 Biological inspiration (a) ants exploring two paths; the shortest path prevails. (b) Honeybee waggle dance communicating a PI vector. (c) Lévy flight and path integration.

13.2.1.2 Foraging behavior of honeybees

Foraging honeybees display two types of behavior, i.e., recruitment and navigation. In order to recruit other colony members for food sources, honeybees inform their nest mates of the distance and direction of these food sources by means of a waggling dance performed on the vertical combs in the hive. This dance (i.e., the bee language) consists of a series of alternating left-hand and right-hand loops, interspersed by a segment in which the bee waggles her abdomen from side to side. The duration of the waggle phase is a measure of the distance to the food. The angle between the sun and the axis of a bee's waggle segment on the vertical comb represents the azimuthal angle between the sun and a target location, i.e., the direction in which a recruit should fly (see Figure 13.1b and c). Other members of the colony can adopt the "advertisement" for a food source. The decision mechanism for adopting an "advertised" food-source location by a potential recruit is not completely understood. It is considered that the recruitment among bees is a function of the quality of the food source.

Different species of social insects, such as honeybees and desert ants, make use of non-pheromone-based navigation. Non-pheromone-based navigation mainly consists of Path Integration (PI), which is the continuous update of a vector by integrating all angles steered, and all distances covered (Collett et al., 1998). A PI vector represents the insect's knowledge of direction and distance toward its destination. To construct a PI vector, the insect does not use a mathematical vector summation, but employs a computationally simple approximation (Collett et al., 1998). Using this approximation, the insect is able to return to its destination directly. More precisely, when the path is unobstructed, the insect solves the problem optimally. However, when the path is obstructed, the insect has to fall back on other strategies such as exploration or landmark navigation (Cheng et al., 1987; Collett et al., 2002) to solve the problem. Obviously, bees are able to fly, i.e., when they encounter an obstacle, they can mostly choose to fly over it. However, even if the path is unobstructed, bees tend to navigate over the entire path using landmarks. The landmarks divide the entire path into segments and each landmark has a PI vector associated with it. This behavior decreases navigation errors and ensures robustness. We refer to a home-pointing PI vector as a Home Vector (HV). PI is used in both exploration and exploitation. During exploration, insects constantly update their HV. It is however not used as an exploration strategy. During exploitation, the insects update both their HV and the PI vectors indicating the food source and use these vectors as guidance to a destination. Section 13.4.1 defines foraging-inspired robot coordination algorithm in details.

13.2.1.3 Pheromone signaling behavior of honeybees

The queen bee selection mechanism in honeybee colonies is used to orchestrate the colony by assigning responsibilities to each individual. Roberts (1986) explains the process of larvae differentiation in beehives as an example of such orchestration. Bees have developed a special hormonal system to ensure every beehive has a queen, which maintains the stability of the colony and orchestrates the behavior of all other bees. Throughout its life, a queen bee stimulates a pheromone called Queen Mandibular Pheromone (QMP), which makes the worker bees aware of its presence as a queen. This hormonal mechanism works as follows: the worker bees lick the queen bee and pass the pheromone on to the others. If there is no pheromone passed through the worker bees anymore, this will be an indicator that the queen is dead or has disappeared. In that case, emergency queen cells will be created and workers will select a larva to be fed with large amounts of the royalactin protein (Roberts, 1986). That protein induces the differentiation of honeybee larvae into a new queen. If worker bees keep receiving the pheromone, they will be aware that there is a queen bee to orchestrate the colony and will take no action toward building emergency queen cells.

We describe our *BeePCo* algorithm in detail based on the analogy described in Table 13.1 (see Caliskanelli et al., 2014). Accordingly, the role of queen bee denotes a robot that is responsible for managing the execution of all service requests it receives. Throughout Section 13.4.2, we will refer to such a robot as Queen Robot (QR). They can dynamically differentiate from other robots to indicate their duties for redundancy control (by tuning the pheromone level parameter). However, for the *BeePCo* algorithm presented in this chapter, we allow all the robots to be QRs and we focus on using pheromones as a way of indicating covered arena area rather than role differentiation for redundancy control (unlike in our previous work on WSNs; Caliskanelli et al., 2012b).

Table 13.1 Correlation between bees' pheromone stimulation and multi-robot systems

Bee's pheromone stimulation	Multi-robot systems
Queen bee	Robot responsible for processing services
Worker bees	Robots
Pheromone level	Parameter used for queen robot selection
Lifetime of bee	Operation lifetime of the robot

13.2.2 Related work

This section gives an overview of relevant literature that has described, analyzed, and deployed bio-inspired coordination techniques for multi-robot systems. It is split into three main parts: we start with a generic definition of the multi-robot coverage problem and overview some examples, we continue by providing significant examples of ant inspired and bee-inspired techniques that are used to solve complex problems (i.e., MAC level routing, load balancing, task allocation and resource scheduling, network coverage, etc.) in the broad research area of autonomous, distributed, and networked systems.

13.2.2.1 Coordination and coverage techniques

The concept of coverage is a metric for evaluating robotic systems, which was first introduced by Gage (1992). He defined three basic types of coverage: blanket coverage, in which the objective is to achieve node formation, which maximizes the total detection area; barrier coverage, which aims to minimize the probability of undetected intrusion through the barrier; and sweep or repetitive coverage with the goal to cover all accessible interest points in a given environment over time, while maximizing the rate of visits over all points and minimizing the total distance traveled by all robots.

Blanket coverage is most common for the deployment of mobile sensor networks in an unknown environment; the sensor nodes are initially placed in a compact configuration, where the nodes are trying to spread out such that the area covered by the network is maximized. One example for such a use case is a hazardous material leak in a damaged structure. Mobile sensor nodes equipped with chemical sensors spread out from an initial position to gather information about location and concentration of the hazard. Due to the fact that the communication infrastructure could be damaged, the nodes have to insure their own network structure even if single nodes get lost or destroyed. Many approaches in this field are based on the potential field technique first introduced by Khatib (1985).

Barrier and repetitive coverage problems originate from the computational geometry *Art Gallery Problem* (El-Sherbeny, 2010) and its variant for mobile guard, the *Watchman Route Problem* (Packer, 2008).

In robotics, *repetitive-coverage* can be described as a problem where a team of robots has to visit multiple *points of interests* (POI) in a known environment frequently, to perform certain tasks. The goal of such algorithms is to keep the average visit frequency over all POIs high, while achieving a minimal total traveled distance and a balanced workload over all robots. Typical real-world use cases for such problems are patrolling, lawn mowing and chemical spill cleanup.

Another important form of multi-robot coverage is *terrain coverage* or multi-robot exploration. It can be defined as a problem where a robot tries to visit each and every location in a continuous bounded unknown environment by avoiding obstacles and perform defined tasks as proposed by Correll and Martinoli (2006), Gabriely and Rimon (2003), and Pirzadeh and Snyder (1990). A terrain coverage algorithm must generate a coverage path, which is a chain of motion steps for a robot; the optimal coverage path takes minimal time and guarantee to cover the entire terrain and perform the task efficiently.

13.2.2.2 Ant-inspired techniques

Ant-based multi-robot coverage is highly inspired by the notion of stigmergic communication introduced by Dorigo (1992). The basic idea underlying this form of communication is that pheromones are used as a medium for transmitting messages among artificial ants. During the past few years, variants of Dorigo's method, known as ant colony optimization, have been developed, and it has been shown that it allows for very efficient distributed control and optimization in a variety of problem domains (Dorigo et al., 2006a). Wagner et al. (1999) were the first, who invested stigmergic multi-robot coordination for covering/patrolling the environment. In their approach, the robots were supposed to be able to (1) deposit chemical odor traces and (2) evaluate the strength of smell at every point they reach. Based on these assumptions, they used robots to model an unmapped environment as a graph, and they proposed basic graph search algorithms (such as Depth-First-Search and Breadth-First-Search) for solving robotic coverage problems. Many other researchers used this graph-based modeling scheme in order to design solutions for multi-robot patrolling/covering problems, e.g., Elor and Bruckstein (2009, 2010), Glad et al. (2008, 2010), and Yanovski et al. (2003).

In contrast to the mentioned graph-based techniques, a geometrical framework can also be used for addressing the swarm robotic coverage problem. One of the most important geometric techniques is Voronoi-based coverage that has been introduced for solving robot coverage problems (e.g., see Cortes et al., 2004, 2005; Schwager et al., 2011, 2009). These Voronoi-based techniques aim at devising coverage algorithms, which work according to the following basic rule: *Each vehicle moves toward the center of its Voronoi region*. Based on this rule, many researchers have proposed modified covering approaches, which are adaptable to changes in the environment and are provably convergent (e.g., Schwager et al., 2009; Breitenmoser et al., 2010). However, all these geometrical algorithms require a group of robots with the capability of direct communication and in most of the cases also need very complex mathematical computations (e.g., calculating margins and center of mass for an individual Voronoi region), which limits their potential real-world usage. In Section 13.3, we show how robots can use stigmergic and Voronoi-like coverages with a very simple technique.

Another related research topic is focused on the "real" implementation of stigmergic communication in real-world experiments. For example, chemical substances such as ethanol (C_2H_5OH) are used instead of natural pheromones by Fujisawa et al. (2008). However, with recent developments in communication technology, electrical devices such as Radio Frequency Identification Devices (RFIDs) have gained much interest for such applications. Johansson and Saffiotti (2009) and Herianto et al. (2007) used RFIDs for mapping and exploring an unknown environment. Moreover, Ziparo et al. (2007) proposed a coordinated exploration and multi-robot SLAM for large teams of rescue robots by using RFIDs as environment features.

13.2.2.3 Bee-inspired techniques

Bee Colony Optimization (BCO) was introduced independently by Lemmens et al. (Lemmens et al., 2007a,b; Lemmens and Tuyls, 2012) and by Karaboga et al. (Karaboga, 2010; Karaboga and Basturk, 2008). Unlike ACO (which is only inspired from the notion of stigmergic communication), scientists are inspired by various behaviors of bees: foraging behavior in Lemmens et al. (Lemmens et al., 2011; Lemmens, 2011), Beehive protocol (Wedde et al., 2004), BeeSensor (Saleem and Farooq, 2007), bees mating procedure (Senthilkumar and Chandrasekaran, 2011; Sahoo et al., 2013), and pheromone signaling mechanism in PS (Caliskanelli et al., 2013).

Karaboga et al. (Karaboga, 2010; Karaboga and Basturk, 2008) introduced the Artificial Bee Colony (ABC) algorithm in which bees represent the search agents and their environment represents the potential solutions. In their work, the high-quality candidate solutions represent a pollen source which encourages further exploration of the region by additional bee agents. In the networking context, protocols have been developed in which network packets are treated as biologically inspired agents. Karaboga et al. improved their technique in Akay and Karaboga (2009) by tuning its parameters and modifying their initial work (Karaboga and Basturk, 2007). In the Beehive protocol (Wedde et al., 2004), packets search for efficient routes through an IP network in a process modeled after the foraging behavior of bees. Similar work targeted specifically at WSNs is BeeSensor (Saleem and Farooq, 2007) in which routing is performed via classes of packets following different types of bee behavior: for example as scouts and foragers. The redundancy introduced by BeeSensor is capable of increasing the proportion of delivered packets compared to AODV (Perkins and Royer, 1999), although it experiences increased latency due to the possibility for bee packets to select suboptimal routes during exploration. A general framework through which a set of biological agents can attempt to simultaneously satisfy multiple possibly conflicting objectives (such as latency, energy efficiency, and delivery success in a WSN) is provided in MONSOON (Boonma and Suzuki, 2008). Previous work has also mapped the bee colony model more directly to WSN hardware, with individual nodes representing individual bees, status within the hive corresponding to node responsibilities, and signaling chemicals corresponding to data packets.

Recent work has applied bee protocols specifically to WSN load balancing (Senthilkumar and Chandrasekaran, 2011), which is inspired by the bees mating procedure. This approach focuses on cluster set-up communication overheads by restricting the communications with bee mating election algorithm. Removing the redundant communications inside the cluster increases the successful delivery ratio while decreasing the latency.

Caliskanelli et al. explore the pheromone signaling mechanism in honeybee colonies in Caliskanelli et al. (2012a, 2013) to solve the load balancing (i.e., to distribute the network load among processing elements) and redundancy control issues in large-scale WSNs. Caliskanelli and Indrusiak improved their parameter-rich technique in Caliskanelli and Indrusiak (2013a) by tuning its parameters and modifying their initial work (Caliskanelli et al., 2013). Later on, they applied pheromone signaling process on WSRNs (Wireless Sensor Robot Networks) in Caliskanelli and Indrusiak (2014).

13.3 Ant-inspired multi-robot coordination

This section describes the concept of Ant-Inspired Multi Robot Coordination, *stigmergy*, and introduces an interesting case study called Stigmergic Coverage (StiCo), which is based on the principle of ants' pheromone-based communication. This case study includes the multi-robot coordination principle, simulations, experiments, and also the modeling of StiCo.

In stigmergic coordination, the environment is used as a medium to transfer information among agents: agents deposit traces in the environment in order to send different types of signals, encoded with the source location, to the other agents. The accumulation of traces in the environment provides a shared memory, which allows memoryless simple agents to coordinate easily, while agents might not have any self-awareness of other agents. Furthermore, the use of stigmergic communication allows for coordination among agents of different types, as well studied by Dorigo et al. (2012) in the Swarmanoid project and also adopted to explain the softly heterogeneous swarms by Ranjbar-Sahraei et al. (2013a). In short, the use of stigmergic communication results in high robustness and adaptability, an extremely easy implementation at the microscopic level and yet very efficient at the macroscopic level.

13.3.1 Case study

The StiCo approach follows the principle of indirect, stigmergic coordination to establish a simple but efficient coverage of the environment. In contrast to the classical stigmergic coordination in Ant System (AS), where (1) agents have a tendency to move straight with minor deviations and (2) traces act as sources of attraction, in StiCo robots orbit in circles, instead of moving straight, and the traces have repulsive characteristics instead of attracting the agents. These two differences turn the path-finding characteristic of AS into efficient area coverage of StiCo.

The robustness, scalability, and functional extensibility (see the work by Ranjbar-Sahraei et al., 2012a) make StiCo an interesting alternative to Voronoi-based and graph-based multi-robot coverage approaches, which currently are dominant in the field. Moreover, because of these features, StiCo has a broad application potential. The multi-robot coverage experiment can be used for various monitoring, rescue, and patrolling missions.

13.3.1.1 StiCo principle

In StiCo, robots are equipped with two simple sensors (in the front-left and front-right directions like an ant antenna), capable of detecting immediate traces. Each robot orbits in a circle with a predetermined radius. Based on the circling direction (CW or CCW), one sensor would be considered as the interior sensor and the other one as the exterior one. When the interior sensor detects pheromone, the robot changes its circling direction immediately as shown in Figure 13.2a–c. Otherwise, if exterior

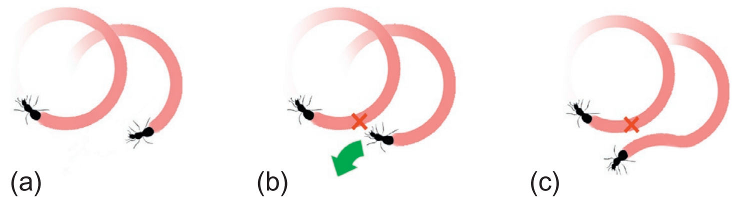

Figure 13.2 StiCo coordination principle: (a) Robots circle around. (b) The right robot detects pheromone. (c) The right robot changes circling direction.

sensor detects pheromone, the robot continues rotating in the same direction until it does not detect pheromone anymore. The amount of pheromone deposited by each robot can practically be adjusted based on pheromone evaporation rates, in a way that robots do not collide with their own pheromones. For further information on StiCo principle, see the work by Ranjbar-Sahraei et al. (2012c).

13.3.1.2 Simulation of StiCo

In order to demonstrate the performance of StiCo, first we translate the previously mentioned rules into an algorithm as shown in Algorithm 1.

The StiCo algorithm is simulated with identical robots in a 40 m $\times$ 40 m field. The linear velocity of each robot is 2 m/s, and the angular velocity is set to $\pm$ 1.0 rad/s. Further details of the simulation environment are provided by Ranjbar-Sahraei et al. (2012b). The coverage algorithm for 40 robots that move based on StiCo is illustrated in Figure 13.3.

In order to demonstrate potential capabilities of this simple algorithm, we consider a nonconvex unknown environment as shown in Figure 13.4a. This environment can

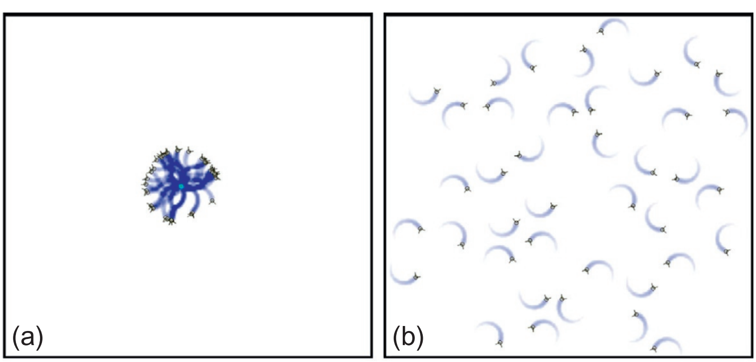

Figure 13.3 Evolution of StiCo in a simple environment (blue shadows are deposited traces). (a) Initial snapshot. (b) Final snapshot.

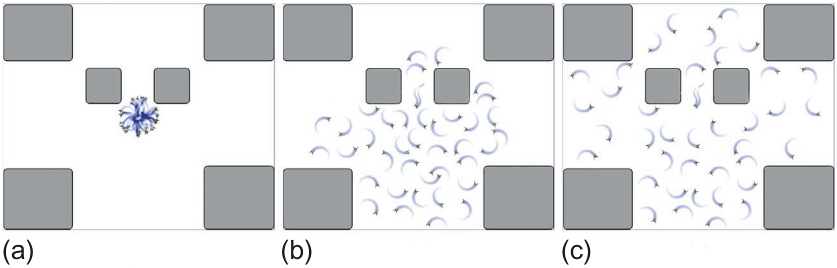

Figure 13.4 Evolution of StiCo in a complex environment. In this simulation, artificial pheromones are deposited on the margins of obstacles to make them detectable for robots. (a) Initial snapshot. (b) Intermediate snapshot. (c) Final snapshot.

represent a devastated area after earthquake or a street map in an emergency condition. Forty robots are initiated at the center of the environment. The coverage steps are illustrated in Figure 13.4a–c.

13.3.1.3 Experiments on StiCo

Motivated by the technique proposed by Kronemann and Hafner (2010), Ranjbar-Sahraei et al. (2013b) have designed a test bed as shown in Figure 13.5. This test bed provides the capability of stigmergic communication to the robots of a swarm.

Figure 13.5 Darkroom with glow-in-the-dark floor, where the e-puck robots circle around and emit UV light onto the floor.

Algorithm 1 StiCo algorithm

```
Require: Each robot can deposit/detect pheromone trails
  Initialize: Choose circling direction (CW/CCW)
  loop
    while (no pheromone is detected) do
      Circle around
      deposit pheromone
    end while
    if (interior sensor detects pheromone) then
      Reverse the circling direction
    else
      while (pheromone is detected) do
        Rotate
      end while
    end if
  end loop
```

In this setting, the floor is covered by a glow-in-the-dark foil (i.e., a foil covered by phosphorescent material, which absorbs UV light and reemits the absorbed light at a lower intensity for up to several minutes after the original excitation), and robots are equipped with UV-LEDs pointing toward the floor. Therefore, as robots move around, they leave glowing trails behind themselves. Furthermore, for detection of these trails, in contrast to the simple method used by Kronemann and Hafner (2010), in which photosensors were used to detect glowing trails, the e-puck's on-board camera is used to detect the trails. By capturing an image and applying a green filter to it, the detailed pattern of green trails in the image are extracted and these patterns are used to measure the presence of trail and also its density over different locations.

Implementation of the StiCo coverage approach with real robots is shown in Figure 13.6a–f.

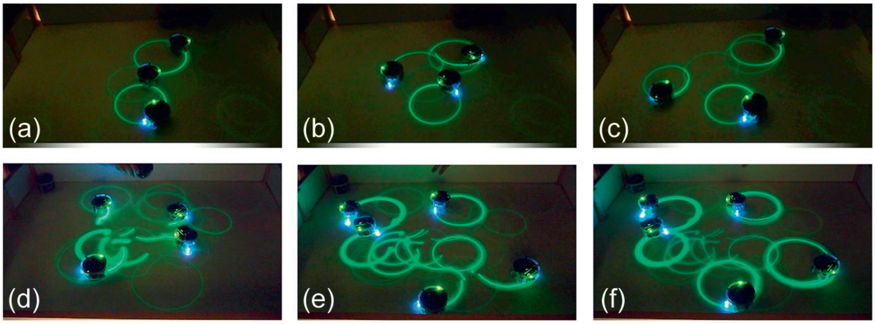

Figure 13.6 Vision-based stigmergic coverage using glowing trails. (a)–(c) Three robots converging into a stable configuration and forming three singular territories. (d)–(f) Five robots converging into a stable configuration and forming five singular territories.

13.3.1.4 Mathematical modeling of StiCo

Due to the simplicity of StiCo's rules for individual robots and the overall complex, yet efficient, behavior of the swarm, many of the classical modeling techniques such as the state-space representation (e.g., used by Ranjbar-Sahraei et al., 2014, to model behaviors in a social network) cannot be used. Instead, the probabilistic modeling techniques studied by Martinoli et al. (2004) and Lerman et al. (2005) are the best fit to such swarm robotic systems.

Ranjbar-Sahraei et al. (2013d) proposed to consider the number of singular robot territories (i.e., the circular region bounded with a single robot traces) as the performance criterion for StiCo. Let us define the state C^n, $n=1, 2,\ldots,M$ for the case that there are n singular territories in the environment. Then, we need a mathematical expression to compute the probability of transition from state C^{n1} to state C^{n2}, in one iteration. This probability is denoted by $P_{n1,n2}$. The first step for computing probability $P_{n1,n2}$ is to partition a general state C^n to all of its possible configurations (the word *partition* refers to a concept of number theory). The configuration $C^n_{T_{a1},T_{a2},\ldots,T_{ak}}$ denotes a configuration in state C^n, in which T_{ai} denotes existence of one ai-tuple in the configuration. If we define $Q^n(K)$ as the probability that a swarm be in state C^n in K-th iteration, then the discrete state transition model can be written as

$$\begin{bmatrix} Q^1(K+1) \\ Q^2(K+1) \\ \vdots \\ Q^M(K+1) \end{bmatrix} = \begin{bmatrix} P_{1,1} & \cdots & P_{M,1} \\ \vdots & \ddots & \vdots \\ P_{1,M} & \cdots & P_{M,M} \end{bmatrix} \begin{bmatrix} Q^1(K) \\ Q^2(K) \\ \vdots \\ Q^M(K) \end{bmatrix} \tag{13.1}$$

For computing $P_{n,n-1}$, which is the transition from C^n to C^{n-1}, the chance that a singular territory becomes a member of a double group should be computed. Let $L(M, n)$ be a function that computes number of possible configurations of M territories, in which exactly n of them are singular. Then, consider the t-th configuration of C^n as

$$C^n_{\underbrace{T_1,T_1,\ldots,T_1}_{n},\underbrace{T_2,T_2,\ldots,T_2}_{r_t},T_{a1_t},T_{a2_t},\ldots,T_{ak_t}} \tag{13.2}$$

The probability for transitions from the t-th configuration of C^n to one of the configurations of C^{n-1}, C^{n+1}, and C^{n+2} are computed as $P_{n,n-1}$, $P_{n,n+1}$, and $P_{n,n+2}$, respectively. Finally, if we ignore the probability for transition from C^n state to the states C^i, in which $i<n-1$ or $i>n+2$, then the probability for remaining in the same state is

$$P_{n,n} = 1 - P_{n,n+1} - P_{n,n+2} - P_{n,n-1} \tag{13.3}$$

In order to check the conditions of fundamental Ergodic Theorem for Markov chains on P matrix, these conditions are simply explained as: (1) P should be stochastic: The values of P must be within the range (0, 1) and each column (or row) sums to 1. (2) P should be irreducible: From each state of our system, it must be possible to get to any other state. (3) P should be aperiodic: The graph represented by P should not be

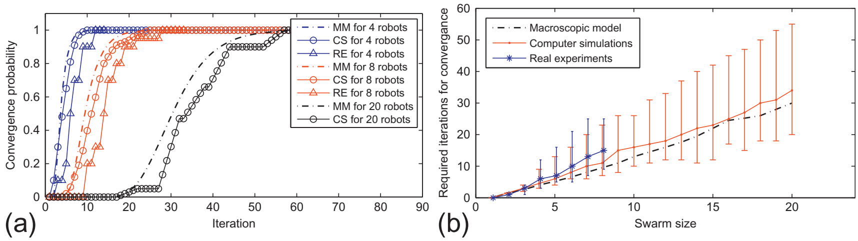

Figure 13.7 Model verification (a) convergence probability in different iterations (MM: Macroscopic Model, CS: Computer Simulations, RE: Real Experiments). (b) Effects of swarm size on convergence time.

bipartite. The first condition holds based on the fact that each probability is in the range of (0, 1), and Equation (13.3), which shows each column sums to 1. The two other conditions can be easily checked with constructing the graph represented by P. Therefore, P is a Markov chain, which can denote a stationary configuration $\prod = \lim_{i\to\infty} P^i \cdot Q(0)$, where $Q(0)$ can be any initial probability distribution for initial configuration.

13.3.1.5 Comparison of model, simulations, and experiments

Three groups of 4, 8, and 20 robots are initialized at the center of an environment. For each group, the probability of being in the final stationary configuration, $Q(\cdot)$, is first computed using the macroscopic model, then computed by using computer simulations, and finally by using real robot experiments. The results of computing the convergence probability are illustrated in Figure 13.7a. The presented results show that the macroscopic model can estimate the behavior of StiCo for robotic swarms of various sizes. As shown in Figure 13.7b, the convergence speed of StiCo increases linearly with growth of the swarm population.

13.4 Bee-inspired multi-robot coordination

This section describes our bee-inspired multi-robot coordination algorithms and is split into two parts. Section 13.4.1 presents foraging behavior-inspired approach, whereas Section 13.4.2 explains the pheromone signaling-inspired approach to solve multi-robot coordination problem.

13.4.1 Bee-inspired biomimicry foraging

In this section, we describe the bees foraging behavior-inspired algorithm, and we illustrate the direct usage of the principles of foraging inspired coordination in swarm intelligence on multi-robot systems. For this set of experiments, we use two types of robot systems: (1) robots with limited resources (e-pucks) and (2) robots with extended resources (Turtlebots). Furthermore, we present results of these two implementations on both robot platforms, discuss the use of sensors for robot localization, camera and video processing for object detection, and the means for having local communication between robots. For further details about our *foraging coordination* approach, please see Alers et al. (2011, 2013a,b, 2014b,c) and Lemmens et al. (2011).

13.4.1.1 Case study

In our approach, recruitment behavior is implemented in analogy with biological bees' dance behavior. Agents share information on previous search experience (i.e., the direction and distance toward a certain food source) only when they are in the hive. Agents in the hive can then decide whether to exploit previous search experience obtained from other agents in the hive, or to exploit their own search experience, if available. As mentioned earlier, bees use an unknown decision mechanism to decide whether to exploit another bee's experience.

The general structure of our bee-inspired algorithm is quite similar to that of algorithms in ASs. It implements both recruitment and navigation behavior and consists of three functions.

13.4.1.2 Bee-inspired algorithm

First, a procedure called ManageBeesActivity() handles agents' activity based on their internal state. Each agent is in one of six internal states. In each state, a specific behavior is performed. Agent state "AtHome" indicates that the agent is located at the hive. While in this state, the agent determines to which new state it will go. Agent state "StayAtHome" also indicates that the agent is located at the hive. However, while in this state it will remain there unless there is previous search experience available to exploit. Previous search experience is represented by a PI vector indicating a food source. If such experience is available, the agent will leave the hive to exploit the previous search experience. Agent state "Exploitation" indicates that the agent is exploiting previous search experience. An agent either exploits its own search experience or acquires a PI vector from other agents inside the hive. The agent determines which cell to move to in order to match the PI vector indicating the food source. Agent state "Exploration" indicates that the agent is exploring its environment in search for food. Agent state "HeadHome" indicates that the agent is heading home without carrying any food. The agent reaches home by following its Homing Vector (HV). The HV is a PI vector indicating the hive. From the moment an agent starts its foraging trip, this HV is continuously calculated for each agent. Agent state "CarryingFood" indicates that the agent has found food and that it is carrying the food back toward the hive. The agent's return path depends on the same HV as with agent state "HeadHome." Next come the experiments, which show how a group of real robots can coordinate using bee-inspired algorithm.

13.4.1.3 Experiments using robots with limited resources (e-pucks)

The bee foraging experiments show the effectiveness of the embodied foraging behavior in a swarm of e-pucks. In Figure 13.8, we present the stages that the experiment goes through. The goal of the experiment is to show that each separate behavior actually works in an embodied swarm. Therefore, the experiment starts with a swarm of e-pucks surrounding the hive (see Figure 13.8a). Figure 13.8b shows the stage in which a portion of the swarm starts foraging while others remain around the hive, waiting for the information to exploit. Figure 13.8c presents the situation in which an exploring e-puck finds food and returns to the hive by using its constructed PI vector. Once returned to the nest, the e-puck communicates its PI findings by means of a virtual dance. The hive collects these experiences and offers these to recruits. Finally, Figure 13.8d gives the situation in which other e-pucks communicated with the hive and have attained the PI vector toward the food source and are traveling to the food source guided by this PI vector. A demonstration movie can be found online.[1]

Experiments using robots with extended resources (Turtlebots)

In this set of experiments, we introduce the swarm robotics with extended resources. These swarms use general purpose computers, high quality and advanced video cameras, 3D sensors for mapping (e.g., laser range finders), accurate wheel encoders that make enhanced odometry possible, fused data of accelerometers, and a gyroscope.

The Turtlebot platform is a robot with extended resources. This robot is equipped with a laptop with core-i3 CPU for computation that is running the Robot Operating System[2] framework. As a main sensing unit, the Turtlebot is equipped with a Kinect sensor. The full RGBD information is used to detect and locate AR markers. For static obstacle detection, we only use the depth information of the sensor together with three

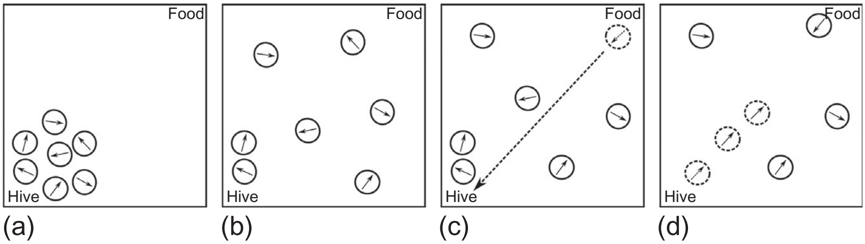

Figure 13.8 The four stages of Biomimicry Foraging. (a) All robots start at the nest location. (b) The robots randomly disperse through the environment looking for a food location. (c) A robot that has found food returns to the nest location by the shortest possible path. (d) The food location is communicated to other robots and they start to exploit this food source.

[1] http://swarmlab.unimaas.nl/papers/bnaic2011demo/.

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

bumpers that are located in the front half of the robot. Furthermore, the robot estimates its position by integrating the wheel odometry and gyro information. Hence, no map of the environment is built and the only known reference point is the target location marker. This can lead to the problem that if the odometry is faulty, the robot does not always find the target location back. As a solution, the robots fall back into a search mode, if this is the case. Another solution could be to implement a Northstar-like navigation system, by providing a fixed frame of reference, which is almost always visible from any location.

To enable visual robot-robot detection, we equipped every Turtlebot with six unique markers, which are oriented in a way that at least one marker is visible from any angle. To track and decode these markers, we make use of a toolkit called ALVAR; more specifically, we use the ROS wrapper[3] of this library. We use a customized bundle detection method to determine the center of the detected robot dependent on the decoded markers. Kalman filtering is applied to get better and more stable readings and consequently a more accurate estimate of the detected robot's position, heading, and speed. These parameters are used again for collision avoidance.

Communication is realized over Wi-Fi with a UDP connection to each Turtlebot using the LCM library.[4] Even though global communication would be possible, we limit the communication such that every robot listens only to its own channel. To simulate local communication, the robots can only communicate with another robot when it is in view and in close proximity, i.e., less than 1 m away.

In order to avoid robot to robot collisions, we rely on the marker detection to predict positions and speeds of the other robots. This information can be used to efficiently compute a noncolliding speed vector as we have developed previously in Claes et al. (2012). In contrast to this previous approach, in which the robot-robot detection was avoided by using a global reference frame and broadcasting the positions to all robots via Wi-Fi, solely the marker detection and the predictions using a Kalman filter are used. This means that a few collisions still might occur due to failure to detect the markers of the other robots and additionally, there are certain configurations in which the robots cannot see each other due to the field of view of the Kinect sensor, e.g., when two robots drive in a V-shape toward each other, the field of view of the Kinect is too narrow to detect the other robot.

As shown by Alers et al. (2014a), multiple Turtlebots perform a foraging task, i.e., starting at the Hive (H) location and randomly exploring the unknown environment for a specific Food (F) location. This is shown in Figure 13.9. Another way of locating a food location is by asking bypassing robots for a known food location, which is done by simulating local communication over Wi-Fi. When the source is found, the robot starts to exploit this source, i.e., driving from the food to the hive location until the food is depleted or a better source is found. A video showing this demonstration can be found in the online material.[5]

[3] http://wiki.ros.org/ar_track_alvar.

[4] https://code.google.com/p/lcm/.

[5] http://swarmlab.unimaas.nl/papers/aamas-2014-foraging.

Figure 13.9 Multi-robot foraging using swarm robots with extended resources. (a) All robots start at the hive (H) location. (b) Robots are exploring the unknown environment randomly. The left two robots have found the food (F) location and are foraging between the hive and the food location. (c) All robots have converged to foraging behavior.

13.4.2 Bee-inspired pheromone signaling

In this section, we show some initial outcomes of the honeybee-inspired pheromone signaling method that allows a team of robots to maximize the total area covered in an environment in a distributed manner.

The same pheromone signaling method is applied to the multi-robot coverage problem as is introduced in Caliskanelli et al. (2014). The proposed coverage technique is inspired by the behavior described in Section 13.2. The role of queen bee denotes a robot that is responsible for managing the execution of all service requests it receives. Throughout this chapter, we will refer these robots as QR, and their responsibility (service) is to patrol in the field. The basic strategy of the algorithm is based on the periodic transmission of pheromones by QRs, and its retransmission by recipients to their neighbors. The pheromone level at each robot decays with time and with distance to the source. All robots accumulate pheromone received from other QRs and if at a particular time the pheromone level of a robot is below a given threshold, this robot will differentiate itself into a QR. To make it clear, the threshold we used for this work is 0 and as such all the robots are QRs at all times. Although we do not particularly benefit from robot differentiation in this work (unlike our previous research on WSNs Caliskanelli et al., 2012a, 2013), we still describe the differentiation process for the sake of completeness and to provide a base for our future work on multi-robot coverage. In the *BeePCo* technique, the level of pheromone indicates the resource usage and robot density in a particular area of the network. Areas in the robotic arena that have lower level of pheromone at a given time demonstrate less resource usage, and less robot density as opposed to other parts of the network. This means, areas with low pheromone level have either a low coverage or not covered at all. Some preliminary performance experiments are shown in the next sections.

13.4.2.1 Case study

To evaluate the performance of *BeePCo*, we have designed a three-tier system-level simulation model that represents the application layer (consisting of tasks), platform layer (consisting of robots), and the mapper (that maps the tasks from the application layer to the platform layer). Our system-level simulator, *Fast*, is written in Java, and it is an abstract simulator—trading accuracy for efficiency, scalability, and flexibility. For further details about *Fast*, please see Caliskanelli et al. (2012a, 2013).

The set of experiments in this section aims to show the area coverage and network connectivity of the proposed *BeePCo* technique. Area coverage in this study is referred to as maximization of the total area covered by the sensors of the involved robot(s), as defined by Gage (1992). On the other hand, network connectivity refers to the ability to transfer data between robots. As the proposed technique is more suitable for noncritical applications, we focus on participial network connectivity instead of full connectivity. The simulation setup consists of a system of 40 robots, each having a sensing and communication radius of 25 cm. The application arena size is set to 300 cm × 300 cm, where the robots are initially deployed randomly in the center of the arena, in a square region of size 5 cm × 5 cm. We examine two different scenarios:

- *BeePCo with network assurance* represents a case where the robots try to keep the wireless communication channels alive throughout the simulation. This restricts robots to move too far away from each other. The algorithm is forcing robots to move backward if a robot is not connected to at least one other robot in the arena. This scenario is developed to be used for applications where events are expected to be reported to the sink, such as patrolling.
- *BeePCo without network assurance* represents a case where a maximum spread of the robots on the arena is required in order to maximize area coverage.

Figure 13.10 shows the spread of 40 robots when network assurance is required from the *BeePCo* algorithm. Figure 13.10a–c represents the layout of the robots and their

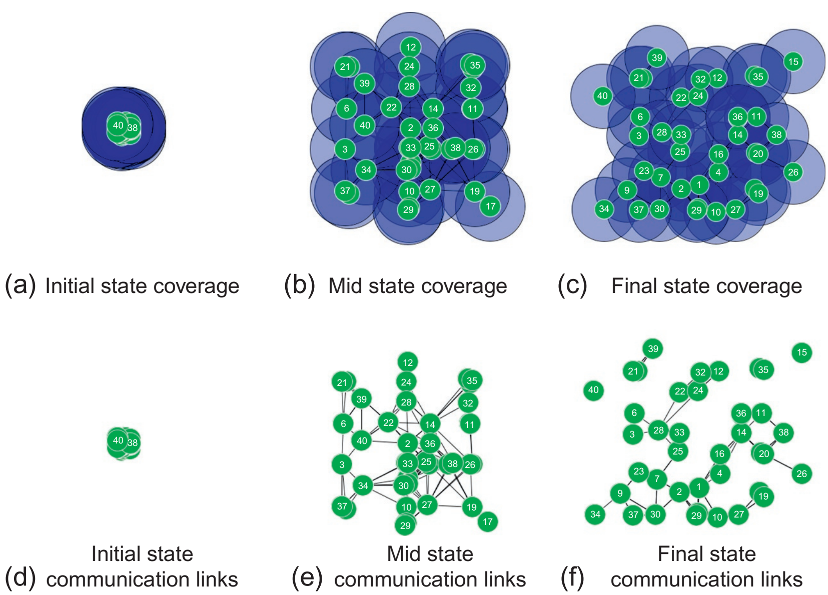

Figure 13.10 Spread of an MRS of 40 robots: (a)–(c) area coverage and (d)–(f) connectivity of the communication links when network connectivity is assured by *BeePCo*.

spread over the arena incrementally. The area that is covered by each robot is plotted by a blue circle indicating the transmission range of each robot, and is normalized with the surface of the arena. Figure 13.10d–f shows how the active wireless communication links between the robots evolve as time elapses. Robots are deployed in the middle of the arena in the initial stage as shown in Figure 13.10a and d. In the initial stage, coverage is very low, whereas connectivity is very high. As time passes by, robots start spreading in the arena as shown in Figure 13.10b and e; the area coverage increases where the network connectivity remains high. At the end of the simulation, robots spread out as much as they can while trying to keep the wireless communication channels active as shown in Figure 13.10c and f.

Figure 13.11 illustrates the spread of the robots when network connectivity is not assured. Similar to Figure 13.10, area coverage and connectivity are inspected on an arena with 40 robots. In this set of experiments, we use the same setup as shown in Figure 13.10. Figure 13.11a and d shows the initial stage of the robots after deployment. As can be seen from Figure 13.11a, the *BeePCo* algorithm without network assurance performs very similar to (Figure 13.10a and d) *BeePCo* algorithm with network assurance. Later on as the simulation evolves, Figure 13.11b and e starts spreading wider as opposed to Figure 13.10b and e. By the end of the simulation, robots are spread all along the arena where the area coverage gets very high as shown in Figure 13.11c. Unlike the area coverage, network connectivity is lost almost entirely.

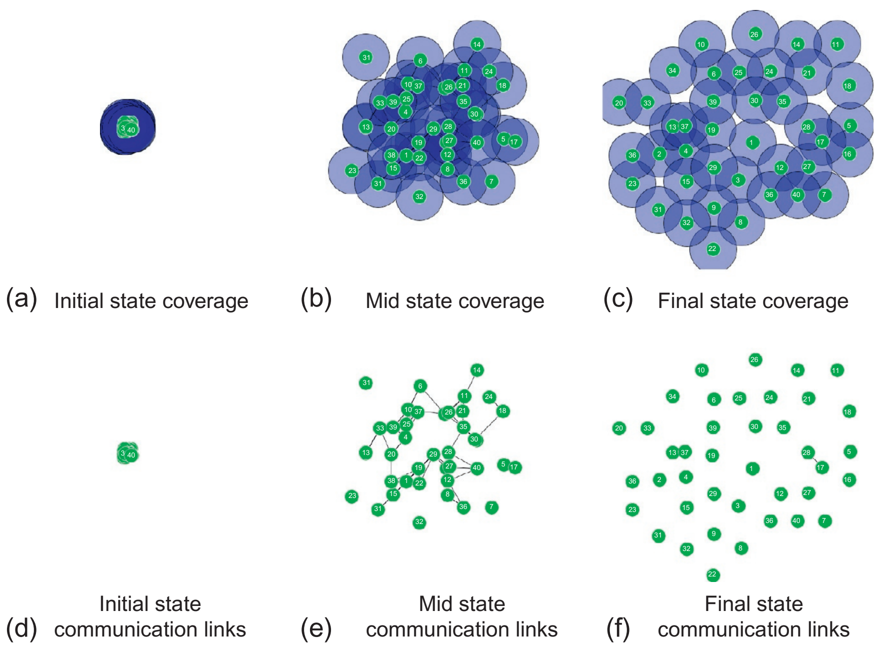

Figure 13.11 Spread of an MRS of 40 robots: (a)–(c) area coverage and (d)–(f) connectivity of the communication links when network connectivity is assured by *BeePCo*.

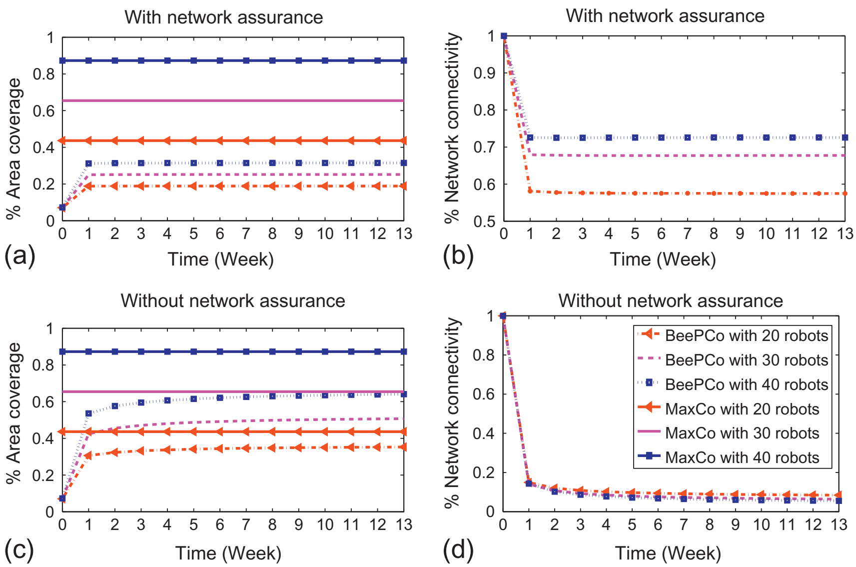

Figure 13.12 Experimental results: (a) and (c) % area coverage, (b) and (d) % network connectivity with different number of robots in an MRS.

Dramatic difference in the network connectivity is observed between Figures 13.10f and 13.11f.

The results shown in Figure 13.12 are based on 30 different runs over six different configurations (with and without network assurance each with the three alternatives for the number of robots in the environment), in a total of 180 simulation runs to ensure the statistical significance. Each run simulated the case study for 13 weeks, to illustrate the long-term effects of the spread of the *BeePCo* coverage algorithm based on bees' pheromone signaling process. MaxCo results are based on mathematical calculations based on the total transmission area over total area.

Figure 13.12 shows the percentage of the area coverage (a) and (c) and the percentage of network connectivity (b) and (d). *BeePCo* algorithm to illustrate the effects of the number of the robots on two different scenarios: with and without the network assurance. In addition to experiments shown in Figures 13.10 and 13.11, the number of robots is varied (with 20 and 30) and experiments are held while the simulation setup is kept the same. Our observations are as follows:

Independent from the network assurance, the area coverage increases as the number of the robots increase. This behavior is shown in Figure 13.12a and c on *BeePCo* algorithm and MaxCo; area coverage achieves highest percentage with 40 robots. Area coverage is approximately 10% more when network connectivity is not assured in all three ranges of number of robots. Network connectivity is very low when network is not assured as shown in Figure 13.12d and is irrelevant from the number of the robots. Figure 13.12a exhibits that *BeePCo* algorithm with network assurance increases network connectivity 50% more than *BeePCo* with no network assurance. This does not reflect on area coverage, which we believe is the benefit of *BeePCo* with network assurance. The performance difference in terms of area coverage between *BeePCo* with and without network assurance is less than 10% in a system with 20 robots. Although this difference increases up to 15% as the number of robots increases, we believe *BeePCo* with network assurance brings much more benefit (as opposed to *BeePCo* without network assurance) in terms of connectivity by 50%.

13.5 Future trends

Polymorphism is a known phenomenon in biological systems, meaning that various forms or types of individuals are seen among the members of a single species. This phenomenon can be seen in many different expressions in nature, ranging from polymorphic ants and birds (with differences in, e.g., size, color, and strength) to different blood types in humans. Natural polymorphism can be categorized into discrete variations (e.g., blood groups) and continuous ones (e.g., smooth height variations in human population).

Recently, heterogeneity of agents has attracted the attention of different research communities in computer science, artificial intelligence, and robotics. For example, Montes de Oca et al. (2009) have studied various types of heterogeneity that can be ascribed to particle swarm optimizers and have shown how this can improve the efficiency of computational techniques. Heterogeneity of complex networks has been also studied, e.g., by Moreno et al. (2002), where the dynamics of epidemics in complex heterogeneous architectures are investigated.

Recently, heterogeneity has found its way to swarm robotics as well. Dorigo et al. (2012) have introduced a distributed robotic swarm, namely *Swarmanoid*, which consists of three different robot types (eye-bots, hand-bots, and foot-bots). Such structures, in which different robots have different capabilities and different goals, refer to the hard heterogeneity in swarms. On the other hand, softly heterogeneous swarm robotics introduced by Ranjbar-Sahraei et al. (2013a) refer to the situation in which a group of similar robots all have the same goals but slightly different levels of capability. As an example of such systems, consider a scenario in which a group of simple robots try to uniformly disperse in an unknown environment. Each robot can simply compute its distance with the neighboring robots and after computing the borders of its own territory, moves toward the center of the territory. Gradually, all robots make a uniform coverage in the area (i.e., known as Voronoi coverage). This approach can be very efficient in convex environments. However, as soon as nonconvexities such as obstacles are added to the environment, this approach fails as robots get stuck behind the obstacles (i.e., reaching the local optimum in a coverage problem). For this specific problem, Staňková et al. (2013) and Ranjbar-Sahraei et al. (2013c) proposed and extended a coverage method, namely StaCo, based on the concept of Stackelberg game, which uses soft heterogeneity in a swarm to overcome the problem of local optima.

The concept of heterogeneity in robotic swarms suggests that these swarms can perform better than the long-established uniform swarms in various scenarios. For instance, they can perform both exploration and exploitation of an environment in a rescue mission, or can act highly flexibly in passing obstacles and nonconvexities in a flocking mission. In short, inspired by polymorphism in biological organisms, using this concept of heterogeneity in robotic swarms is a promising practice for overcoming the available limitations in swarm robotics.

13.6 Conclusions

Coordination is a key challenge when deploying teams of distributed multi-robot systems. Lightweight interactions among robots (i.e., wireless communications) are not only a desired feature for such platforms; they are a great need. Furthermore, simple yet effective algorithms that avoid complex heavy computations are desirable in multi-robot systems. Therefore, bio-inspired solutions for the challenging problem of multi-robot coordination are gaining more importance. In this chapter, we presented an overview of our work on ant- and bee-inspired coordination principles for coordination in multi-robot systems.

In Section 13.3, we described our ant-inspired StiCo approach and illustrate its effectiveness on a case study of deployed e-pucks. We illustrated the performance of the StiCo approach using a dark room that allows emission of UV light on the floor. The glowing trails of the robots are captured by on-board cameras of e-pucks. Experimental results showed that simple communication principle of ants can be used to address the multi-robot coordination problem.

In Section 13.4.1, we presented the bee-inspired foraging algorithm to solve the coordination problem of multi-robot systems. We showed the impact of the foraging algorithm on two different experimental setups: (1) a swarm of robots with limited resources (e-pucks) and (2) a swarm of robots with extended resources (Turtlebots). The experiments served as a proof of concept: First, we showed how the bee-inspired mechanism can be used in a real-life autonomous robotic swarm, which mimics the basic foraging behavior of bees. Second, by the direct deployment of bee-inspired algorithms onto a robot swarm, scalability, robustness, and efficiency on foraging tasks in more complex and dynamic environments were investigated.

In Section 13.4.2, we explained the bee-inspired pheromone signaling mechanism to address coordination in multi-robot systems. We explored its performance consequences on a case study using a system-level simulator, *Fast*. We illustrated the spread of a multi-robot system of 40 robots using the *BeePCo* algorithm with and without the network connectivity assurance. We report area coverage and communication links for both scenarios. Illustrations of both scenarios showed that the pheromone signaling algorithm can be used to address the coordination and coverage problem in multi-robot systems. We also showed that the *BeePCo* algorithm achieves larger coverage when the network connectivity is not assured (for one-shot applications, i.e., fast exploration of an unknown environment) as opposed to scenarios when the network connectivity is assured (where events are expected to be reported, i.e., long-term periodic patrolling).

References

Akay, B., Karaboga, D., 2009. Parameter tuning for the artificial bee colony algorithm. In: Proceedings of the 5th International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, Wroclaw, Poland. Springer-Verlag, Berlin, ISBN: 978-3-642-04440-3, pp. 608–619.

Alers, S., Bloembergen, D., Hennes, D., de Jong, S., Kaisers, M., Lemmens, N., Tuyls, K., Weiss, G., 2011. Bee-inspired foraging in an embodied swarm. In: Proceedings of the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 1311–1312.

Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., Weiss, G., 2013a. An experimental framework for exploiting vision in swarm robotics. In: ADAPTIVE 2013, The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications, pp. 83–88.

Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., Weiss, G., 2013b. Evaluation of an experimental framework for exploiting vision in swarm robotics. In: Advances in Artificial Life, ECAL, vol. 12, pp. 775–782.

Alers, S., Claes, D., Tuyls, K., Weiss, G., 2014a. Biologically inspired multi-robot foraging. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). International Foundation for Autonomous Agents and Multiagent Systems, pp. 1683–1684.

Alers, S., Claes, D., Tuyls, K., Weiss, G., 2014b. Biologically inspired multi-robot foraging. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 1683–1684.

Alers, S., Tuyls, K., Ranjbar-Sahraei, B., Claes, D., Weiss, G., 2014c. Insect-inspired robot coordination: foraging and coverage. In: Artificial Life, vol. 14.

Boonma, P., Suzuki, J., 2008. MONSOON: a coevolutionary multiobjective adaptation framework for dynamic wireless sensor networks. In: Proceedings of the 41st Annual International Conference on System Sciences, January 2008, p. 497. http://dx.doi.org/10.1109/HICSS.2008.323.

Breitenmoser, A., Schwager, M., Metzger, J.C., Siegwart, R., Rus, D., 2010. Voronoi coverage of non-convex environments with a group of networked robots. In: Proceedings of the International Conference on Robotics and Automation (ICRA 10), May 2010, pp. 4982–4989.

Caliskanelli, I., Harbin, J., Indrusiak, F., Polack, L., Mitchell, P., Chesmore, D., 2012a. Runtime optimisation in wsns for load balancing using pheromone signalling. In: 3rd IEEE International Conference on NESEA, December 2012.

Caliskanelli, I., Harbin, J., Indrusiak, L.S., Mitchell, P., Chesmore, D., Polack, F., 2012b. Runtime optimisation in wsns for load balancing using pheromone signalling. In: 2012 IEEE 3rd International Conference on Networked Embedded Systems for Every Application (NESEA), pp. 1–8.

Caliskanelli, I., Harbin, J., Indrusiak, L.S., Mitchell, P., Polack, F., Chesmore, D., 2013. Bio-inspired load balancing in large-scale wsns using pheromone signalling. Int. J. Distrib. Sens. Netw. 2013, 1–14.

Caliskanelli, I., Indrusiak, L., 2013a. Search-based parameter tuning on application-level load balancing for distributed embedded systems. In: Proceedings of the 11th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2013).

Caliskanelli, I., Indrusiak, L., 2014. Using mobile robotic agents to increase service availability and extend network lifetime on wireless sensor and robot networks. In: 12th IEEE International Conference on INDIN, July 2014.

Caliskanelli, I., Broecker, B., Tuyls, K., 2014. Multi-robot coverage: a bee pheromone signalling approach. In: Submitted to the International Conference on Artificial Life and Intelligent Agents (ALIA 14).
Cheng, K., Collett, T., Pickhard, A., Wehner, R., 1987. The use of visual landmarks by honeybees: bees weight landmarks according to their distance from the goal. J. Comp. Physiol. A. 161 (3), 469–475.
Claes, D., Hennes, D., Tuyls, K., Meeussen, W., 2012. Collision avoidance under bounded localization uncertainty. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012), Vilamoura, Portugal, October 2012.
Collett, M., Collett, T.S., Bisch, S., Wehner, R., 1998. Local and global vectors in desert ant navigation. Nature 394 (6690), 269–272.
Collett, M., Harland, D., Collett, T.S., 2002. The use of landmarks and panoramic context in the performance of local vectors by navigating honeybees. J. Exp. Biol. 205, 807–814.
Correll, N., Martinoli, A., 2006. Collective inspection of regular structures using a swarm of miniature robots. In: Ang, J., Marcelo, H., Khatib, O. (Eds.), Experimental Robotics IX. In: Springer Tracts in Advanced Robotics, vol. 21. Springer, Berlin, Heidelberg, ISBN: 978-3-540-28816-9, pp. 375–386. http://dx.doi.org/10.1007/11552246_36.
Cortes, J., Martinez, S., Karatas, T., Bullo, F., 2004. Coverage control for mobile sensing networks. IEEE Trans. Robot. Autom. 20 (2), 243–255.
Cortes, J., Martinez, S., Bullo, F., 2005. Spatially-distributed coverage optimization and control with limited-range interactions. ESAIM Control. Optim. Calc. Var. 11, 691–719.
Dorigo, M., 1992. Optimization, learning and natural algorithms. Thesis report, Politecnico di Milano, Italy.
Dorigo, M., Blumb, C., 2005. Ant colony optimization theory: a survey. Theor. Comput. Sci. 344, 243–278.
Dorigo, M., Roosevelt, A.F., 2004. Swarm robotics. In: Special Issue: Autonomous Robots, Citeseer.
Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. A Bradford Book. MIT Press, Cambridge.
Dorigo, M., Birattari, M., Stutzle, T., 2006a. Ant colony optimization. IEEE Comput. Intell. Mag. 1556-603X. 1 (4), 28–39. http://dx.doi.org/10.1109/MCI.2006.329691.
Dorigo, M., Birattari, M., Stutzle, T., 2006b. Ant colony optimization: artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag. 1 (4), 28–39.
Dorigo, M., Floreano, D., Gambardella, L.M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., et al., 2012. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. IEEE Robot. Autom. Mag. 20 (4), 60–71.
Dressler, F., Akan, O.B., 2010. A survey on bio-inspired networking. Comput. Netw. 54 (6), 881–900.
Elor, Y., Bruckstein, A.M., 2009. Multi-a(ge)nt graph patrolling and partitioning. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology—Volume 02, WI-IAT'09, Washington, DC, USA. IEEE Computer Society, Washington, DC, USA, ISBN: 978-0-7695-3801-3, pp. 52–57. http://dx.doi.org/10.1109/WI-IAT.2009.125.
Elor, Y., Bruckstein, A., 2010. Autonomous multi-agent cycle based patrolling. In: Swarm Intelligence. Lecture Notes in Computer Science, vol. 6234. Springer, Berlin, Heidelberg, ISBN: 978-3-642-15460-7, pp. 119–130.
El-Sherbeny, N.A., 2010. Vehicle routing with time windows: an overview of exact, heuristic and metaheuristic methods. J. King Saud Univ. Sci. 22 (3), 123–131. http://dx.doi.org/10.1016/j.jksus.2010.03.002.

Floreano, D., Mattiussi, C., 2008. Bio-inspired artificial intelligence: theories, methods, and technologies. The MIT Press, Cambridge, MA, USA.

Fujisawa, R., Imamura, H., Hashimoto, T., Matsuno, F., 2008. Communication using pheromone field for multiple robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, IROS 2008, September 2008, pp. 1391–1396.

Gabriely, Y., Rimon, E., 2003. Competitive on-line coverage of grid environments by a mobile robot. Comput. Geom. 24 (3), 197–224. http://dx.doi.org/10.1016/S0925-7721(02)00110-4.

Gage, D.W., 1992. Command control for many-robot systems. In: 19th Annual AUVS Technical Symposium, pp. 22–24.

Glad, A., Simonin, O., Buffet, O., Charpillet, F., 2008. Theoretical study of ant-based algorithms for multi-agent patrolling. In: Proceeding of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence. IOS Press, Amsterdam, The Netherlands, ISBN: 978-158603-891-5, pp. 626–630. http://dl.acm.org/citation.cfm?id=1567281.1567417.

Glad, A., Simonin, O., Buffet, O., Charpillet, F., 2010. Influence of different execution models on patrolling ant behaviors: from agents to robots. In: Proceedings of the Ninth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10).

Herianto, Sakakibara, T., Kurabayashi, D., 2007. Artificial pheromone system using RFID for navigation of autonomous robots. J. Bionic Eng. 4 (4), 245–253.

Johansson, R., Saffiotti, A., 2009. Navigating by stigmergy: a realization on an RFID floor for minimalistic robots. In: IEEE International Conference on Robotics and Automation, 2009, ICRA'09, May 2009, pp. 245–252.

Karaboga, D., 2010. Artificial bee colony algorithm. Scholarpedia 5 (3), 6915.

Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. 39 (3), 459–471.

Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. 8 (1), 687–697.

Khatib, O., 1985. Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the 1985 IEEE International Conference on Robotics and Automation, March 1985, vol. 2, pp. 500–505. http://dx.doi.org/10.1109/ROBOT.1985.1087247.

Kronemann, M.L., Hafner, V.V., 2010. Lumibots—making emergence graspable in a swarm of robots. In: The ACM Designing Interactive Systems Conference, pp. 408–411.

Lemmens, N., 2011. Bee-Inspired Distributed Optimization. Maastricht University, Netherlands.

Lemmens, N., Tuyls, K., 2012. Stigmergic landmark optimization. Adv. Complex Syst. 15 (8), 1150025.

Lemmens, N., de Jong, S., Tuyls, K., Nowé, A., 2007a. A bee algorithm for multi-agent systems: recruitment and navigation combined. In: ALAg 2007, AAMAS'07 Honolulu, Hawai.

Lemmens, N., de Jong, S., Tuyls, K., Nowé, A., 2007b. Bee behaviour in multi-agent systems. In: Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning, 5th, 6th, and 7th European Symposium, ALAMAS 2005-2007 on Adaptive and Learning Agents and Multi-Agent Systems, Revised Selected Papers, pp. 145–156.

Lemmens, N., Alers, S., Tuyls, K., 2011. Bee-inspired foraging in a real-life autonomous robot collective. In: Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC), pp. 459–460.

Lerman, K., Martinoli, A., Galstyan, A., 2005. A review of probabilistic macroscopic models for swarm robotic systems. In: Swarm Robotics. Springer, Berlin, pp. 143–152.

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

Mondada, F., Bonani, M., et al., 2009. The e-puck, a robot designed for education in engineering. In: 9th Conference on Autonomous Robot Systems and Competitions, vol. 1. IPCB: Instituto Politcnico de Castelo Branco, Castelo Branco, Portugal, pp. 59–65. http://www.est.ipcb.pt/robotica2009/.

Montes de Oca, M.A., Pena, J., Stutzle, T., Pinciroli, C., Dorigo, M., 2009. Heterogeneous particle swarm optimizers. In: IEEE Congress on Evolutionary Computation, 2009, CEC'09. IEEE, Piscataway, NJ, pp. 698–705.

Moreno, Y., Pastor-Satorras, R., Vespignani, A., 2002. Epidemic outbreaks in complex heterogeneous networks. Eur. Phys. J. B 26 (4), 521–529.

Packer, E., 2008. Robust geometric computing and optimal visibility coverage. PhD thesis, Stony Brook, New York, USA, AAI3338238.

Perkins, C., Royer, E., 1999. Ad-hoc on-demand distance vector routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999, pp. 90–100. http://dx.doi.org/10.1109/MCSA.1999.749281.

Pirzadeh, A., Snyder, W., 1990. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In: Proceedings of the IEEE International Conference on Robotics and Automation, May 1990, vol. 3, pp. 2113–2119. http://dx.doi.org/10.1109/ROBOT.1990.126317.

Ranjbar-Sahraei, B., Weiss, G., Nakisaee, A., 2012a. An adaptive stigmergic coverage approach for robot team. In: 24th Benelux Conference on Artificial Intelligence (BNAIC), pp. 210–217.

Ranjbar-Sahraei, B., Weiss, G., Nakisaee, A., 2012b. Stigmergic coverage algorithm for multi-robot systems (demonstration). In: Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS), vol. 3, pp. 1497–1498.

Ranjbar-Sahraei, B., Weiss, G., Nakisaee, A., 2012c. A multi-robot coverage approach based on stigmergic communication. In: Multiagent System Technologies. In: Lecture Notes in Computer Science, vol. 7598. Springer, Berlin, pp. 126–138, ISBN: 978-3-642-33689-8.

Ranjbar-Sahraei, B., Alers, S., Staňková, K., Tuyls, K., Weiss, G., 2013a. Towards soft heterogeneity in robotic swarms. In: Proceedings of the 25th Benelux Conference on Artificial Intelligence (BNAIC), pp. 384–385.

Ranjbar-Sahraei, B., Alers, S., Tuyls, K., Weiss, G., 2013b. Stico in action. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems. International Foundation for Autonomous Agents and Multiagent Systems, St. Paul, MN, USA., pp. 1403–1404.

Ranjbar-Sahraei, B., Staňková, K., Tuyls, K., Weiss, G., 2013c. Stackelberg-based coverage approach in nonconvex environments. In: Proceedings of the 12th European Conference on Artificial Life, vol. 12, pp. 462–469.

Ranjbar-Sahraei, B., Weiss, G., Tuyls, K., 2013d. A macroscopic model for multi-robot stigmergic coverage. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems. International Foundation for Autonomous Agents and Multiagent Systems, St. Paul, MN, USA., pp. 1233–1234.

Ranjbar-Sahraei, B., Bou-Ammar, H., Bloembergen, D., Tuyls, K., Weiss, G., 2014. Theory of cooperation in complex social networks. In: 28th AAAI Conference on Artificial Intelligence.

Roberts, M.B.V., 1986. Biology: A Functional Approach. Nelson, London, ISBN: 0174480199. http://www.worldcat.org/isbn/0174480199.

Şahin, E., 2005. Swarm robotics: from sources of inspiration to domains of application. In: Swarm Robotics. Springer, Berlin, pp. 10–20.

Sahoo, R.R., Singh, M., Sahoo, B.M., Majumder, K., Ray, S., Sarkar, S.K., 2013. A light weight trust based secure and energy efficient clustering in wireless sensor network: honey bee mating intelligence approach. In: Procedia Technology.In: First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013, vol. 10. pp. 515–523. http://dx.doi.org/10.1016/j.protcy.2013.12.390.http://www.sciencedirect.com/science/article/pii/S2212017313005525.

Saleem, M., Farooq, M., 2007. Beesensor: a bee-inspired power aware routing protocol for wireless sensor networks. In: Giacobini, M. (Ed.), Applications of Evolutionary Computing. In: Lecture Notes in Computer Science, vol. 4448. Springer, Berlin, Heidelberg, ISBN: 978-3-540-71804-8, pp. 81–90. http://dx.doi.org/10.1007/978-3-540-71805-5_9.

Schwager, M., Rus, D., Slotine, J.J., 2009. Decentralized, adaptive coverage control for networked robots. Int. J. Robot. Res. 28 (3), 357–375.

Schwager, M., Rus, D., Slotine, J.J., 2011. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. Int. J. Robot. Res. 30 (3), 371–383. http://dx.doi.org/10.1177/0278364910383444.

Senthilkumar, J., Chandrasekaran, M., 2011. Improving the performance of wireless sensor network using bee's mating, intelligence. Eur. J. Sci. Res. 55, 452–465.

Staňková, K., Ranjbar-Sahraei, B., Weiss, G., Tuyls, K., 2013. Staco: Stackelberg-based coverage approach in robotic swarms. In: The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE).

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

Wedde, H.F., Farooq, M., Zhang, Y., 2004. Beehive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L., Mondada, F., Stützle, T. (Eds.), Ant colony optimization and swarm intelligence. In: Lecture Notes in Computer Science, vol. 3172. Springer, Berlin, Heidelberg, pp. 83–94, ISBN: 978-3-540-22672-7. http://dx.doi.org/10.1007/978-3-540-28646-2_8.

Willow Garage, 2014. Turtlebot. https://www.willowgarage.com/turtlebot.

Yanovski, V., Wagner, I.A., Bruckstein, A.M., 2003. A distributed ant algorithm for efficiently patrolling a network. Algorithmica 37, 165–186.

Ziparo, V., Kleiner, A., Marchetti, L., Farinelli, A., Nardi, D., 2007. Cooperative exploration for USAR robots with indirect communication. In: Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'07).