Neural Comput & Applic (2019) 31:263–286
DOI 10.1007/s00521-017-2998-4

ORIGINAL ARTICLE

Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption

Nunzia Palmieri[1] · Xin-She Yang[2] · Floriano De Rango[1] · Salvatore Marano[1]

Received: 27 July 2016 / Accepted: 4 April 2017 / Published online: 4 May 2017

Abstract Many applications, related to autonomous mobile robots, require to explore in an unknown environment searching for static targets, without any *a priori* information about the environment topology and target locations. Targets in such rescue missions can be fire, mines, human victims, or dangerous material that the robots have to handle. In these scenarios, some cooperation among the robots is required for accomplishing the mission. This paper focuses on the application of different bio-inspired metaheuristics for the coordination of a swarm of mobile robots that have to explore an unknown area in order to rescue and handle cooperatively some distributed targets. This problem is formulated by first defining an optimization model and then considering two sub-problems: exploration and recruiting. Firstly, the environment is incrementally explored by robots using a modified version of ant colony optimization. Then, when a robot detects a target, a recruiting mechanism is carried out to recruit a certain number of robots to deal with the found target together. For this latter purpose, we have proposed and compared three approaches based on three different bio-inspired algorithms (Firefly Algorithm, Particle Swarm Optimization, and Artificial Bee Algorithm). A computational study and extensive simulations have been carried out to assess the behavior of the proposed approaches and to analyze their performance in terms of total energy consumed by the robots to complete the mission. Simulation results indicate that the firefly-based strategy usually provides superior performance and can reduce the wastage of energy, especially in complex scenarios.

Keywords Multi-robot systems · Swarm intelligence · Energy consumption · Nature-inspired algorithms · Metaheuristics

✉ Nunzia Palmieri
n.palmieri@dimes.unical.it

Xin-She Yang
x.yang@mdx.ac.uk

Floriano De Rango
derango@dimes.unical.it

Salvatore Marano
marano@dimes.unical.it

1 Department of Computer Engineering, Modeling, Electronics and Systems Science, University of Calabria, Rende, Italy

2 School of Design Engineering and Mathematics, Middlesex University, London, UK

1 Introduction

With the increasing importance of mobile robots in many critical applications, the study of multi-robot systems has grown significantly in size and intensity in recent years. In applications that are too risky for humans, multi-robot systems can play a crucial role to perform such critical tasks. Possible applications include planetary exploration, urban search and rescue mission, environmental monitoring, air traffic control, surveillance, and cleaning of disastrous materials [1]. The main goal is to coordinate a swarm of robots in such a way that some predefined global objectives can be achieved more efficiently. A particularly interesting situation is when all the mobile robots have no *a prior* information about the environment or target's locations, and these robots have to cooperate for finding the targets and then dealing with them jointly. In this work, we will focus on first exploration and then robot coordination. We suppose that a target can be detected by proper sensors but we will not focus on the details of such sensors. Since we are interested in how to provide a communication system, our emphasis will be on how to achieve a cooperative behaviour so as to perform the mission with a decisions mechanism under the assumption that the information about the environment for each robot is only partially available.

This paper first proposes different approaches based on three different bio-inspired algorithms and then carries out a comparison of bio-inspired metaheuristics applied to a swarm of robot that have to complete a mission with the objective to minimize the energy consumption. Energy limitation is one of the most important challenges for mobile robots. A robot is usually comprised of multiple components such as motors, sensors, controllers, and embedded computers. The energy consumption is related to the physical and mechanical structure of the robots and their abilities for moving, rotating, and sensing. The power consumption of a robot can be divided into motion, power, sensing power, control power, and computation power accordingly. Batteries are often used to provide power in mobile robots; however, they are heavy to carry and have a limited energy capacity.

Previous studies indicate that sensing, computation, and communication can consume a significant amount of power [2]. In order to minimize the energy consumed by robots to complete the assigned tasks, multi-robot algorithms should ideally have the following characteristics: distributed among many robots, computationally simple, low communication traffic, and scalable. Furthermore, the swarm of robots should be able to adapt and cooperate towards a low energy consumption rate energy, despite the limited sensing and communication abilities of the individual robots and the simple local interaction rules [3]. At the same time, the swarm should be able to complete the required tasks and achieve the objectives in the most efficient way.

One of the key issues is how to specify the rules of behavior and interactions at the level of an individual robot in order to minimize unnecessary movements, turning, and communication that can cause significant energy consumption. In this paper, the problem is first divided into two major phases: exploring the area for searching targets and targets resolving. The proposed approaches related to each phase form the main contributions of this work.

The exploration stage aims to explore the region and detect some targets distributed randomly in an unknown area and this is mainly implemented through an ant based strategy. In nature, ants deposit a specific type of chemical substance (pheromone) in the terrain while they are moving [4]. There are different types of pheromone, each of which is associated with a particular meaning and thus enables the ants to make decisions [5]. We use the pheromone to guide the robots during exploration. Using this approach, it is assumed that the robots do not know their positions and the positions of the others in the area, but they move according to what they can sense into the environment. When a robot detects a target during the exploration phase, it becomes a coordinator for this target and it starts to initiate a recruitment process so as to attract other robots. This coordinator robot, together with recruited robots, will perform the handling of the found target to make it safe cooperatively.

For this purpose, the coordinator robot uses a wireless communication sending out help requests through packets to its neighbors. The robots, which receive the help requests, choose in autonomous and individual manner if and what target they eventually go to. The recruiting task occurs in real time as soon as the targets are found. Three bio-inspired metaheuristic approaches are proposed as a decision mechanism for the recruited robots. Therefore, the aim of this paper is to evaluate and then compare these techniques, which provides some insight into how a group of robots can respond to a task of demands effectively in terms of total energy consumed by the swarm. One approach is to use the strategy based on the Firefly Algorithm [6], inspired by the flashing behaviour of tropical fireflies. The other methods are Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC), and they are inspired by social behavior of bird flocking [7] or fish schooling and the social behaviour of honey bees [8], respectively.

Therefore, the remainder of the paper is organized as follows. Section 2 provides a review of the related work. The description of the problem is the focus of Section 3. Section 4 and Section 5 describe the essence of the bio-inspired exploration algorithms used and the recruiting approaches, respectively. Section 6 presents the simulation results obtained from a set of experiments and finally Section 7 draws the main research conclusions.

2 Related work

Coordination of multi-robot systems has received much attention in recent years due to its vast potential in real-world applications. Simple robots work together to accomplish some tasks. In order to maximize the benefits from the cooperation among robots, a good coordination strategy is essential. The communication among the swarm is inevitable when the robots cooperate with each other, and it is the core part for controlling swarm behaviours. Robots' coordination strategies can be broadly divided into two main categories: explicit coordination and implicit coordination.

Explicit coordination refers to the direct exchange of information between robots, which can be made in the form of the unicast or broadcast of intentional messages. This often requires a dedicated on-board communication module. Existing coordination methods are mainly based on the use of explicit communication that allows the accuracy of the exchange of information among the swarm. However, such communication implies a waste of resources that can lead to the deterioration of the overall performance of the robot system. Instead, implicit coordination is usually associated with implicit communication, which requires the explorative robots to perceive, model, and reason others' behavior. In this case, an individual robot makes independent decisions on how to behave, based on the information it gathers through its own perception with others. When the robots use an implicit communication to coordinate, although the information obtained by the robots is not completely reliable, the stability, reliability, and fault tolerance of the overall system can be improved [9, 10].

Bio-inspired algorithms for modelling self-organizing robot systems have been proposed in recent years, inspired by a variety of biological systems. One of the well known is inspired by the collective behaviour of insect colonies such as ants and fireflies [4, 6]. These algorithms emphasise on decentralised local control, local communication, and on the emergence of global behaviour as the result of self-organization. Ant and other social animals are known to produce chemical substances called pheromone and use them to mark the paths in the environment that are used as a medium for sharing information. Pheromone trails provide a type of distributed information that artificial agents may use to make decisions. Many works can be found in the literature using this kind of biology metaphor [5, 11, 12]. Chemical trail-following strategies have been implemented with real robots. For example, ethanol trails were deposited and followed by the robots in Fujisawa et al. [13], but the use of decaying chemical trails by real robots can be problematic. Other robotic implementations of insect-style pheromone trail following have instead used non-chemical substitutes for the trail chemicals. For example, Garnier et al. [14] used a downward-pointing LCD projector mounted above their robots' arena to project light trails onto the floor. Other works that apply this similar approach were presented in [15–17].

In essence, most of the nature-inspired approaches use a combination of stochastic components or moves with some deterministic moves so as to form a multi-agent system with evolving states. Such a swarming system evolves and potentially self-organizes into a self-organized state some emergent characteristics. Another well-known bio-inspired approach takes inspiration from the behaviour of the birds, called Particle Swarm Optimization (PSO). PSO-inspired methods have received much attention in recent years. Pugh and Martinoli [18] applied an adapted version of PSO learning algorithm to carry out unsupervised robotic learning in groups of robots with only local information. Masár [19] proposed a modified version of PSO for the purpose of space exploration. Hereford and Siebold [20] presented a version of PSO for finding targets in the environment. A modified version of this algorithm is a robotic Darwinian-PSO approach by mimicking natural selection using the principles of social exclusion and inclusion (i.e., adding and removing robots to swarms) [21]. Another nature-inspired algorithm called Bees Algorithm (BA) that mimics the food foraging behaviour of swarms of honey bees and its modified versions has also been applied to robotic systems, demonstrating aggregation [22] and collective decision-making [23, 24].

Other studies take inspiration from the chemotactic behaviour of bacteria such as the *Escherichia coli*, called Bacterial Foraging Optimization (BFO). Bacteria movements mainly consist of two mobile behaviours: run in a particular direction and tumble to change its direction [25]. Such behaviour depends on the nutrient information around them. Yang et al. [26] applied this method for a target search and trapping problem. An extensive review of research related to the bio-inspired techniques and the most behaviour of the robots can be found in [9, 27]. Regarding the energy consumption problem, researchers have approached this problem in different ways, including minimizing the weight of robots, pre-positioning energy sources into the environments, minimizing communication ranges of robots, sending data in a simple form [28], reducing the direct communication and the use of multi-hop communication links between robots [29], and minimizing the distance of the traveling path [30, 31]. For example, Barca et al. addressed the problems related to energy consumption in [32].

In this paper, we apply bio-inspired algorithms to investigate the self-organization in a swarm of robots for targets searching. A combination of indirect communication and direct local communication is used to minimize the total energy consumed by the swarm. The main contributions of our work can be summarized as follows:

1. The mathematical formulation of the optimization model is presented with the objective to minimize the total energy consumed by a swarm of robots for exploring an unknown area and dealing with the found targets.
2. Development of energy models of mobile robots consisting of multiple components.
3. A combination of indirect and direct communication to execute the tasks:
   - Indirect communication is used for the exploration task, based on the repulsion behavior of the robots towards the pheromone deposited into the visited cells. This mechanism is the same for all recruitment strategies.
   - A direct local communication mechanism is used in terms of a wireless medium for the coordination of the robots in the recruiting process. For this

purpose, three bio-inspired algorithms are used and compared in order to evaluate the performance in terms of energy consumptions.

3 Problem statement

Let us consider the following swarm scenario. There are a number of targets scattered in an unknown area, according to a uniform distribution. A swarm of mobile robots are deployed in this area with the goal to explore the area for searching the targets and then removing/dismantling them cooperatively. Since it is either impossible or too expensive for a single robot to handle a target individually, it is necessary that when a robot detects a target, a coalition of some robots has to be formed to perform the removal task jointly. A coalition can handle a target only if the necessary robots are in the target's location. Moreover, it is assumed that there is no prior knowledge about the targets such as their total number and locations. Therefore, the only way to ensure the detection and the fulfillment of all targets is to explore the overall area. Since, the targets' locations are detected gradually through searching, the recruitment task must start in real time as the targets are found. The challenge is to complete the mission without any centralized control and using only minimal local sensing and communication among the swarm of robots, and the main objective is to minimize the total energy consumed by the team.

Broadly speaking, we can divide the mission into two phases: exploring and recruiting. During the exploring phase, since no targets has been detected yet, it would be more efficient deploying, in a distributed manner, the robots in different regions of the area at the same time. At each step, a robot from the current location starts to sense its neighbor cells through some sensors in order to make the decision where to go next. In this phase, the robots do not use wireless communication, and the decisions are made by the robots on the basis of partial available knowledge about the environment. When a robot detects a target, since it lacks the capabilities to carry out the rest of the task itself, it starts a recruiting process using wireless communication in this case. The robots, receiving the signal, then make the decision to get involved or not through mechanisms inheriting the swarm intelligence principles. The aim is to distribute the robots into the environment and, at the same time, allocate a sufficient number of robots among target's locations, while avoiding redundancy. It is worth pointing out that the exploration and coordination tasks are not entirely decoupled; it is possible for a robot to perform both simultaneously for example when it moves towards the target's location, it also implicitly explores the area.

3.1 Assumptions of the model

First of all, the characteristics of the unknown area and the capabilities of the robots are introduced. Then, the problem is modeled as an optimization problem subject to constraints.

The environment is mapped as a 2D plane. As a symbolic representation of the working space, the proposed method uses a grid map $A \subset \mathbb{R}^2$ with m and n cells in the x and y direction, respectively. Each cell $c \in A$ is the basic element of the grid and it is uniquely determined by its coordinates (x, y), with $x \in \{1, 2, \ldots, m\}$ and $y \in \{1, 2, \ldots, n\}$ elements. In the area, a set R of homogeneous robots are developed where $R = \{ k \mid k \in \{1, 2, \ldots, N^R\}\}$ and, at each step t, the current state of a robot k can be represented by its coordinates (x_k^t, y_k^t).

As far as the characterization of the robots is concerned, we assume that they live in a discrete-time domain and they can move on a cell-by-cell manner; that is, one cell at a time. The movements of the robots in the area are described by changing their coordinates in time. They can visit all cells in the area except the cells occupied by an obstacle or other robots. We assume that a robot uses 45° as the unit for turning, since we only allow the robot to move from one cell to one of its eight neighbour cells, if all cells are free. Figures 1 and 2 show an example. However, for simplicity, it is assumed that the robots have a simple set of common reactive behaviours that enable them to avoid the obstacles and recognize the other robots in order to accomplish the mission together. They have limited computing and memory capacities and they are capable of discovering and partially executing the tasks.

In addition, it is also assumed that the robots are equipped with proper sensors to perceive, leave the pheromone, and detect the targets. They can mark the visited cells with pheromone and they can sense the level of the pheromone in their local neighborhood. They are able to self-localize themselves in the given area using some onboard equipment, such as GPS. Since a robot's communication range R_t cannot cover the whole area, the communication capability thus enables the robot to directly communicate to others within the communication range as shown in Fig. 3.

The robots must explore the area for searching and dealing with a set T of N^T static targets disseminated in the area, i.e., $T = \{ z \mid z \in \{1, 2, \ldots, N^T\}\}$. Each target is represented by its coordinates (x_z, y_z). A target z is detected by a robot k when the target's coordinates coincide with the robot's coordinates. Once a robot finds a target, it sends help requests through packets (that contain mainly the coordinates of the found target) to the robots into its wireless range R_t (see Fig. 3). We define RR_k as a set that keeps track of the help requests that the robot k receives, expressed in terms of targets, thus $RR_k \subset T$.

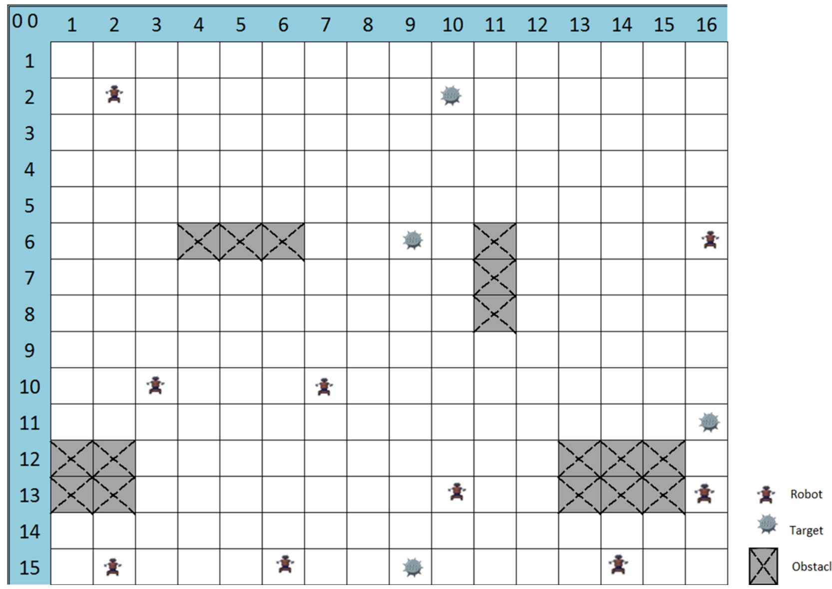


Fig. 1 A representation of the simulation environment

Moreover, the problem studied in this paper is based on the following assumptions: (1) the robots work in a static environment, (2) the number of targets is smaller than the number of the robots in order to avoid deadlock, (3) no changing or charging battery is required. All robots, at each step, follows simple behavioral rules, depicted in the Fig. 4 on the basis of the events that occur and described as follows:

– Explorer state: it is the initial state of each robot. At this state, the robots explore the area for target's detection and they can communicate with other members of the swarm through the environment (indirect communication).
– Coordinator state: a robot becomes a coordinator when it detects a target and it tries to recruit the necessary robots, by sending packets using a wireless communication module. The packets contain mostly the coordinates of the found target and they are received only by the neighbor's robots within its wireless range (see Fig. 3).

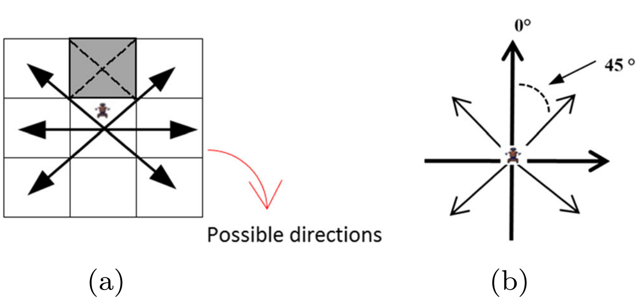


Fig. 2 **a** Possible robot's directions. **b** Possible robot's turning

– Recruited state: a robot switches to this state when it is recruited by one or more neighbor coordinators to accomplish a target, through the receiving of packets. Then, the robot will make the decision about where to move and what target to perform according to different bio-inspired algorithms such as the Firefly Algorithm, Particle Swarm Algorithm, and Artificial Bee Algorithm. A key aspect of this state occurs when robot k, that has received help requests by one or more coordinators, applying one of the bio-inspired coordination algorithms, moves too far from the target's position. Given a robot k located at the step t in the cell of coordinates (x_k^t, y_k^t) and the target z with coordinates (x_z, y_z), we define the distance between the robot k and the target z as the Euclidean distance $r_{kz} = \sqrt{(x_k^t - x_z)^2 + (y_k^t - y_z)^2}$. If $r_{kz} \geq (R_t + \Delta)\ \forall\, z \in RR_k$ means that the robot k moves too far from the target's locations and in this case, if it has not got other requests, it will change its states into Explorer state (see Fig. 5).
– Waiting state: a recruited robot, once reached the target's location, it has to wait until it receives the order by the coordinator to perform the target.
– Execution state: once all needed robots have reached the target's location, they can deal with the target for a defined time (it is regulated by a fixed timer).

The overall procedure and interchange of states can be summarized in the flowchart as shown in Fig. 6.

3.2 Mathematical model

In order to describe the proposed system as proper mathematical models, it is useful to introduce the following notations and definitions:

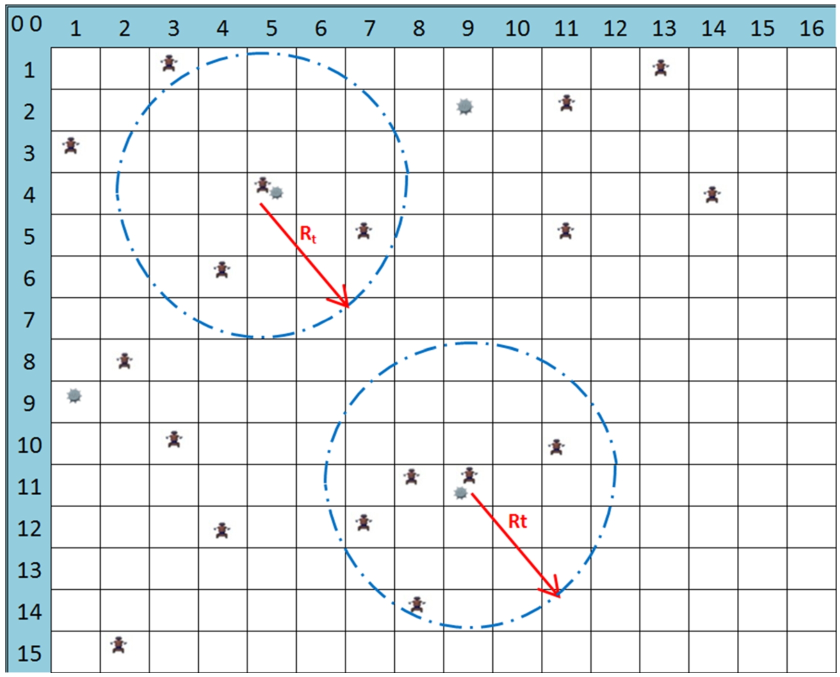


Fig. 3 The robots in the cells with coordinates (4,5) and (11,9) have detected a target. They start a recruitment process by sending packets that will be received by the robots within their wireless range R_t

- A: operational area, discretized as a grid map and $A \subset \mathbb{R}^2$
- R : set of robots
- N^R : number of robots $N^R = |\mathrm{R}|$
- N^R_{min} = number of robots needed to deal with a target
- S: set of recruited robots $S \subset R$
- T: set of targets
- N^T: number of targets, $N^T = |\mathrm{T}|$
- F: set of the found targets, $F \subset T$
- RR_k: set of help requests expressed in terms of found targets received by the robot k where $RR_k \subset F \subset T$.

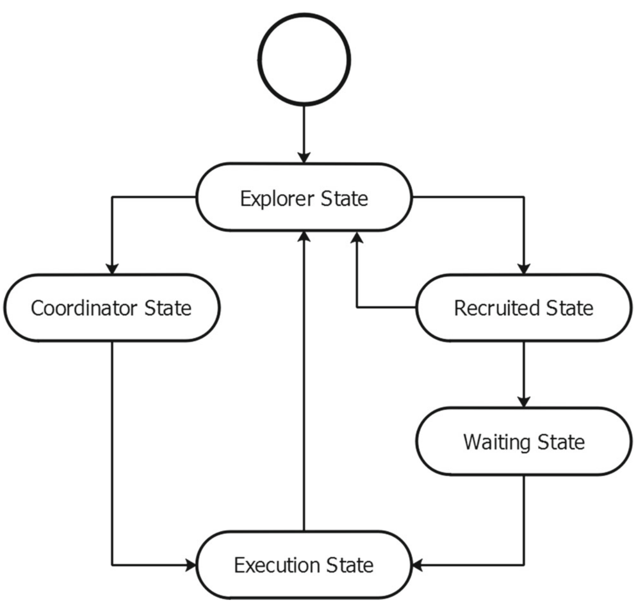


Fig. 4 Possible states of a robot in our proposal

Two main decisions have to be modelled properly. On the one hand, the position expressed by the coordinates where each robot $k \in R$ should be located at each step. On the other hand, given a robot k and a found target $z \in RR_k$, it has to decide if it is to get involved in the recruiting process of the found target z. The first decision is mathematically represented by the decision variables:

$$v^k_{xy} = \begin{cases} 1 & \text{if the robot } k \text{ visits the cell } (x, y), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, the following decision variables allow us to model if a robot k is involved in the recruitment process of the target z:

$$u^k_z = \begin{cases} 1 & \text{if robot } k \text{ is involved with target } z, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

For each activity executed by a robot k, a certain amount of energy is consumed. In our study, the energy model reflects mostly two activities: energy for communication and energy for mobility. The mobility energy depends on several factors. For simplicity, the mobility cost for a robot k in our model can be calculated by considering the distance traversed and it is expressed as follows:

$$E^k_m = \sum_{x=1}^{m} \sum_{y=1}^{n} C_m \, v^k_{xy}, \tag{3}$$

where $\sum_{x=1}^{m} \sum_{y=1}^{n} v^k_{xy}$ is the total number of visited cells for each robot k while moving in the exploration phase and recruiting phase; C_m is the cost given to move to one cell to another and takes into account both the cost for moving and turning.

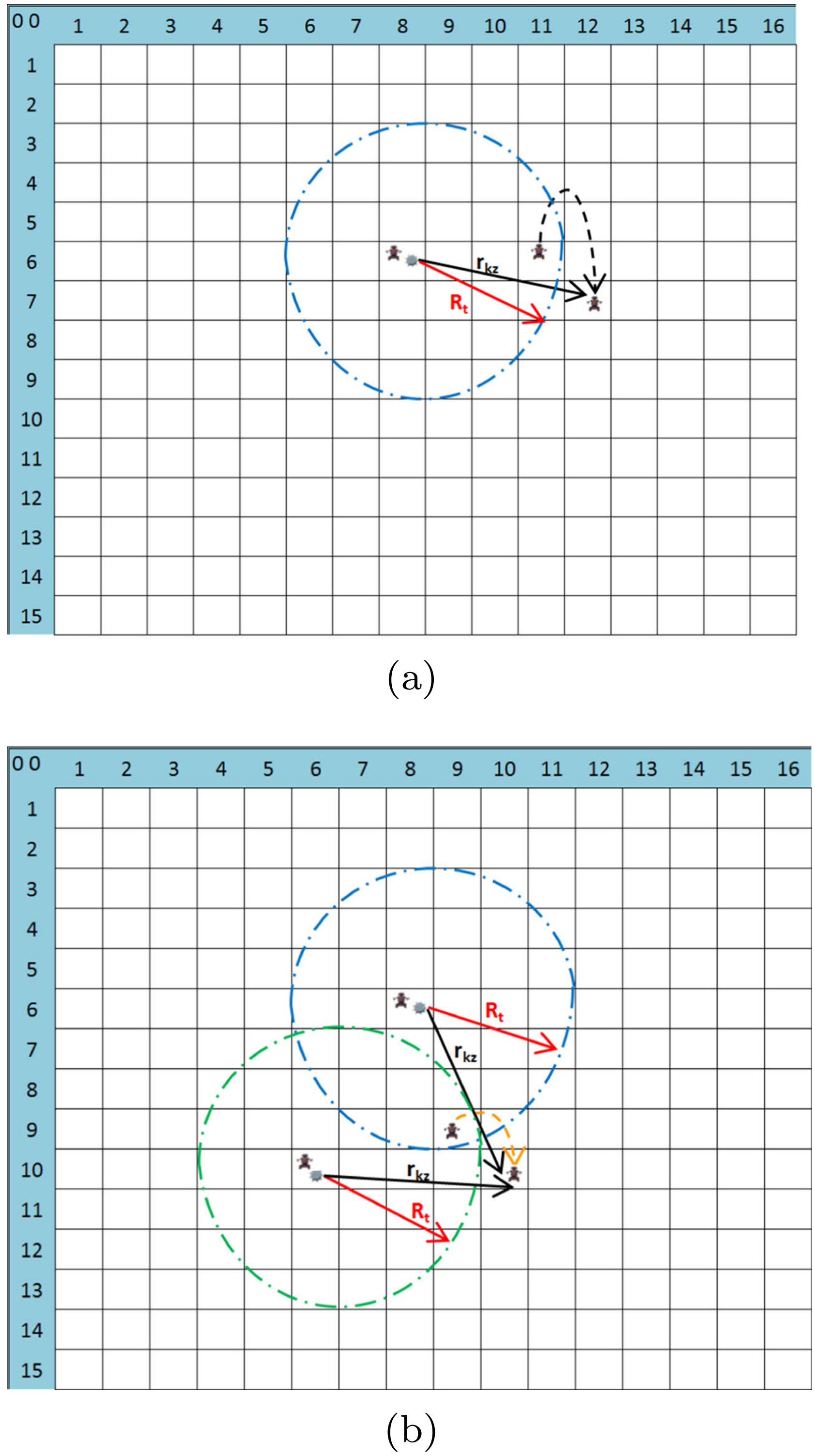


Fig. 5 **a** The robot in the cell (6,11) that is recruited by the robot in the cell (6,8) after the application of a coordination strategy moves into the cell too far from the target, thus it changes its state becoming an explorer. **b** The robot in the cell (9,9) that is recruited by two robots in the cells (6,8) and (10,6) respectively, after the application of a coordination strategy moves into the cell too far from both targets, thus it changes its state becoming an explorer

When a target is detected, the energy consumed is instead related to the communication and to the cost for performing the planned task on the target. Since we use a wireless communication system in this phase, the energy consumed depends on the transmission and reception of the packets to communicate the position of the targets. In this case, we assume that the energy consumed by robot k to transmit E^k_{tx} and receive E^k_{rx} a packet [33] is related to the maximum transmission range R_t and to the packet size (l) as follows:

$$E^k_{tx} = l\,(R_t^{\psi}\, e_{tx} + e_{cct}), \tag{4}$$

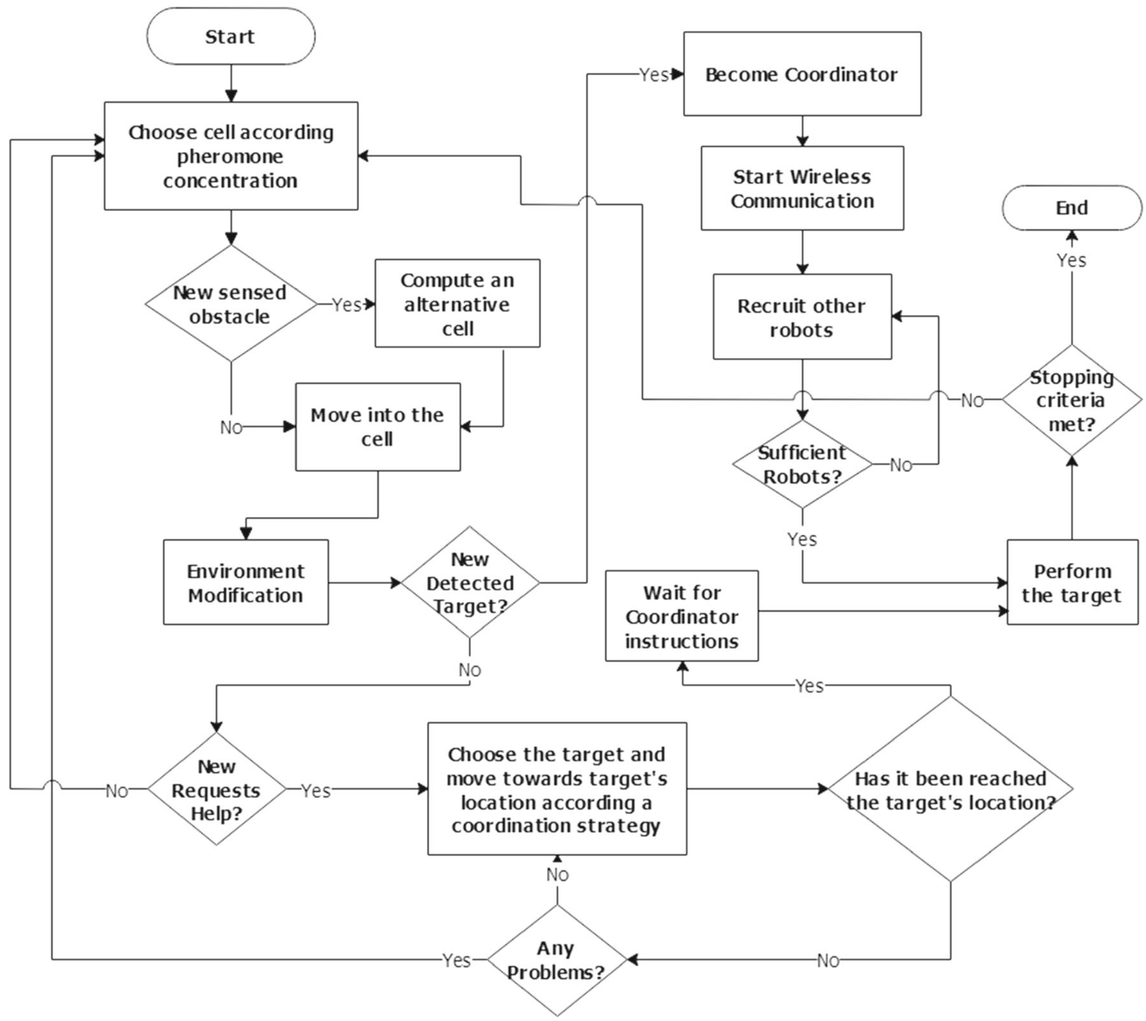


Fig. 6 Flowchart of the proposed model

where e_{tx} is the energy required by the power amplifier of transceiver to transmit one bit data over the distance of one meter, and e_{cct} is the energy consumed in the electronic circuits of the transceiver to transmit or receive one bit. Here, ψ is called the path loss exponent of the transmission medium where $\psi \in [2, 6]$.

On the other hand, the energy consumption for receiving a packet is independent of the distance between communication nodes and it is defined as:

$$E^k_{rx} = l\, e_{cct}, \tag{5}$$

The energy consumed to deal with a target is:

$$E^k_d = C_d, \tag{6}$$

where C_d is the cost given to the working task for handling a target properly, and it is the same for each robot and it is related, for simplicity, to the mechanical movement. Essentially, we model the energy consumed for the coordination task by the robot k that is involved in the targets issue as:

$$E^k_{coord} = \sum_{z=1}^{N^T} \left(E^k_{tx} + E^k_{rx} + E^k_d \right) u^k_z. \tag{7}$$

Based on the previous considerations and models, we now introduce a performance index, called Total-Energy-Swarm-Consumption (TESC), as:

$$TESC = \sum_{k=1}^{N^R} E^k_m + \sum_{k=1}^{N^R} E^k_{coord}. \tag{8}$$

That is, the total energy consumed by all the robots and it is the sum of two contributions: energy consumption for moving into the area and energy consumption for the wireless communication when they are involved in the performing of the targets.

3.3 Objective function and constraints

The optimization problem in this paper has an objective function related to the minimization of the overall energy consumption by the robot swarm to complete the mission. Thus, the optimization problem can be mathematically represented as follows:

$$\text{Minimize} \quad TESC = \sum_{k=1}^{N^R} E_m^k + \sum_{k=1}^{N^R} E_{coord}^k =$$

$$\sum_{k=1}^{N^R}\sum_{x=1}^{m}\sum_{y=1}^{n} C_m\, v_{xy}^k + \sum_{k=1}^{N^R}\sum_{z=1}^{N^T}(E_{tx}^k + E_{rx}^k + E_d^k)\, u_z^k, \tag{9}$$

subject to

$$\sum_{k=1}^{N^R} v_{xy}^k \geq 1 \qquad \forall\, (x, y) \in A, \tag{10}$$

$$\sum_{k=1}^{N^R} u_z^k = N_{min}^R \qquad \forall\, z \in T, \tag{11}$$

$$v_{xy}^k \in \{0, 1\} \qquad \forall\, (x, y) \in A, k \in R, \tag{12}$$

$$u_z^k \in \{0, 1\} \qquad \forall\, z \in T, k \in R. \tag{13}$$

The objective function in (9) to be minimized represents the total energy consumed by the swarm of robots. Constraint (10) ensures that each cell is visited at least once. Constraint (11) defines that each target z must be handled safely by N_{min}^R robots. The constraints (12)–(13) specify the domain of the decision variables. It is worth pointing out that the optimization problem here is intrinsically multi-objective, but we have formulated it as a single objective optimization problem. The main reason is that we will focus on the comparison of different bio-inspired approaches in solving this challenging problem. Future work will focus on the extension of the current approach to multi-objective optimization.

4 Exploration strategy

In our model, the robots are initially deployed in the environment, according to a uniform distribution. Some communication via environment (stigmergy) is used to share local knowledge on cells gained by individual robots. To minimize the revisit of visited cells, we introduce a repulsive pheromone mechanism into the swarm. During the exploration task, this pheromone is deposited, immediately when a robot reaches a new cell in order to mark all cells that has been visited. The use of pheromone is similar to the use in Ant Colony Optimization method, but unlike ants, the robots should search for the cells without any pheromone or with the smallest pheromone value. The pheromone deposited by a robot on a cell diffuses outwards cell-by-cell until a certain distance R_s such that $R_s \subset A \subset \mathbb{R}^2$ and the amount of the pheromone decreases as the distance from the robot increases (see Fig. 7).

The model for the pheromone diffusion is defined as follows: consider that robot k at iteration t is located in a cell

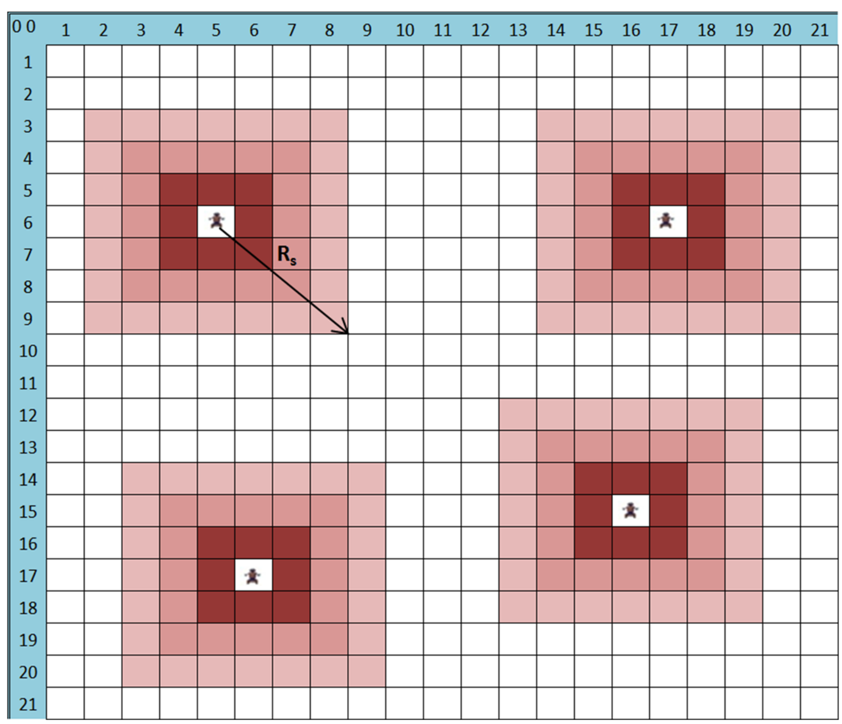


Fig. 7 Example of pheromone diffusion. When the robots move into new cells, they spread the pheromone within a certain distance R_s. The intensity of pheromone decays according to the distance from the cell

of coordinates $(x_k^t, y_k^t) \in A$, then the amount of pheromone that the kth robot deposits at the cell c of coordinates (x, y) is given by:

$$\Delta \tau_c^{k,t} = \begin{cases} \Delta \tau_0 \, e^{\frac{-r_{kc}}{a_1}} - \frac{\varepsilon}{a_2} & \text{if } r_{kc} \le R_s, \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where r_{kc} is the distance between the kth robot and the cell c and it is defined as:

$$r_{kc} = \sqrt{(x_k^t - x)^2 + (y_k^t - y)^2}. \tag{15}$$

In addition, $\Delta\tau_o$ is the quantity of pheromone sprayed in the cell where the robots is placed and it is the maximum amount of pheromone, ε is an heuristic value (noise) and $\varepsilon \in (0, 1)$. Furthermore, a_1 and a_2 are two constants to reduce or increase the effect of the noise and pheromone. It should be noted that multiple robots can deposit pheromone in the environment at same time, then the total amount of pheromone that can be sensed in a cell c depends on the contribution of many robots.

Furthermore, the deposited pheromone concentration is not fixed and can evaporate with the time. The rate of evaporation of pheromone is given by ρ, and the total amount of pheromone evaporated in the cell c at step t is given by the following function:

$$\xi_c^t = \rho \, \tau_c^t, \tag{16}$$

where τ_c^t is the total amount of the pheromone on the cell c at iteration t. For the calculation of ρ, we introduced a coefficient, called $ERTU_{\%}$ (Evaporation Rate Time Unit) that regards the evaporation rate per unit of time spent. Let the last time in which the cell has been visited be t_v and the current time t, $(t - t_v)$ is the time spent since the last visit of the cell. Multiplying this time per $ERTU_{\%}$, the percentage of substance that evaporates will be

$$\rho = (t - t_v) \, ERTU_{\%}. \tag{17}$$

Considering the evaporation of the pheromone and the diffusion according with the distance, the total amount of pheromone in the cell c at iteration t is given by

$$\tau_c^t = \tau_c^{(t-1)} - \xi_c^{(t-1)} + \sum_{k=1}^{N^R} \Delta \tau_c^{k,t}, \tag{18}$$

Each cell has an initial pheromone value set to zero that represents that the cell has not yet been visited by any of the robots.

4.1 Cells selection

Each robot k, at each step t, is placed on a particular cell c_k^t that is surrounded by a set of accessible neighbor cells $N(c_k^t)$. Essentially, each robot perceives the pheromone deposited into the nearby cells, and then it chooses which cell to move to at the next step. The probability at each step t for a robot k of moving from cell c_k^t to cell $c \in N(c_k^t)$ can be calculated by

$$p(c|c_k^t) = \frac{(\tau_c^t)^{\varphi} \, (\eta_c^t)^{\lambda}}{\sum_{b \in N(c_k^t)} (\tau_b^t)^{\varphi} \, (\eta_b^t)^{\lambda}}, \quad \forall \, c \in N(c_k^t) \tag{19}$$

where $(\tau_c^t)^{\varphi}$ is the quantity of pheromone in the cell c at iteration t, and $(\eta_c^t)^{\lambda}$ is the heuristic variable to avoid that the robots being trapped in a local minimum. In addition, φ and λ are two constant parameters which balance the weight to be given to pheromone values and heuristic values, respectively. The robot k moves into the cell that satisfies the following condition:

$$c = \min[p(c|c_k^t)]. \tag{20}$$

In this way the robots will prefer less frequented areas and is more likely to direct towards an unexplored region. The exploration strategy was previously validated in [11] and essentially the structure is given by the Algorithm 1.

Algorithm 1 Exploration algorithm inspired by ant colony optimization

```
begin
  Step 1: Initialization.
          Set t: {t is the step counter}.
          Define φ, λ, a1, a2, ϵ, Δτ0 and ERTU%,
          Rs
  Step 2: Generation coordination system. For
          the whole swarm, set the initial locations in
          terms of coordinates in x and y directions.
  Step 3: Procedure
  while the stop criteria are not satisfied do
     for each robot k in Explorer State (k ∈ R) do
        evaluate the current position c_k^t
        evaluate neighboorhood N(c_k^t)
        compute c according (20)
        if (c.hasObstacle() or c.isOccupated()) then
            choose a random cell c* ∈ N(c_k^t)
            move robot k towards c*;
        else
            move robot k towards c;
        end if
     end foreach
     foreach cell c ∈ A do
        update pheromone according (18);
     end foreach
     update t;
  end while
end
```

The Algorithm 1 is an iterative process. At the first iteration, each cell has the same value of the pheromone trail, so that the initial probability that a cell would be chosen is almost random. Then, the robots move from a cell to another based on rules expressed in (19)–(20). The pheromone trails on the visited cells by robots are updated according to (18) and unvisited cells become more attractive to the robots. The objective is to avoid any overlapping and redundancy efforts in order to save energy and complete the mission as quickly as possible. Regarding the energy consumption, the energy consumed by each robot is related to the mobility according to (3). The algorithm stops executing when a robot k becomes a coordinator or a recruited or if the mission is completed (that is, all cells are visited and all targets are found and performed).

5 Recruitment strategies

When a robot detects a target, it starts a recruiting process in order to handle it cooperatively. For this purpose, wireless communication is used as a coordination mechanism. In this case, each robot is assumed to have transmitters and receivers, using which it can send packets to other robots within its wireless range R_t and there is no propagation of the packets (one hop communication) as shown in Fig. 3. The packets contain mostly the coordinates of the detected targets. Therefore, the volume of information that is communicated among the robots is small, but the robots lack global knowledge of the environment. It is worth mentioning that the decisions to be made by the robots are independent, and the robots and the coordinator do not know which robots are arriving, so the coordinator will continue to send packets until the needed robots have actually arrived. This happens because the decision mechanism is dynamic and it depends on what the robots decide individually.

5.1 Firefly-based team strategy for robots recruitment (FTS-RR)

Firefly Algorithm (FA) is a nature-inspired stochastic global optimization method that was developed by Yang [6, 34]. FA tries to mimic the flashing behaviour of a swarm of fireflies. In the algorithm, the two important issues are the variations of light intensity and the formulation of attractiveness. The brightness of a firefly is determined by the landscape of the object function. Attractiveness is proportional to the brightness and, thus, for any two flashing fireflies, the less bright one will move towards the brighter one. In addition, the light intensity decays with the square of the distance, so the fireflies have limited visibility to other fireflies. This plays an important role in the communication of the fireflies and the attractiveness, which may be impaired by the distance. The distance between any two fireflies i and j, at positions x_i and x_j, respectively, can be defined as the Euclidean distance as follows:

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{d=1}^{D}(x_{i,d} - x_{j,d})^2}, \tag{21}$$

where $x_{i,d}$ is the dth component of the spatial coordinate x_i of the ith firefly and D is the number of dimensions. In 2-D case, we have

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{22}$$

In the firefly algorithm, as the attractiveness function of a firefly j varies with distance, one should select any monotonically decreasing function of the distance to the chosen firefly. For example, we can use the following exponential function:

$$\beta = \beta_0 \, e^{-\gamma r_{ij}^2}, \tag{23}$$

where r_{ij} is the distance defined as in (21), β_0 is the initial attractiveness at the distance $r_{ij} = 0$, and γ is an absorption coefficient at the source which controls the decrease of the light intensity. The movement of a firefly i which is attracted by a more attractive (i.e., brighter) firefly j is governed by the following evolution equation:

$$x_i^{t+1} = x_i^t + \beta_0 \, e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha\left(\sigma - \frac{1}{2}\right), \tag{24}$$

where the first term on the right-hand side is the current position of the firefly i, the second term is used for modelling the attractiveness of the firefly as the light intensity seen by adjacent fireflies, and the third term is randomization with α being the randomization parameter and it is determined by the problem of interest. Here, σ is a scaling factor that controls the distance of visibility and in most cases we can use $\sigma = 1$. The convergence and stability require good parameter settings, as it is true for almost all meta-heuristic algorithms [35]. Previous studies investigated the influence of algorithm-dependent parameters on the convergence of the strategy [36]. We adapted this strategy for our problem. In particular, when a robot detects a target, it becomes a coordinator and it tries to attract the other robots (like a firefly), on the basis of the target's position, in order to handle the target in a cooperative manner.

The original version of FA is applied in the continuous space and cannot be applied directly to tackle discrete problems, so we have modified the algorithm in order to solve our problem. In our case, a robot can move in a 2-D discrete space and it can go just in the adjacent cells. This means that when a robot k, at iteration t, in the cell c_k^t with coordinates (x_k^t, y_k^t) receives a packet by a coordinator robot that has found a target, this robot will move in the next step $(t+1)$ to a new position (x_k^{t+1}, y_k^{t+1}), according to the FA attraction rules such as expressed below:

$$\begin{cases} x_k^{t+1} = x_k^t + \beta_0\, e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}), \\ y_k^{t+1} = y_k^t + \beta_0\, e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}), \end{cases} \tag{25}$$

where x_z and y_z represent the coordinates of the selected target translated in terms of row and column of the matrix area, r_{kz} is the Euclidean distance between the target z and the recruited robot. It should be noticed that the targets are static and a robot can receive more than one request. In the latter case, it will choose to move towards the brighter target within the minimum distance from the target as expressed in (23). A robot's movement is conditioned by target's position and by a random component that it is useful to avoid the situation that more recruited robots go towards the same target if more targets have found. This last condition enables to the algorithm to jump out of any local optimum (Fig. 8).

It is worth mentioning that $r_{kz} \leq (R_t + \Delta)$. This last condition ensures that if the robot k, during the movement for reaching the selected target z, chooses a cell too far from the target's location $(R_t + \Delta)$ where Δ is a perturbation coefficient, it switches its role and continues to explore the area (Fig. 5).

In order to modify the FA to a discrete version, the robot movements have been modelled by three kinds of possible value updates for each coordinates { -1, 0, 1 }, according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) > 0], \\ x_k^{t+1} = x_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) < 0], \\ x_k^{t+1} = x_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) = 0], \end{cases} \tag{26}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) > 0], \\ y_k^{t+1} = y_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) < 0], \\ y_k^{t+1} = y_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) = 0]. \end{cases} \tag{27}$$

A robot (e.g., robot k) that is in the cell with coordinates (x_k^t, y_k^t) as depicted in Fig. 9 can move into eight possible cells according to the three possible values attributed to x_k and y_k. For example, if the result of (26)–(27) is (-1, 1), the robot will move into the cell $(x_k^t - 1, y_k^t + 1)$.

In the described problem, the algorithm for the Firefly-based strategy is shown in Algorithm 2.

Algorithm 2 FTS-RR strategy

```
begin
    Step 1: Initialization. Set t {t is the step counter};
            Set the detected targets z ∈ F, the wireless
            range R_t, and the robots in wireless range of
            the detected targets k ∈ S. Define the light
            absorption coefficient γ, the randomization
            parameter α, the random number σ and the
            attractiveness β_0.
    Step 2: Generation coordination system. For
            the detected targets and the recruited
            robots, set the initial locations in terms of
            coordinates in x and y directions.
    Step 3: Procedure.
    while The stop criteria are not satisfied do
        foreach robot k in Recruited State (k ∈ S) do
            set RR_k
            evaluate the current position c_k^t
            foreach target z ∈ RR_k do
                evaluate β according to (23)
                choose the best target z
            end foreach
            evaluate N(c_k^t)
            compute the cell c_k^{t+1} according to
            (26)–(27)
            if (c_k^{t+1}.hasObstacle() or
             c_k^{t+1}.isOccupated()) then
                choose a random cell c* ∈ N(c_k^t);
                move robot k towards c*;
            else
                move robot k towards c_k^{t+1};
            end if
        end foreach
        update t;
    end while
end
```

The Algorithm 2 is executed when one or more targets are found and some robots are recruited by others. If no targets are detected or all targets are removed or handled, the robots perform the exploration task according to Algorithm 1. This happens because the nature of the problem is bi-objective and the robots have to balance the two tasks. In

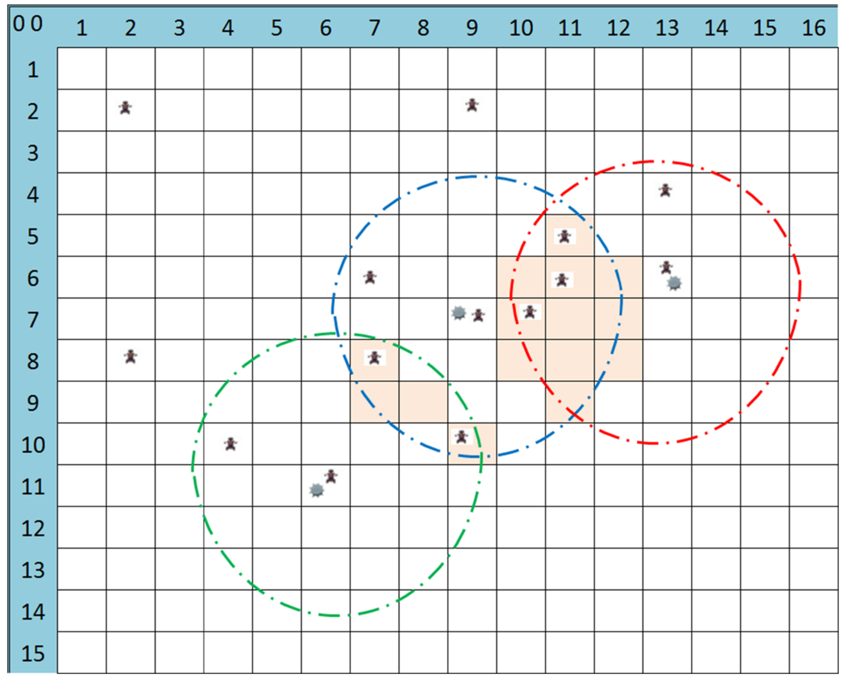


Fig. 8 Example of an overlap region in which some robots are in the wireless ranges of different coordinator robots and thus they must decide towards which target to move, according to a bio-inspired strategy

this case, the energy model comprises the mobility cost for reaching the target's locations, communication cost for the transmission and reception of the packets to communicate the position of the found targets, and the cost related to the processing of a target (3–6).

5.2 Particle swarm optimization for robot recruitment (PSO-RR)

Particle Swarm Optimization (PSO) is an optimization technique which uses a population of multiple agents [7]. This technique was inspired by the movement of flocking birds and their interactions with their neighbours in the swarm. Each particle i moves in the search space and has a velocity v_i^t and a position vector x_i^t. A particle updates its velocity according to the best previous positions and the global best position achieved by its neighbours:

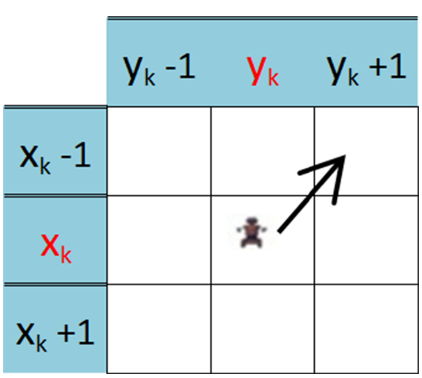


Fig. 9 A possible selected cell after the application of a bio-inspired strategy

$$v_i^{t+1} = \omega v_i^t + r_1 c_1 (g_{best} - x_i^t) + r_2 c_2 (p_{best} - x_i^t), \qquad (28)$$

where the individual best value is the best solution has been achieved by each particle so far that is called p_{best}. The overall best value is the best value (best position with the highest fitness function) that is found among the swarm, which is called g_{best}. Here, r_j $(j = 1, 2)$ are the uniformly generated random numbers between 0 and 1, while ω is the inertial weight and c_j $(j = 1, 2)$ are the acceleration coefficients. In addition, (28) is used to calculate the new velocity v_i^{t+1} of a particle using its previous velocity v_i^t and the distances between its current position and its own best found position; that is, its own best experience p_{best} and the swarm global best g_{best}. The new position of particle i are calculated by

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \qquad (29)$$

However, like Firefly Algortihm, directly using this PSO-based decision strategy in our recruiting task would be problematic. Firstly, on the 2D map, there are only a limited number of possible directions for the robots to move and since we assumed that the robots can only move one cell at a time, the next position of the particles (robots) is limited to the neighbour cells as shown in Fig. 2. Moreover, in the recruiting phase, we are interested in reaching the target location (that is our g_{best}) and we do not take into account p_{best} of the robots. Therefore, a modified PSO version is proposed and this means that for each robot k at iteration t in a cell with coordinates (x_k^t, y_k^t), Eqs. (28)–(29) can be written as the follows:

$$\begin{cases} v_{x_k}^{t+1} = \omega v_{x_k}^t + r_1 c_1 (x_z - x_k^t), \\ v_{y_k}^{t+1} = \omega v_{y_k}^t + r_1 c_1 (y_z - y_k^t), \end{cases} \tag{30}$$

$$\begin{cases} x_k^{t+1} = x_k^t + v_{x_k}^{t+1}, \\ y_k^{t+1} = y_k^t + v_{y_k}^{t+1}, \end{cases} \tag{31}$$

where (x_z, y_z) represent the coordinates of the detected target translated in terms of row and column of the matrix area. In order to modify the PSO to a discrete version, similar to the case of the FA, the robot movements have been considered as three possible value updates for each coordinates:{ -1, 0, 1 } according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [v_{x_k}^{t+1} > 0], \\ x_k^{t+1} = x_k^t - 1 & \text{if } [v_{x_k}^{t+1} < 0], \\ x_k^{t+1} = x_k^t & \text{if } [v_{x_k}^{t+1} = 0], \end{cases} \tag{32}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [v_{y_k}^{t+1} > 0], \\ y_k^{t+1} = y_k^t - 1 & \text{if } [v_{y_k}^{t+1} < 0], \\ y_k^{t+1} = y_k^t & \text{if } [v_{y_k}^{t+1} = 0]. \end{cases} \tag{33}$$

When a robot receives more requests, it will choose to move toward the target at the minimum distance. In the described problem, the Particle Swarm Algorithm is executed as shown in Algorithm 3. Like FA, the steps are executed when the robots are recruited by others, but in the case when no targets are detected or all targets are handled, the robots continue to explore the area. Similarly, the energy model comprises the mobility cost for reaching the target's locations, communication cost for the transmission and reception of the packets to communicate the position of the found targets and the cost related to the processing of a target (3–6).

Algorithm 3 PSO-RR strategy

begin
  Step 1: **Initialization.** Set t {t is the step counter}; set the detected targets $z \in F$, the wireless range R_t, and the robots in wireless range of the detected targets $k \in S$. Define the inertia weight ω, randomization parameter r_1 and acceleration coefficient c_1
  Step 2: **Generation coordination system.** For the detected targets and the recruited robots, set the initial locations in terms of coordinates in x and y directions.
  Step 3: **Procedure.**
  while The stop criteria are not satisfied **do**
    foreach robot k in Recruited State ($k \in S$) **do**
      set RR_k
      evaluate the current position c_k^t
      foreach target $z \in RR_k$ **do**
        choose the best target z
      end foreach
      evaluate $N(c_k^t)$
      compute the cell c_k^{t+1} according (32)–(33)
      if (c_k^{t+1}.hasObstacle() or c_k^{t+1}.isOccupated()) **then**
        choose a random cell $c^* \in N(c_k^t)$;
        move robot k towards c^*;
      else
        move robot k towards c_k^{t+1};
      end if
    end foreach
    update t;
  end while
end

5.3 Artificial bee colony algorithm for robot recruitment (ABC-RR)

Another evolutionary approach is the Artificial Bee Colony (ABC) algorithm by Karaboga et al. [8]. This algorithm is inspired by the foraging behaviour of honey bees when seeking a quality food source. In the ABC algorithm, there is a population of food positions and the artificial bees modify these food positions along time. The algorithm uses a set of computational agents called honey bees to find the optimal solution. The honey bees in ABC can be categorized into three groups: employed bees, onlooker bees, and scout bees. The employed bees exploit the food positions, while the onlooker bees are waiting for information from the employed bees about nectar amount of the food positions. The onlooker bees select food positions using the employed bee information and they exploit the selected food positions.

Finally, the scout bees find new random food positions. Each solution, in the search space, consists of a set of optimization parameters which represent a food source position. The number of employed bees is equal to the number of food sources. The quality of food source is called its “fitness value” and it is associated with its position.

In the algorithm, the employed bees will be responsible for investigating their food sources (using fitness values) and sharing the information to recruit the onlooker bees. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population (SN). Each solution (food source) $x_i (i = 1, 2, \ldots, SN)$ is a D-dimensional vector. The onlooker bees will make a decision to choose a food source based on this information. A food source with a higher quality will have a larger probability of being selected by onlooker bees. This process of a bee swarm seeking, advertising, and eventually selecting the best known food source is the process used to find the optimal solution. An onlooker bee chooses a food source depending on the probability value associated with that food source p_i calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{q=1}^{SN} fit_q}, \tag{34}$$

where fit_i is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees (BN). In this way, the employed bees exchange their information with the onlookers. In order to produce a candidate food position from the old one, the ABC uses the following expression:

$$x^*_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{lj}), \tag{35}$$

where x^*_{ij} is the new feasible food source, which is selected by comparing the previous food source x_{ij} and the randomly selected food source, $l \in$ {1,2,. . . , SN} and $j \in$ {1,2,. . . ,D} are randomly chosen indexes. ϕ_{ij} is a random number between [-1,1] which is used to adjust the old food source to become the new food source in the next iteration. We have modified (35) to fit with our specific domain of interest as follows:

$$\begin{cases} x_k^{t+1} = x_k^t + \phi(x_k^t - x_z), \\ y_k^{t+1} = y_k^t + \phi(y_k^t - y_z), \end{cases} \tag{36}$$

where (x_z y_z) represent the coordinates of selected target translated in terms of row and column of the matrix area. Here, (x_k^t, y_k^t) is the current position of a robot k, and the (x_k^{t+1}, y_k^{t+1}) is the new position of the recruited robot. In order to modify the ABC to a discrete version, like the FA and PSO, the robot movements have been limited to three possible value updates for each coordinates: {-1, 0, 1} according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [\phi(x_k^t - x_z) > 0], \\ x_k^{t+1} = x_k^t - 1 & \text{if } [\phi(x_k^t - x_z) < 0], \\ x_k^{t+1} = x_k^t & \text{if } [\phi(x_k^t - x_z) = 0], \end{cases} \tag{37}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [\phi(y_k^t - y_z) > 0], \\ y_k^{t+1} = y_k^t - 1 & \text{if } [\phi(y_k^t - y_z) < 0], \\ y_k^{t+1} = y_k^t & \text{if } [\phi(y_k^t - y_z) = 0]. \end{cases} \tag{38}$$

Essentially, we have two cases. The first is when a robot receives only one recruitment request and in this case, it will move towards the target location according to (37)–(38). If a robot receives more than one request, it needs to decide which target it will move to. In this case, we uses a concept according to the Distributed Bee Algorithm presented in [23]. Basically, when a robot k in the cell c_k^t receives a packet from a coordinator in the cell c_z^t, the cost of the target z for the robot k at step t is calculated as the Euclidean distance between the robot and the target in the 2D area:

$$r_{kz} = \sqrt{(x_k^t - x_z)^2 + (y_k^t - y_z)^2}, \quad \forall z \in RR_k \tag{39}$$

We first define the utility of a target z for the robot k the reciprocal value of the distance as:

$$\mu_z^k = \frac{1}{r_{kz}}. \tag{40}$$

Then, a probability that the robot k chooses the target z can be calculated by

$$p_z^k = \frac{\mu_z^k}{\sum_{b=1}^{RR_k} \mu_b^k}, \tag{41}$$

where $RR_k \subset F \subset T$. From (41), it is easy to show that

$$\sum_{z=1}^{RR_k} p_z^k = 1 \tag{42}$$

The underlying decision-making mechanism adopts the roulette rule, also known as the wheel-selection rule. That is, each target has been associated with a probability which it is chosen from a set of detected targets. Once all the probabilities are calculated according to (41), the robot will choose the target by spinning the wheel. Next the robot will move according to Eqs. (37)–(38). Such a coordination technique is well-suited, like the FA, to avoid that several robots approach the same target and spreading the robots over different target’s locations (Fig. 8). In the described problem, the algorithm for the bees based strategy is shown in Algorithm 4.

Like FTS-RR and PSO-RR, these steps are executed when the robots are recruited by others. In case when no targets are detected or all the tasks about the targets are performed, the robots continue to explore the area until the mission ends. Moreover, the energy model comprises the mobility cost for reaching the target's locations, communication cost for the transmission and reception of the packets to communicate the position of the found targets, and the cost related to the processing of a target (3–6).

It is worth pointing out that for all strategies, the decision mechanism is done at each step; this implies that if a recruited robot at step t chooses a target z, at the step $t+1$ takes again the decision and it could then choose another better target.

Algorithm 4 ABC-RR strategy

```
begin
    Step 1: Initialization. Set t {t is the step counter};
            set the detected targets z ∈ F, the wireless
            range R_t, and the robots in wireless range of
            the detected targets k ∈ S. Define
            randomization parameter φ
    Step 2: Generation coordination system. For
            the detected targets and the recruited
            robots, set the initial locations in terms of
            coordinates in x and y direction.
    Step 3 Procedure.
    while The stop criteria are not satisfied do
        foreach robot k in Recruited State (k ∈ S) do
            set RR_k
            evaluate the current position c_k^t
            foreach target z ∈ RR_k do
                evaluate p_z^k according to (41)
                choose the best target z according to the
                wheel-selection rule
            end foreach
            evaluate N(c_k^t)
            compute the cell c_k^{t+1} according to
            (37)–(38)
            if (c_k^{t+1}.hasObstacle() or
             c_k^{t+1}.isOccupated())
                choose a random cell c* ∈ N(c_k^t);
                move robot k towards c*;
            else
                move robot k towards c_k^{t+1};
            end if
        end foreach
        update t;
    end while
end
```

6 Simulation experiments

6.1 Selection of parameters

At the start of the simulations, all robots are in the explorative state. Robots and targets are initially deployed in the operative area according to a uniform distribution. At each step of the simulation, a robot will consume an amount of energy varying its state and the robots employ different actions in different states (Fig. 4). For example, a robot will consume more energy when performing a target than when wandering in the search area. The cell is a square with each side being one unit length. A robot consumes 1 unit of energy for traveling from one cell to another. One stop takes an extra energy of 0.5 unit. A turn of 45° takes 0.4 unit of energy. Turn of 90°, 135°, 180°, takes 0.6, 0.8, and 1 unit of energy, respectively. These numbers are approximately derived from energy measurements for Pioneer 3-DX robot [37]. We estimate the energy for performing a planned task for removing or dismantling the target is 5 units of energy for each robot involved in the task. For the exploration task, the system parameters used in the experiments are shown in Table 1 according to our previous studies [11]. Regarding the wireless communication, the value of the parameters are modelled empirically according to the previous study presented in [33] and shown in Table 2.

In our model, e_{cc}, e_{tx}, and e_{rc} have been recalculated to express them in terms of the unit of energy. Regarding the values of the parameters of the Firefly Algorithm, please refer to our previous paper in [36]. For PSO and ABC techniques, we have used the values of previous studies [38, 39], respectively. Table 3 shows the parameters used in the coordination strategies.

To evaluate the proposed techniques, we have implemented and built a Java-based simulator. In the simulations, we have considered the environment with different levels of complexity depending on the following factors: the dimension of grid, the size of the swarm of robots, and the number of targets to be treated, distributed in the area. It is worth

Table 1 Parameters used in the exploration algorithm

Parameters	Value
Sensing range R_s	4
$ERTU_{\%}$	0.2
$\Delta\tau_0$	2
φ	1
λ	1
η	0.9
a_1	0.5
a_2	0.5
ε	Uniform [0 1]

Table 2 Cost related to the wireless communication

Parameters	Value
Bit rate (B)	3
Energy consumed by a transceiver circuitry to transmit o receive a bit, e_{cc} (Joule)	10^{-7}
Energy consumed by a transceiver amplifier to transmit one bit data over one meter, e_{tx} (Joule)	10^{-12}
Energy to receive a bit, e_{rc} (Joule)	10^{-7}
Path loss exponent, ψ	[2,6]
Wireless range R_t	6,8,10

pointing out that the simulations were done by applying the same exploration strategy explained in Section 4, since the main focus of the work is to analyze the performance of the coordination techniques applied to the recruiting task.

6.2 Simulation experiments I: influence of the size of the swarm and the dimension of grid on the energy consumption

These experiments are designed to analyze the performance of the coordination strategies by varying the number of the robots in the area $k = \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60\}$ and the grid area with different numbers of cells in x and y dimension $\{40 \times 40, 50 \times 50, 60 \times 60\}$, keeping a constant number of targets and the number of robots needed to perform a target. We have evaluated in this case the behavior of the approaches when a few or many robots are used in the area of different sizes. We also consider that for dealing with a target, it is required that three robots work together.

The simulation results are summarized in Fig. 10 where each point is the average of running the proposed algorithms 50 times and it summarizes the cumulated total energy consumed by the robots (TESC), collected by each algorithm. Results show that, as the size of the robot increases, the average energy of the system decreases and as the size of the operative grid increases the energy consumed increases. It is reasonable to expect that by increasing the number of robots, the efficiency of the swarm improves in terms of energy. Regarding the three strategies, the results of Fig 10a show that the performance gap is small for a grid area with 40×40 cells, but is higher with the increase of the complexity of the mission as shown in Fig. 10b and c.

This difference is greater, comparing the PSO-RR with the others. No significant difference between the FTS-RR and ABC-RR. One possible explanation is that the decision mechanisms in FTS-RR and ABC-RR take into account different criteria. PSO-RR takes into account the distance between the positions of the robots and the targets. Instead, FTS-RR considers both distance and random metrics and ATS-RR adopts the roulette rules. Therefore, both approaches, typically, allow to distribute better the robots among the targets.

Table 3 Parameters used in the coordination algorithms

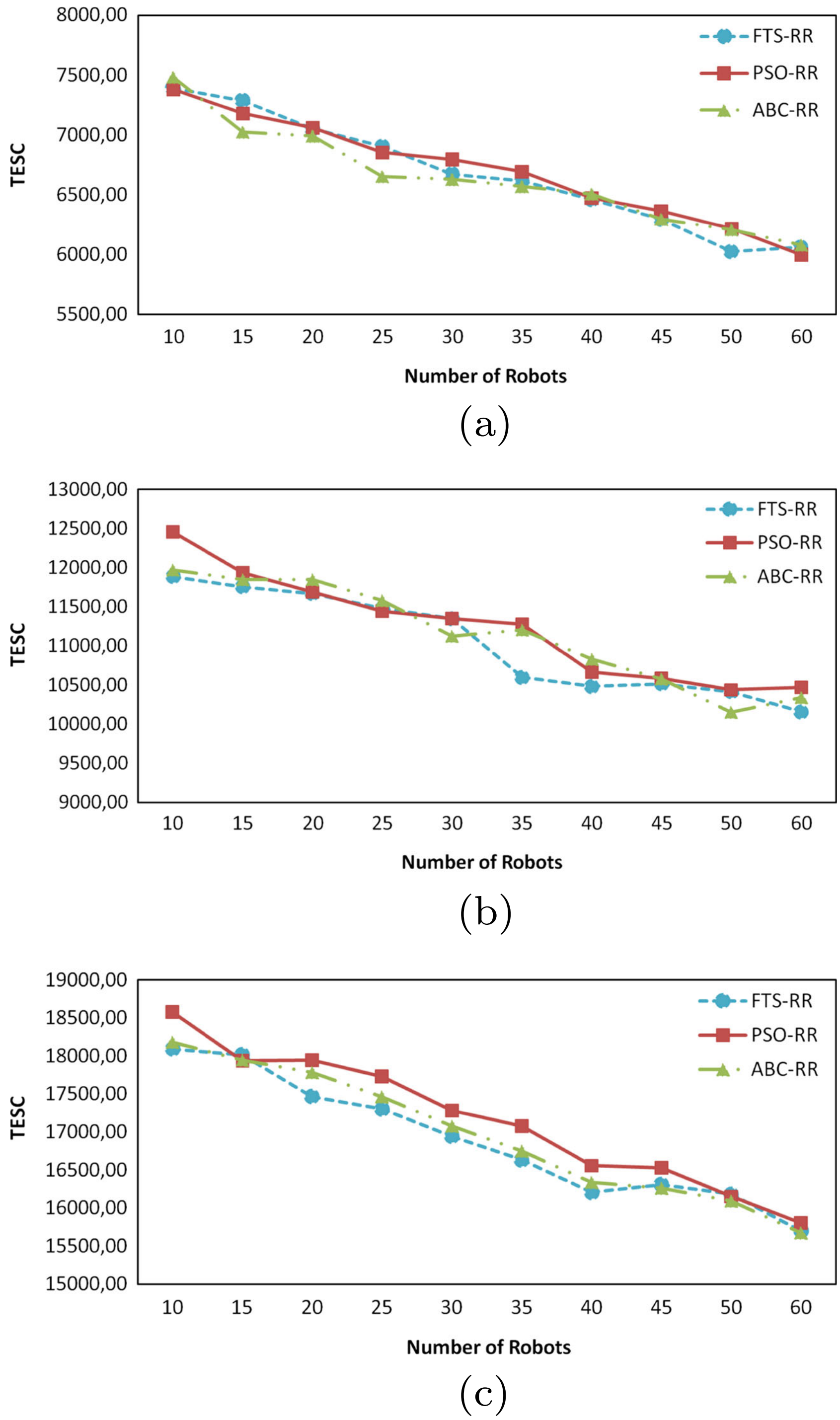
Parameters	Value
α	0.2
β_0	0.5
γ	$\frac{1}{L}$ ($L = \max\{m,n\}$)
σ	Uniform [0,1]
ω	0.729
r_1	Uniform [0,1]
c_1	2
ϕ	Uniform [-1,1]

6.3 Simulation experiments II. Influence of the number of targets on the energy consumption

Now, we evaluate the energy consumed by the system applying the strategies, when a few or many targets exist, varying the terrain size and the number of involved robots. We considered $z = \{3, 5, 7, 10\}$, the dimension of the swarm of robots $k = \{20, 30, 40\}$ and the grid area with different number of cells in x and y dimension $\{40 \times 40, 50 \times 50, 60 \times 60\}$. Some interesting features can be observed from Fig. 11. The ABC-RR and FTS-RR techniques perform better and help to allocate reasonable robots to different targets saving the energy, especially when the number of robots is small. However, a larger robot team obtains more benefit and there is no significant difference between the three strategies.

However, a team with a larger number of robots generally increase the performance, saving the total consumed energy. Obviously, the more targets are introduced, the more energy is consumed. Nevertheless, increasing the number of targets, the recruiting tasks become more complex and the used strategy becomes more important. The difference of the three strategies in terms of energy consumption is high, especially when the size of swarm in the operative area is low and it is comparable when the number of robots increases at the same condition of the size of the area. When the complexity of the task increases, it can be seen from Fig. 11b and c that it is possible that more robots in an overlapped region receive the same requests and go towards the same targets, creating unnecessary redundancy. However, in most scenarios, FTS-RR exhibits superior performance and distributes the robots better in the area, especially in comparison with the PSO-RR. Regarding the difference between the FTS-RR and ABC-RR, the measure of the total energy is comparable and not significant difference when the task is not complex and number of robots to in the area is high. But increasing the number of targets and using a small team (e.g., 20), the FTS-RR exhibits superior performance in

Fig. 10 Evaluation of Total-Energy-System-Consumed (TESC) in terms of units of energy, for performing three targets varying the number of robots involved in the mission considering three robots needed to deal with a single target **a** 40 × 40 grid, **b** 50 × 50 grid, and **c** 60 × 60 grid



terms of energy consumed. This implies that the FTS-RR would be more promising for solving recruitment tasks in complex scenarios.

6.4 Simulation experiments III. Influence of the wireless range on the energy consumption

The last experiment in this paper is designed to evaluate the influence of the wireless range on the energy consumption by varying different ranges $R_t \in \{6, 8, 10\}$. Here we considered a grid area 50x50, $z = (7, 10, 15)$ and 3 robots needed to treat a target. It is important to point out that effective communication between the robots is highly dependent on the parameters of the problem such as the size of the swarm of robots, and the number of disseminated targets in the area.

The results are summarized in Fig. 12 where some interesting features can be observed. A robot team with a small number of robots (e.g., 20) is mainly affected by the positive

Fig. 11 Evaluation of Total-Energy-System-Consumed (TESC) in terms of units of energy, for performing 3, 5, 7, and 10 targets and 3 robots needed to perform a target **a** 40 × 40 grid, **b** 50 × 50 grid, and **c** 60 × 60 grid

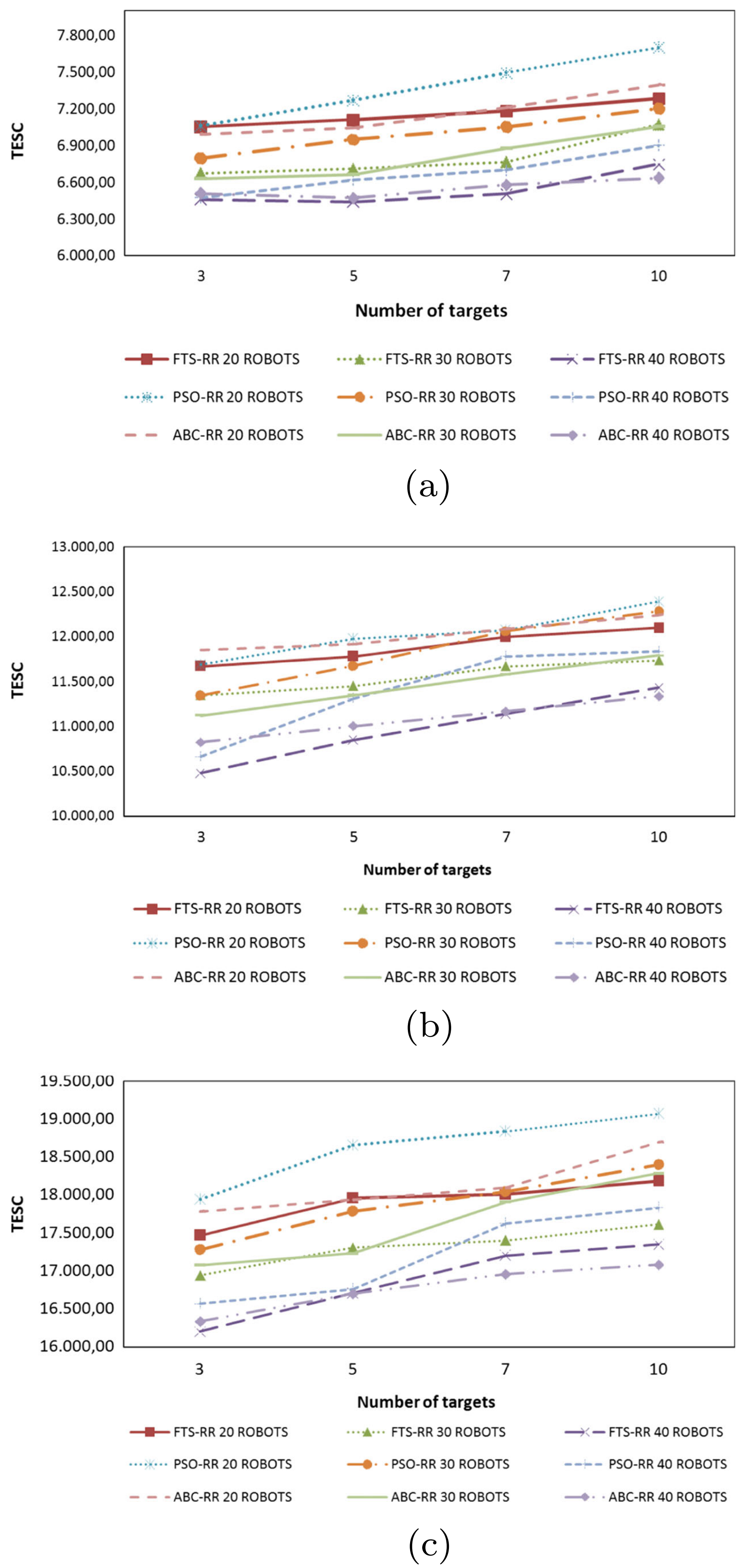
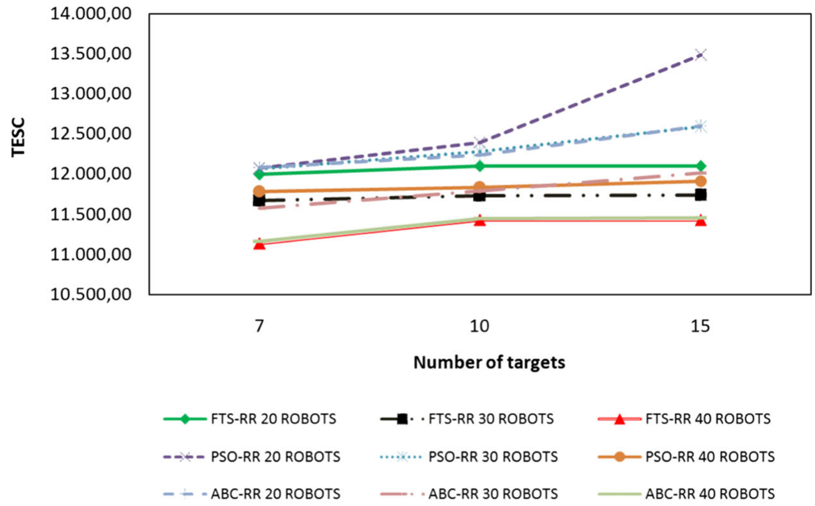
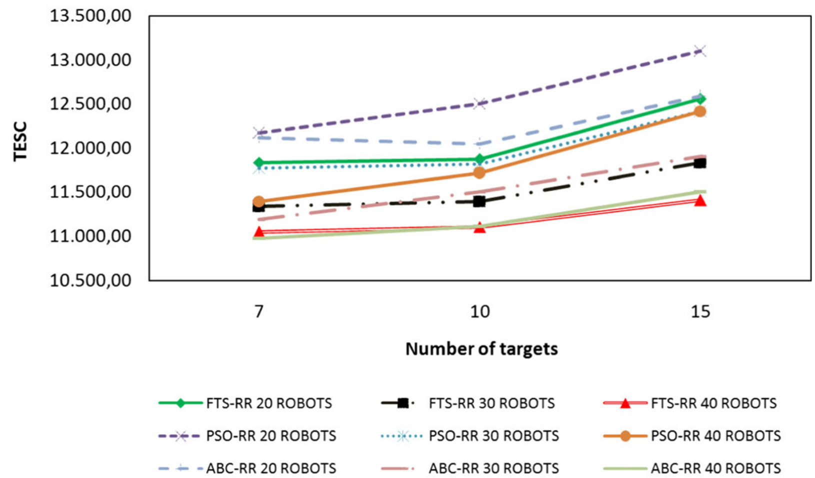


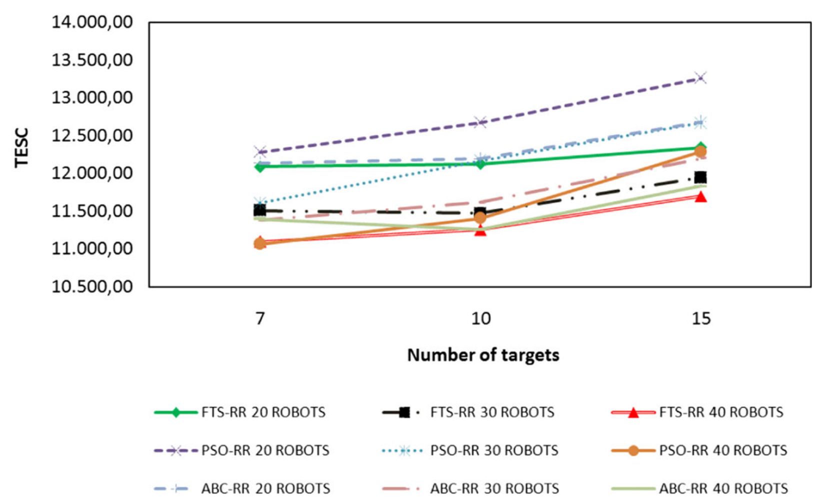
Fig. 12 Evaluation of Total-Energy-System-Consumed (TESC) in terms of units of energy, for performing 7, 10, and 15 targets and 3 robots needed to perform a target in 50×50 grid **a** $R_t = 6$, **b** $R_t = 8$, and **c** $R_t = 10$



(a)



(b)



(c)

Table 4 Results of *p* value test for FTS-RR, PSO-RR, and ABC-RR

	FTS-RR vs PSO-RR			FTS-RR vs ABC-RRl			PSO-RR vs ABC-RR		
	Fig. 10a	Fig. 10b	Fig. 10c	Fig. 10a	Fig. 10b	Fig. 10c	Fig. 10a	Fig. 10b	Fig. 10c
p value	0.1961	0.0156	0.0012	0.2421	0.0879	0.0544	0.0584	0.1218	0.0028

Table 5 Results of *p* value in *t* test for FTS-RR and PSO-RR

	FTS-RR vs PSO-RR								
	20 Robots varying the number of targets Fig. 11a	30 Robots varying the number of targets Fig. 11a	40 Robots varying the number of targets Fig. 11a	20 Robots varying the number of targets Fig. 11b	30 Robots varying the number of targets Fig. 11b	40 Robots varying the number of targets Fig. 11b	20 Robots varying the number of targets Fig. 11c	30 Robots varying the number of targets Fig. 11c	40 Robots varying the number of targets Fig. 11c
p value	0.0412	0.0158	0.0221	0.0489	0.0455	0.0103	0.0267	0.0405	0.0277

Table 6 Results of *p* value in *t* test for FTS-RR and ABC-RR

	FTS-RR vs ABC-RR								
	20 Robots varying the number of targets Fig. 11a	30 Robots varying the number of targets Fig. 11a	40 Robots varying the number of targets Fig. 11a	20 Robots varying the number of targets Fig. 11b	30 Robots varying the number of targets Fig. 11b	40 Robots varying the number of targets Fig. 11b	20 Robots varying the number of targets Fig. 11c	30 Robots varying the number of targets Fig. 11c	40 Robots varying the number of targets Fig. 11c
p value	0.4812	0.4921	0.4189	0.0412	0.1005	0.1675	0.0798	0.0837	0.1933

Table 7 Results of *p* value in *t* test for PSO-RR and ABC-RR

	PSO-RR vs ABC-RR								
	20 Robots varying the number of targets Fig. 11a	30 Robots varying the number of targets Fig. 11a	40 Robots varying the number of targets Fig. 11a	20 Robots varying the number of targets Fig. 11b	30 Robots varying the number of targets Fig. 11b	40 Robots varying the number of targets Fig. 11b	20 Robots varying the number of targets Fig. 11c	30 Robots varying the number of targets Fig. 11c	40 Robots varying the number of targets Fig. 11c
p value	0.0247	0.0459	0.0663	0.4469	0.0445	0.0889	0.0192	0.0451	0.0419

Table 8 Results of *p* value in *t* test for FTS-RR and PSO-RR related to the wireless range

	FTS-RR vs PSO-RR								
	20 Robots varying the number of targets Fig. 12a	30 Robots varying the number of targets Fig. 12a	40 Robots varying the number of targets Fig. 12a	20 Robots varying the number of targets Fig. 12b	30 Robots varying the number of targets Fig. 12b	40 Robots varying the number of targets Fig. 12b	20 Robots varying the number of targets Fig. 12c	30 Robots varying the number of targets Fig. 12c	40 Robots varying the number of targets Fig. 12c
p value	0.1426	0.0469	0.0186	0.0276	0.0112	0.0413	0.0600	0.0651	0.1633

Table 9 Results of p value in t test for FTS-RR and ABC-RR elated to the wireless range

	FTS-RR vs ABC-RR								
	20 Robots varying the number of targets Fig. 12a	30 Robots varying the number of targets Fig. 12a	40 Robots varying the number of targets Fig. 12a	20 Robots varying the number of targets Fig. 12b	30 Robots varying the number of targets Fig. 12b	40 Robots varying the number of targets Fig. 12b	20 Robots varying the number of targets Fig. 12c	30 Robots varying the number of targets Fig. 12c	40 Robots varying the number of targets Fig. 12c
p value	0.0978	0.2625	0.0317	0.0795	0.4542	0.4321	0.1237	0.2523	0.1142

side of a high communication range, although a relatively shorter communication range means lower power consumption. The reason in that over long communication range, more robots can be recruited and they can be allocated to different targets in a shorter time. However, the results also show that, when the communication range is increased, the performance improves up to a certain point beyond which there is no change in the performance of the system and in such case the increasing of the total energy consumed. A scenario with a huge amount of robots (e.g., 40) implies a huge amount of consumed energy since the recruitment task involves multiple robots, usually unnecessary, with some consequent waste of energy. For example, Fig. 12a highlights lower consumption of energy for a larger number of robots using a short communication range than the use of the longer communication ranges (Fig. 12b–c). Regarding the three strategies, both FTS-RR and ABC-RR perform better than the PSO-RR, especially in a small robot team (e.g., 20 robots) and many targets disseminated in the area (e.g., 15). Concerning the difference between the FTS-RR and ABC-RR, FTS-RR outperforms the other mainly in complex scenarios and thus allows to spread the robots in a better way over the environment, avoiding the situation that several robots approach the same target and thus saving the energy.

6.5 Statistical tests

To validate the quality of solutions and performance of the three meta-heuristic techniques, we have also considered the p values of Student t tests. The t tests were used to analyze the relationships between the results obtained from the three meta-heuristics. The parameter of interest is the p value. Tables 4, 5, 6, 7, 8, 9, and 10 show the p value obtained from the t tests using all above simulation results for all considered scenario. If $p < 0.05$, there is a statistical evidence of the difference between the strategies.

The statistical tests confirm that ABC-RR and FTS-RR perform better than the PSO-RR when the tasks to be completed is complex in terms of the terrain size and the number of targets in the area. Regarding the difference between the FTS-RR and ABC-RR, we can say that the performance of the two strategies is comparable. However, increasing the complexity of the tasks in terms of the size of area and the number of targets using a small robots team, the FRS-RR will be better with the slightly reduced energy consumption.

7 Conclusion and future work

We have developed and tested three biologically inspired coordination strategies for robot swarm coordination under complex constraints. These techniques have been based on the firefly, particle swarm and artificial bee behaviour, and some discrete modifications have been carried out to make these algorithms suitable for the purpose. We have also formulated the problem as an optimization problem with mathematical models for energy consumptions. The main objective has been formulated to minimize the overall energy consumption for the exploration and recruitment tasks. The energy consumed by the system is a measure of

Table 10 Results of p value in t test for PSO-RR and ABC-RR elated to the wireless range

	PSO-RR vs ABC-RR								
	20 Robots varying the number of targets Fig. 12a	30 Robots varying the number of targets Fig. 12a	40 Robots varying the number of targets Fig. 12a	20 Robots varying the number of targets Fig. 12b	30 Robots varying the number of targets Fig. 12b	40 Robots varying the number of targets Fig. 12b	20 Robots varying the number of targets Fig. 12c	30 Robots varying the number of targets Fig. 12c	40 Robots varying the number of targets Fig. 12c
pvalue	0.1781	0.0031	0.0181	0.0712	0.0288	0.0471	0.0469	0.0518	0.3647

how efficient the recruiting strategy is. The most important features of the proposed approach are as follows:

– Flexibility: parameters can be easily tuned so that the proposed methodology can used to carry out exploration and recruitment tasks for a system of mobile robots.
– Scalability: the algorithm works well for any number of robots and targets.
– Adaptability: the approach can be used in the environment, allowing different conditions and distributions of targets and robots.
– Parallelism: the algorithm is distributed and each robot performs its task in parallel and make decision individually, based on local partial information.

Our experiments through simulation have showed that the energy consumption is higher for the Particle Swarm approach, especially when the size of swarm is low and the dimension of area and the the number of targets are high. The FTS-RR and ATS-RR methods are comparable when the task is not complex, but the difference is more evident when the number of targets to be performed increases and the number of robots in the area is small. Therefore, the coordination mechanism becomes more complex for complex tasks, and the firefly-based strategy usually gives better performance.

The work and approaches presented in this paper have paved a way for exploring new bio-inspired techniques for optimizing complex tasks for swarming robots. Future work will focus on the extension of the current approaches to multi-objective optimization by considering multiple objectives such as the minimization of the exploration and handling time, the maximization of the exploration coverage, and the minimization of the computational costs. Extension will also explore the possibility of more complex, 2D geometrical areas with multiple obstacles or barriers and even 3D terrains with inaccessible regions such as rivers and lakes. Other bio-inspired approaches will also be investigated. It can be expected that it will inspire more active research in this exciting area of swarming robots with potentially more realistic real-world applications.

Compliance with ethical standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Parker LE (2008) Distributed intelligence: overview of the field and its application in multi-robot system. J Phys Agents 2(1):5–14
2. Mei Y, Lu YH, Hu YC, Lee CSG (2005) A case study of mobile robot's energy consumption and conservation techniques. In: Proceedings of the 12th international conference on advanced robotics (ICAR '05). Piscataway, pp 492–497
3. Liu W, Winfield AF, Sa J, Chen J, Dou L (2007) Towards energy optimization: emergent task allocation in a swarm of foraging robots. Adapt Behav 15(3):289–305
4. Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
5. De Rango F, Palmieri N, Yang XS, Marano S (2015) Bio-inspired exploring and recruiting tasks in a team of distributed robots over mined regions. In: International symposium on performance evaluation of computer and telecommunication system (SPECTS 2015). Chicago, pp 1–8
6. Yang XS (2009) Watanabe O, Zeugmann T (eds). Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T (eds) Proceedings of 5th symposium on stochastic algorithms, foundations and applications, SAGA 2009, pp 169178. Springer, Heidelberg
7. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Piscataway, pp 1942–1948
8. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
9. Senanayake M, Senthoorana I, Barca J, Chung H, Kamruzzamanc J, Murshedc M (2016) Search and tracking algorithms for a swarm of robots: a survey. J Robot Auton Syst 75:422–434
10. Yan Z, Jouandeau N, Ali Cherif A (2013) A survey and analysis of multi-robot coordination. Int J Adv Robot Syst 10(399):1–18
11. De Rango F, Palmieri N (2012) A swarm-based robot team coordination protocol for mine detection and unknown space discovery. In: 8th international conference on wireless communications and mobile computing (IWCMC 2012). Limassol, pp 703–709
12. Chen X, Kong Y, Fang X, Wu Q (2013) A fast two-stage ACO algorithm for robotic path planning. J Neural Comput Applic 22(2):313–319
13. Fujisawa R, Dobata S, Kubota D, Imamura H, Matsun F (2008) Dependency by concentration of pheromone trail for multiple robots. In: Proceedings of the 6th international conference on ant colony optimization and swarm intelligence (ANTS 2008). Berlin, pp 283–290
14. Garnier S, Tache F, Combe M, Grimal A, Theraulaz G (2007) Alice in pheromone land: an experimental setup for the study ofant-like robots. In: Proceedings of the IEEE swarm intelligence symp. (SIS 2007), Washington DC, pp 37–44
15. Sugawara K, Kazama T, Watanabe T (2004) Foraging behavior of interacting robots with virtual pheromone. In: Proceedings of 2004 IEEE/RSJ international conference on intelligent robots and systems. IEEE Press, Los Alamitos, pp 3074–3079
16. Masár M (2013) A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In: Proceedings of the IEEE 17th internatinal conference on intelligent engineering systems (INES 2013). San Jose, pp 271–275
17. Ducatelle F, Di Caro GA, Pinciroli C, Gambardella LM (2011) Selforganized cooperation between robotic swarms. Swarm Intell 5(2):73–96
18. Pugh J, Martinoli A, Zhang AY (2005) Particle swarm optimization for unsupervised robotic learning. In: Proceedings of IEEE swarm intelligence symposium (SIS 2005). Piscataway, pp 92–99
19. Masár M, Zelenka J (2012) Modification of PSO algorithm for the purpose of space exploration. In: Proceedings of IEEE international conference on intelligent engineering systems (INES 2012). Piscataway, pp 223–226
20. Hereford JM, Siebold MA (2010) Bio-inspired search strategies for robot swarms. In: Swarm robotics, from biology to robotics, pp 1–27

21. Couceiro MS, Vargas PA, Rocha RP, Ferreira NMF (2014) Benchmark of swarm robotics distributed techniques in a search task. Robot Auton Syst 62(2):200–213
22. Kernbach S, Thenius R, Kernbach O, Schmickl T (2009) Reembodiment of honeybee aggregation behavior in artificial microrobotic system. Adapt Behav 17(3):237–259
23. Jevtic A, Andina GAD, Jamshidi M (2012) Distributed bees algorithm for task allocation in swarm of robots. IEEE Syst J 6(2):296–304
24. Contreras-Cruz MA, Ayala-Ramirez V, Hernandez-Belmonte UH (2015) Mobile robot path planning using artificial bee colony and evolutionary programming. Appl Soft Comput 30:319–328
25. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 22(3):52–67
26. Yang B, Ding Y, Jin Y, Hao K (2015) Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis. Robot Auton Syst 72:83–92
27. Bayindir L (2016) A review of swarm robotics tasks. Neurocomputing 172:292–321
28. Mills KL (2007) A brief survey of self-organization in wireless sensor networks. J Wireless Commun Mobile Comput 7(7):823–834
29. Gu Y, Bozdag D, Brewer R, Ekici E (2006) Data harvesting with mobile elements in wireless sensor networks. Comput Netw 50(17):3449–3465
30. Stirling T, Floreano D (2010) Energy efficient swarm deployment for search in unknown environments. In: Swarm intelligence. Springer, Berlin, pp 562–563. doi:10.1007/978-3-642-15461-4_61
31. Heo N, Varshney PK (2005) Energy-efficient deployment of intelligent mobile sensor networks. IEEE Trans Syst Man Cybern Part A Syst Humans 35(1):78–92
32. Barca J, Sekercioglu YA (2013) Swarm robotics reviewed. J Robotica 31(3):345–359
33. Ooi CC, Schindelhauer C (2009) Minimal energy path planning for wireless robots. Mobile Netw Appl 14(3):309–321
34. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2(2):78–84
35. Yang XS, Deb S, Hanne T, He X (2015) Attraction and diffusion in nature-inspired optimization algorithms. J Neural Comput Applic, 1–8
36. Palmieri N, Marano S (2016) Discrete firefly algorithm for recruiting task in a swarm of robots. In: Yang XS (ed) Nature-inspired computation in engineering. Springer, pp 133–150
37. Mei Y, Lu YH, Lee CSG, Hu YC (2006) Energy efficient mobile robot exploration. In: The proceedings of the IEEE international conference of robotics and automation (ICRA 2006). Orlando FL Ma, pp 505–551
38. Clerc C, Kennedy J (2002) The particle swarm-explosion, stability and convergence in a multidimensioanl complex space. IEEE Trans Evol Comput 6(1):58–73
39. Zhang B, Liu T, Zhang C (2016) Artificial bee colony algorithm with strategy and parameter adaption for artificial bee colony algorithm with strategy and parameter adaption for global optimization. J Neural Comput & Applic, 1–16