J Intell Robot Syst (2014) 76:353–381
DOI 10.1007/s10846-014-0030-0

Mechanism and Convergence Analysis of a Multi-Robot Swarm Approach Based on Natural Selection

Micael S. Couceiro · Fernando M. L. Martins · Rui P. Rocha · Nuno M. F. Ferreira

Received: 16 June 2013 / Accepted: 22 January 2014 / Published online: 13 February 2014

Abstract The Darwinian Particle Swarm Optimization (DPSO) is an evolutionary algorithm that extends the Particle Swarm Optimization (PSO) using natural selection, or survival-of-the-fittest, to enhance the ability to escape from local optima. An extension of the DPSO to multi-robot applications has been recently proposed and denoted as Robotic Darwinian PSO (RDPSO), benefiting from the dynamical partitioning of the whole population of robots. Therefore, the RDPSO decreases the amount of required information exchange among robots, and is scalable to large populations of robots. This paper presents a stability analysis of the RDPSO to better understand the relationship between the algorithm parameters and the robot's convergence. Moreover, the analysis of the RDPSO is further extended for real robot constraints (e.g., robot dynamics, obstacles and communication constraints) and experimental assessment with physical robots. The optimal parameters are evaluated in groups of physical robots and a larger population of simulated mobile robots for different target distributions within larger scenarios. Experimental results show that robots are able to converge regardless of the RDPSO parameters within the defined attraction domain. However, a more conservative parametrization presents a significant influence on the convergence time. To further evaluate the herein proposed approach, the RDPSO is further compared with four state-of-the-art swarm robotic alternatives under simulation. It is observed that the RDPSO algorithm provably converges to the optimal solution faster and more accurately than the other approaches.

Keywords Swarm robotics · Natural selection · Convergence analysis · Robot constraints · Parameterization · Source localization

Mathematics Subject Classifications (2010) 39A30 · 70E60 · 65L20

M. S. Couceiro (✉) · R. P. Rocha
Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal
e-mail: micaelcouceiro@isr.uc.pt

R. P. Rocha
e-mail: rprocha@isr.uc.pt

F. M. L. Martins
Instituto de Telecomunicações (Covilhã), RoboCorp, Coimbra College of Education, Rua Dom João III - Solum, 3030-329 Coimbra, Portugal
e-mail: fmlmartins@esec.pt

N. M. F. Ferreira
RoboCorp, Electrical Engineering Department, Engineering Institute of Coimbra, Rua Pedro Nunes - Quinta da Nora, 3030-199 Coimbra, Portugal
e-mail: nunomig@isec.pt

1 Introduction

Bio-inspired algorithms have been employed in situations where conventional optimization techniques

cannot find a satisfactory solution, for instance, when the optimization function to be optimized is discontinuous, non-differentiable, and/or presents too many nonlinearly related parameters [1]. One of the most well-known bio-inspired algorithms is the Particle Swarm Optimization (PSO) which basically consists of a machine-learning technique inspired by birds flocking in search of food [2]. However, a general problem with the PSO and other optimization algorithms is that they can be trapped in sub-optimal solutions, which may work well in some problems but fail in others [1]. In search of a better model of natural selection inspired by the PSO algorithm, the Darwinian Particle Swarm Optimization (DPSO) was formulated by Tillett et al. [3]. In this algorithm, multiple swarms of test solutions, each one of them performing just like an ordinary PSO, may exist at any time with some rules governing the collection of swarms which are designed to simulate natural selection.

Using the findings inherent to bio-inspired algorithms on multi-robot applications may contribute to the solution of several real-world problems such as search and rescue [4]. Nevertheless, it should be noted that, contrary to the herein proposed multi-robot swarm approach, denoted as Robotic Darwinian PSO (RDPSO), most works only consider PSO and its main variants applied to computational optimization problems, wherein particles are not endowed with physical constraints inherent to mobile platforms [5]. Unlike particles, i.e. virtual agents, robots are designed to act in the real world where their dynamical characteristics and environmental obstacles need to be taken into account. Moreover, when the communication infrastructure is damaged or missing (e.g., hostile environments, search and rescue, disaster recovery, battlefields, space and others), the use of mobile ad-hoc networks (MANETs) and multi-hop communication, wherein robots also act as routers to support the collective cooperation and coordination is necessary. In other words, robots need to be able to create and maintain a MANET, wherein each one of them is both a communication and a relay node, thus presenting itself as a suitable solution to most real robotic applications. Nevertheless, such strategy requires a distributed self-spreading over a geographical area and a constrained collective control of mobile nodes in order to ensure the network connectedness.

Bearing this idea in mind, this paper contribution can be divided into three research aspects:

i) Review of the RDPSO in which its main mechanisms that were initially introduced in [6] and [7] are now highlighted and further explored , thus formally presenting the distributed RDPSO algorithm for the first time.
ii) Formulation of a stability analysis of the RDPSO so as to obtain an attraction domain relating its several parameters, while taking into account real robot constraints (e.g., dynamical characteristics, exploration capabilities, obstacle avoidance and MANET connectivity). This problem was previously addressed in [8] and is now further extended, thoroughly describing the stability analysis and evaluating the influence of each parameter within the defined attraction domain using two physical robots.
iii) Evaluation of the RDPSO on groups of real and simulated robots under a source localization problem with different target distribution based on commonly used benchmark functions. A Multivariate Analysis of Variance (MANOVA) is carried out to evaluate the performance of the algorithm based on the number of robots, set of parameters and target distribution. Furthermore, the parameterized RDPSO is compared with four state-of-the-art swarm robotic alternatives under a simulated mapping task.

This paper is organized as follows. Section 3 revises the RDPSO and all its main features, presenting the distributed algorithm for the first time. Section 4 studies the stability of the RDPSO and the influence of its parameters on robots' convergence considering real world constraints. Section 5 thoroughly evaluates the RDPSO on both real and simulated experiments by statistically comparing its performance under different parameter configuration and with different robotic swarm strategies. Finally, the take-home message and the main conclusions are outlined in Sections 6 and 7, respectively.

2 Related Work

Regardless of PSO main variants [9, 10], the difficulties in setting and adjusting the parameters, as well

as in maintaining and improving the search capabilities for higher dimensional or constrained problems, are still a matter that has been addressed in recent works (e.g., [11, 12]). One of the most common methods to deal with such issue presented in the literature to deal with this issue is based on the stability analysis of the algorithm. For instance, Clerc and Kennedy [11] analyzed the individual particles' trajectories leading to a generalized model of the algorithm, which contains a set of coefficients to control the system's convergence tendencies. The resulting system was a second-order linear dynamical system whose stability and parameters depended on the system poles or the eigenvalues of the state matrix. Alternatively, Kadirkamanathan et al. [12] proposed a stability analysis of a stochastic particle dynamics by representing it as a nonlinear feedback controlled system. The Lyapunov stability method was applied to the particle dynamics to determine the sufficient and conservative conditions for asymptotic stability. However, the analysis provided by the authors has addressed only the issue of absolute stability, thus ignoring the optimization convergence to the optimal solution. More recently, a theoretical and empirical analysis of a PSO variation was presented in [13]. Similarly to the herein proposed work, the authors presented the convergence analysis of the traditional PSO, thus providing a significant insight on the stochastic dynamic of the algorithm. The comparison with genetic algorithms (GA) showed that a good parameter control ensures a better performance of the PSO. However, the authors concluded that the introduction of evolutionary operators, such as natural selection, inevitably improves the algorithm convergence significantly.

Many other works have benefitted from PSO-based algorithms applied to Multi-Robot Systems (MRS) for search applications. However, still none of them has presented a formal convergence analysis to find the best parameters while considering MRS characteristics. As an example, the work of Pugh and Martinoli [14, 15] introduced an adapted version of the PSO to distributed unsupervised robotic learning in groups of robots. The authors analyzed how the performance was affected if the standard PSO neighborhood structure was adapted to a more closely model, which can be found in real robot groups with limited communication abilities. Experimental results showed that the adapted version of the PSO maintained a better performance for groups of robots of various sizes when compared to other bio-inspired methods. However, contrary to the RDPSO algorithm, all bio-inspired methods used, including the adapted PSO, tend to get trapped in local solutions. Also, the authors only evaluated the adapted PSO using a set of parameters without any justification of their choice. More recently, the work of Prasanna and Saikishan [16] involved the path-planning and coordination of multiple robots in a static-obstacle environment based on the PSO and Bacteria Foraging Algorithm (BFA). As this work uses natural selection to avoid getting trapped in sub-optimal solutions, the one proposed by the authors enhances the local search using the BFA. Experimental results were conducted in a simulation environment developed in Visual Studio where the pose and shape of obstacles were previously known. However, only one target and two robots were used, thus limiting the evaluation of the proposed algorithm. Moreover, the authors did not mention the influence of the parameters on the algorithm performance. Similarly, Hereford and Siebold [17] proposed an embedded version of the PSO to swarm platforms. As in the RDPSO, there are no central agents to coordinate robots' movements or actions. Despite the potentialities of the physically-embedded PSO, experimental results were carried out using a population of only three robots, performing a distributed search in a scenario without sub-optimal solutions. Furthermore, collision avoidance and fulfillment of MANET connectivity were not considered. The authors found the inertial coefficient using a trial-and-error methodology but did not present any results about the choice regarding the other PSO parameters. The work of Jatmiko et al. [18] presented modified form of the PSO to control robots behavior for odor source localization, thus studying how they respond to turbulence and wind changes. Once again, the authors did not go to any lengths to explain their choice of the PSO parameters.

Next section presents the RDPSO main features which will be further explored in the subsequent sections, thus obtaining the relation between the algoritm parameters.

3 Robotic Darwinian PSO

This section presents an overview of the RDPSO algorithm proposed in [6] and further extended in [7]. Since the RDPSO approach is a complete adaptation

of the DPSO [3] for real mobile robots, five general features were proposed: i) an improved inertial influence based on fractional calculus concept, taking into account convergence dynamics; ii) an obstacle avoidance behavior to avoid collisions; iii) an algorithm to ensure that the MANET remains connected throughout the mission; iv) a novel methodology to establish the initial planar deployment of robots preserving the connectivity of the MANET while spreading out the robots as much as possible; and v) a novel "punish"-"reward" mechanism to emulate the rejetion and addition of robots based on social exclusion and inclusion phenomena.

The behavior of robot n can then be described by the following discrete equations at each discrete time, or iteration, $t \in \mathbb{N}_0$:

$$v_n[t+1] = w_n[t] + \sum_{i=1}^{4} \rho_i r_i \left(\chi_i[t] - x_n[t]\right), \tag{1}$$

$$x_n[t+1] = x_n[t] + v_n[t+1], \tag{2}$$

where coefficients $\rho_i, i = 1, 2, 3, 4$, assign weights to the inertial influence, the local best (cognitive component), the local best (social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity, with $\rho_i > 0$. Note that the size of all vectors (ϖ) depends on the dimensionality $\mathbb{R}^{\varpi}$ of the physical space being explored, e.g., $\varpi = 2$ for planar problems. $v_n[t]$ and $x_n[t]$ represent the velocity and position vector of robot n, respectively. In the common PSO algorithm, the inertial component $w_n[t]$ is usually proportional to the inertial influence. The RDPSO uses fractional calculus (FC) [19, 20], to describe the dynamic phenomenon of a robot's trajectory. Although next section briefly presents this feature, a more detailed description can be found in [21]. $\chi_i[t]$ represents the best position for the cognitive, social, obstacle and MANET matrix components. The cognitive $\chi_1[t]$ and social $\chi_2[t]$ components are the commonly presented in the classical *PSO* algorithm. $\chi_1[t]$ represents the local best position of robot n while $\chi_2[t]$ represents the global best position of robot n. Since the other features $\chi_3[t]$ and $\chi_4[t]$ are novel, they are briefly explored in the subsequent sections. In brief, $\chi_3[t]$ represents the local best position of robot n regarding the sensed obstacles so far. Similarly, $\chi_4[t]$ represents the local best position of robot n that allows maintaining a connected MANET based on its closest neighbor. Note that the size of all vectors (ϖ) depends on the dimensionality $\mathbb{R}^{\varpi}$ of the physical space being explored, e.g., $\varpi = 2$ for planar problems.

Considering Eqs. (1) and (2), it is noteworthy that robots will tend to converge to the optimal solution. However, although all robots within a swarm agree with the best solution, they must also fulfill other requirements (i.e., avoid obstacles and maintain a certain distance between neighbors).

3.1 Fractional Order Convergence

Fractional calculus (FC) has attracted the attention of many researchers due to its application in various scientific fields such as engineering, computational mathematics, fluid mechanics, among others [19, 20]. FC can be considered as a generalization of integer-order calculus, thus accomplishing what integer-order calculus cannot. As a natural extension of the integer (i.e., classical) derivatives, fractional derivatives provide an excellent tool for the description of memory and hereditary properties of processes. One of the most common approaches presented in the literature (cf., [22]) is the approximate discrete time Grnwald-Letnikov fractional difference of order α, $0 < \alpha \leq 1$, given by the following equation:

$$D^{\alpha}[v_n[t+1]] = \frac{1}{T^{\alpha}} \sum_{k=0}^{r} \frac{(-1)^k \Gamma(\alpha+1) v_n[t+1-kT]}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{3}$$

where T is the sampling period, Γ the gamma function and r is the truncation order of the fractional difference of the discrete velocity $v_n[t+1]$. Based on Eqs. (1) and (3), and considering $T = 1$, $r = 4$, the inertial component of robot n can be defined as:

$$\begin{aligned} w_n[t] = {} & \alpha v_n[t] + \frac{1}{2}\alpha v_n[t-1] \\ & + \frac{1}{6}\alpha(1-\alpha) v_n[t-2] \\ & + \frac{1}{24}\alpha(1-\alpha)(2-\alpha) v_n[t-3]. \end{aligned} \tag{4}$$

The truncation r of Eq. (3) depends on the requirements of the application and the features of the robot. For instance, for the eSwarBot (Educative Swarm Robot) platforms previously presented in [23], a $r = 4$

leads to results of the same type as for $r > 4$. Although one could consider the processing power to be the main reason to use a limited number of terms, the kinematical features of the platform and the mission requirements also need to be considered. Please refer to [21] for a more detailed description of the fractional calculus extension of the RDPSO algorithm and the memory complexity inherent to it. The features revealed by fractional calculus make this mathematical tool well suited to describe phenomena such as irreversibility and chaos because of its inherent memory property. In this line of thought, the dynamic phenomenon of a robot's trajectory configures a case wherein fractional calculus tools fit adequately.

3.2 Obstacle Avoidance

When a robot must move from any arbitrary start position to any target position in the environment, it must be able to avoid both static and dynamic obstacles [24]. Therefore, a new cost or fitness function is defined in such a way that it would guide the robot to perform the main mission while avoiding obstacles. For this purpose, it is assumed that each robot is equipped with sensors capable of scanning the environment for obstacle detection within a finite sensing radius r_s. A monotonic and positive sensing function $g(x_n[t])$ depending on the sensing information (i.e., distance from the robot to obstacle) is defined. In most situations $g(x_n[t])$ can be represented as the relation between the analog output voltage of range sensors and the distance to the detected object. $\chi_3[t]$ is then represented by the position of each robot that optimizes the monotonically decreasing or increasing $g(x_n[t])$. In an obstacle-free environment, the obstacle susceptibility weight ρ_3 is zero. However, in real-world scenarios, obstacles need to be taken into account and the value of ρ_3 depends on several conditions related to the main objective (i.e., minimizing a cost function or maximizing a fitness function) and the sensing information (i.e., monotonicity of $g(x_n[t])$). Furthermore, the relation between ρ_3 and the other weights depends on the susceptibility of each robot to obstacle avoidance behavior and the sensing radius r_s in which a robot is expected detect obstacles.

Recently, different strategies have been presented in the literature to handle collision avoidance by using low level control routines triggered whenever robots sense obstacles (e.g., [25]). Although such methodology allows for decoupling of the high level behavior of robots from collision avoidance routines, such strategy would be unfeasible within the herein proposed model since the stability analysis needs to consider obstacles influence on robots, so as to find a relation between the several components of the algorithm, such as the MANET component presented next.

3.3 Ensuring MANET Connectivity

Robots' position needs to be controlled in order to maintain the communication based on constraints such as maximum distance or minimum signal quality. The way the network will be forced to preserve connectivity depends on communication characteristics (e.g., multi-hop, biconnectivity) [26]. Assuming that the network supports multi-hop connectivity, the communication between two end nodes (i.e., robots) is carried out through a number of intermediate nodes whose function is to relay information from one point to another. Note that any robot may be used as a relay node independently of its swarm. Considering that nodes are mobile, it is necessary to guarantee the existence of a multi-hop communication path between any pair of nodes [7].

In the case of each robot corresponding to a node, in order to overcome the non-connectivity between them, the desired position, i.e., $x_n[t+1]$, must be controlled since it influences the adjacency matrix A. The adjacency matrix, on the other hand, depends on the maximum communication range d_{max} or the minimum signal quality represented by the link matrix $L = \{l_{ij}\}$ for an N-node network, where each entry represents the link between robots i and j (cf. Fig. 1).

One way to ensure the full connectivity of the MANET is to implement the following basic principle: to ensure that each robot is able to communicate with its nearest neighbor which has not already chosen it as its nearest neighbor. Since the connectivity depends on the distance/signal quality, connectivity between nodes may be ensured by computing the minimum/maximum value of each line of link matrix L, after excluding zeros and (i, j) pairs previously chosen. Therefore, the MANET component $\chi_4[t]$ is represented by the position of the nearest neighbor increased by the maximum communication range d_{max} toward the robot's current position. A higher ρ_4

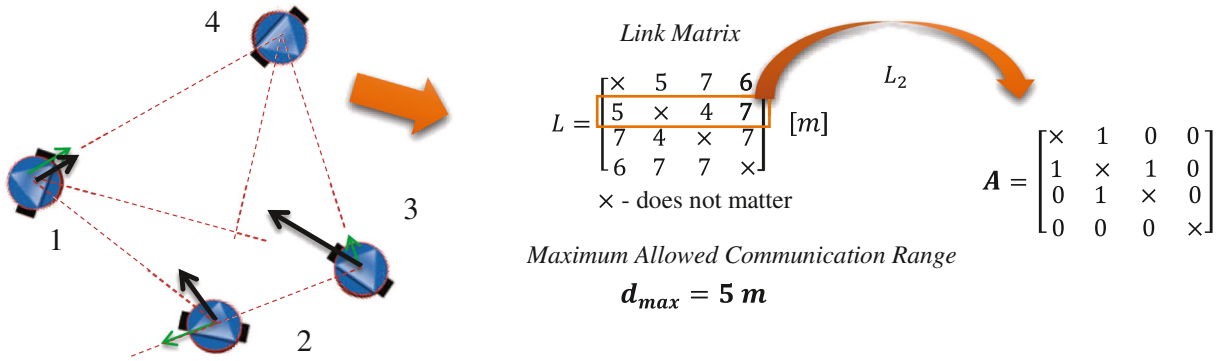

Fig. 1 Illustration of a MANET topology of a swarm. The dashed lines represent the maximum distance d_{max} between each pair of robots and the arrows represent the force vectors that ensure MANET connectivity

may enhance the ability to maintain the network connectedness ensuring a specific range or signal quality between robots.

To further understand how the RDPSO maintains the MANET connectivity, consider the topology in Fig. 1. As it may be perceived, robot 2 is the nearest neighbor of robot 1 and it is also at the correct distance d_{max} (corresponding to 5 m in this example) resulting in a null force connectivity vector. The nearest neighbor of robot 2 is robot 3 which is too close, thus resulting in a repulsive force at robot 2 in order to ensure d_{max}. Finally, the nearest neighbor of robot 3, which has not been chosen already, is robot 4 which is too far away, thus invoking an attraction force affecting robot 3 to move toward robot 4.

Nevertheless, one of the major concerns in this approach is that all robots should have an initial deployment that preserves the communication between the robots in the population. Moreover, it is also known that in classical PSO algorithms particles need to be scattered throughout the scenario. The following section presents the RDPSO initial planar deployment of the robots that preserves the connectivity of the MANET while spreading out the robots as much as possible.

3.4 Initial Deployment

One of the common approaches to the initial deployment of mobile robots is to use a random distribution along the scenario [27]. This methodology is the simplest way of deploying robots since, robots since, in most cases, the distribution of the points of interest is random. However, in real situations, it is necessary to ensure several constraints of the system, such as MANET connectivity, hence increasing the complexity of the random distribution. In addition, random deployment may result in unbalanced deployment and therefore increase the hardware cost.

This approach tries to get the benefits of a random planar deployment of robots while eliminating the disadvantages inherent to it. Furthermore, the herein proposed approach takes into account the communication constraints using a deployment strategy based on the Spiral of Theodorus (aka, square root spiral). This spiral is composed of contiguous right triangles (formerly called rectangled triangles) with each cathetus (aka, leg) having a unit length of 1 [28]. Each of the triangle's hypotenuses gives the square root to a consecutive natural number.

Since this approach uses the Spiral of Theodorus to carry out the initial deployment of robots, two general adjustments need to be considered: i) the initial position of each robot is set at the further vertex of the centre of the spiral for each right triangle with a random orientation; and ii) the size of the cathetus is set as the maximum communication range d_{max} (instead of having the unit length 1) consequently changing the triangles' hypotenuses to the product between the maximum communication range and the square root of the consecutive natural number.

In real situations, the maximum communication distance d_{max} should be established considering the worst case situation (i.e., urban environment). These assumptions make it possible to have an initial deployment of the robots depending on both the number of robots and the communication constraints.

The total angle φ_n of the n^{th} robot situated in the k^{th} triangle (or spiral segment), can be calculated as the cumulative sum:

$$\varphi_n = \sigma_s \sum_{k=1}^{n} arctan\left(\frac{1}{\sqrt{k}}\right), \tag{5}$$

in which σ_s is randomly set to ± 1 for each swarm, thus allowing for a computation of the initial planar position of each robot n as if follows:

$$x_n[0] = x_0 + \begin{bmatrix} d_{max}\sqrt{n+1}\cos(\varphi_n + \varphi_0) \\ d_{max}\sqrt{n+1}\sin(\varphi_n + \varphi_0) \end{bmatrix}, \tag{6}$$

where x_0 and φ_0 are the center and orientation of the spiral which can be randomly assigned at each trial ensuring the efficiency of the stochastic algorithms.

In short, the initial deployment of each swarm of the RDPSO will correspond to a spiral in which the position of each robot depends on the prior deployed robot and the center of the spiral x_0 (Fig. 2). To allow for an autonomous deployment of robots in a scenario, a preprocessing of the environment needs to be undertaken in order to prevent robots from being deployed into areas of no interest (e.g., water, obstacles, other robots). This can be accomplished with unmanned aerial vehicles (UAVs) through image segmentation (cf., [29]).

In the future, since it is out of the scope of this work, further improvements to the deployment strategy will be addressed in order to spare the need of a preprocessing procedure.

3.5 Punish-Reward Mechanism

In the common DPSO [3], natural selection is represented by a "punish"-"reward" mechanism. The "punish" represents the rejetion of particles and swarms, while the "reward" represents the addition of new particles and swarms. In order to adapt DPSO to mobile robotics, the rejetion and addition of a robot are modeled by the mechanisms of social exclusion and social inclusion, respectively [30]. These concepts of social exclusion and inclusion may also be found in

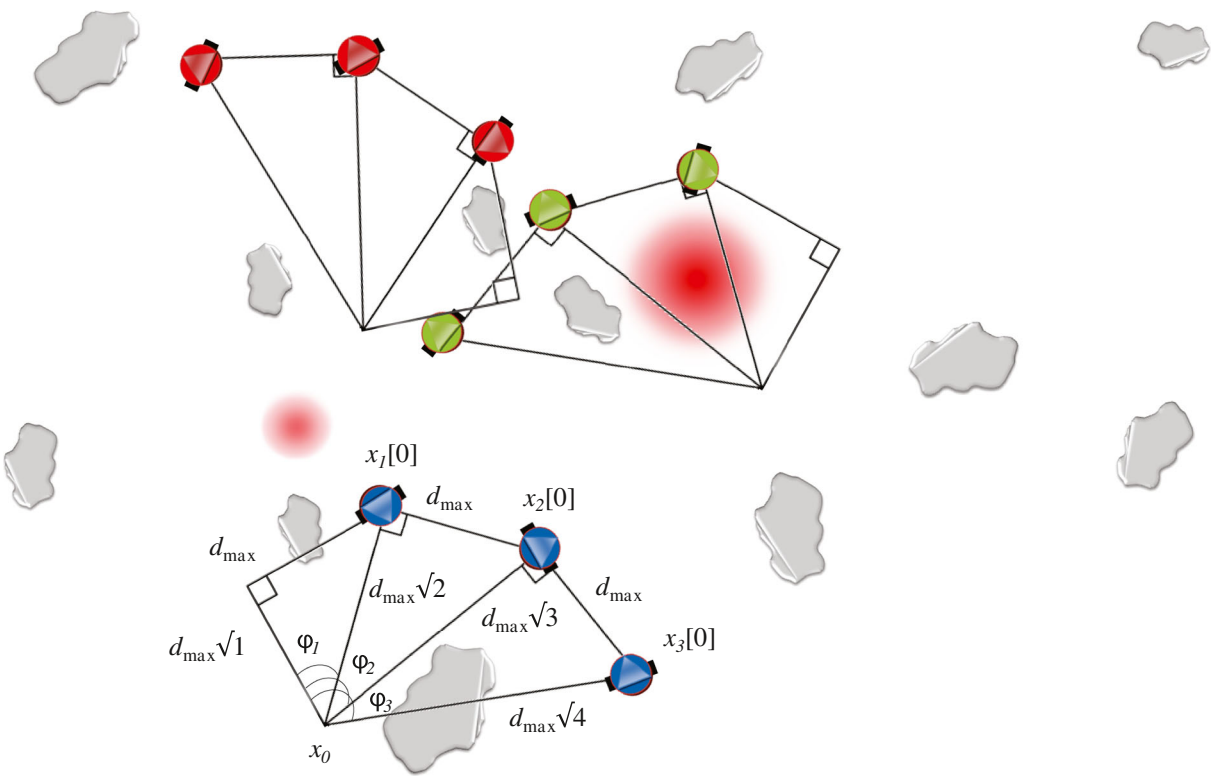

Fig. 2 Initial deployment of the RDPSO algorithm of a population of robots divided in 3 swarms of 3 robots each based on the Spiral of Theodorus [7]

nonhuman animals through stigmatization processes such as:

- Territoriality (e.g., fish, birds, reptiles and mammals) - exclusion of other members of the same species (e.g., certain sex) from an area;
- Status hierarchies (e.g., some bird species, lions, baboons, chimps) - individual at the top of the hierarchy excludes others from resources (e.g., food, territory, mates);
- Social ostracism (e.g., some fish species, lemurs, baboons, chimps) - prevent others from joining social group or forcing expulsion.

For instance, three-spined sticklebacks (a type of fish) avoid others of the same specie with parasites, while Grizzlies (bear) present a hierarchy-related behavior that provides a mechanism that mutes the potential social costs of membership in stable aggregations [31].

The RDPSO is then represented by multiple swarms, i.e., multiple groups of robots that altogether form a population. Each swarm individually performs just like a PSO adapted to multi-robot applications (explained in the previous subsections) in search for the solution and the whole population of robots is controlled by a set of rules. If there was no improvement in a swarm's objective over a period of time, the swarm is punished by excluding the worst performing robot, which is added to a socially excluded group. The worst performing robot is evaluated by the value of its objective function compared to other members in the same swarm. In other words, if the objective is to maximize the fitness function, the robot to be excluded will be the one with the lower fitness value. The RDPSO "punish"-"reward" rules are summarized in Table 1. Alternative strategies to dynamically divide the population into stochastic clusters were previously proposed and results always outperformed the static solutions, thus avoiding sub-optimality and stagnation [32].

The socially excluded robots, instead of searching for the objective function's optimal solution like the other robots in the active swarms, they randomly wander in the scenario. This approach improves the algorithm, making it less susceptible of becoming trapped in a sub-optimal solution. Note, however, that they are always aware of their individual solution and the global solution of the socially excluded group.

Having multiple swarms ensures a distributed approach because the network that was previously defined by the whole population of robots is now divided into multiple smaller networks (one for each swarm), thus decreasing the number of nodes (i.e., robots) and the information exchanged between robots of the same network. In other words, robots interaction with other robots through communication is confined to local interactions inside the same group (swarm), thus making RDPSO scalable to large populations of robots.

To easily understand RDPSO dynamics, let us consider the following geometrical illustration from Fig. 3. Considering a swarm of two robots as Fig. 3

Table 1 Punish-reward RDPSO rules previously introduced in [6]

PUNISH	REWARD
If a socially active subgroup does not improve during a specific threshold SC_{max} (stagnancy counter $SC_s = SC_{max}$) and the number of robots is superior to N_{min} ($N_s > N_{min}$), then the subgroup is punished by socially excluding the worst performing robot	If a socially active subgroup improves and its current number of robots is inferior to N_{max} ($N_s < N_{max}$) and there is, at least, one socially excluded robot, then it is rewarded with the best performing socially excluded robot
If a socially active subgroup does not improve during a specific threshold SC_{max} (stagnancy counter $SC_s = SC_{max}$) and the number of robots is N_{min} ($N_s = N_{min}$), then the subgroup is punished by being dismantled, i.e., all robots from that subgroup are socially excluded	If a socially active subgroup is not stagnated (stagnancy counter $SC_s = 0$) and there are, at least, N_I socially excluded robots, then it has a small probability p_{sp} of spawning a new socially active subgroup

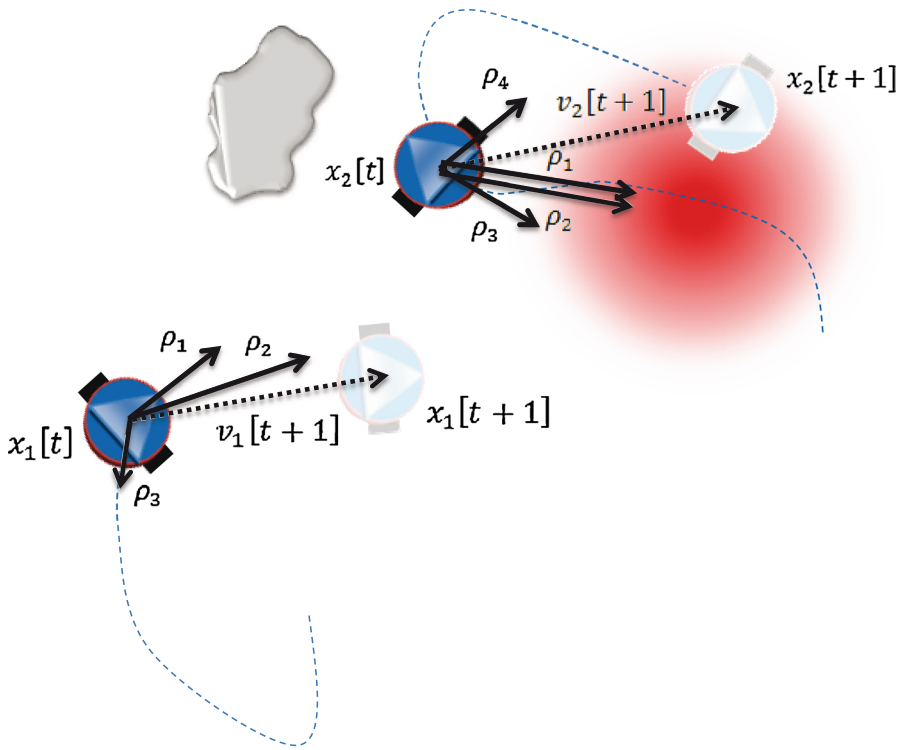

Fig. 3 Geometrical Illustration of the RDPSO using a swarm of two robots. The main variables from Eqs. (1–2) are illustrated as vectors. ρ_1 influences robots to improve their own individual solutions. ρ_2 influences robots toward the same best global position of the swarm. ρ_3 influences robots to move to a previous position in which obstacles were not detected within robot's range. ρ_4 influences robots to maintain a certain maximum distance or minimum signal quality between themselves. α influences the next position of both robots as an inertial factor that considers their trajectory

depicts, the cognitive coefficient ρ_1 influences robots to improve their own individual solutions. In the case of robot 2, both cognitive ρ_1 and social coefficients ρ_2 influence it toward the same position since it is the best performing robot of the swarm. On the other hand, the social coefficient ρ_2 of robot 1 attracts it to the global best position found so far by robot 2. The obstacle susceptibility weight ρ_3 influences robots to move to a previous position in which obstacles were not detected within robot's range. As for the enforcing communication component ρ_4, since robot 2 was the first to choose robot 1 as is nearest neighbor, being at a distance inferior to d_{max}, it is slightly repelled by it. It is noteworthy that the fractional coefficient α influences the next position of both robots, $x_1[t+1]$ and $x_2[t+1]$, as an inertial factor that considers their trajectory.

Algorithm 1 summarizes the distributed RDPSO algorithm that encompasses all of the previous features for robot n when trying to maximize a given objective function. The sensing function $g(x_n[t])$ is monotonically decreasing, i.e., as the robot gets near the obstacle, $g(x_n[t])$ decreases. Also, let us consider that the adjacency matrix depends on the maximum range d_{max} represented by the link matrix $L = \{l_{ij}\}$. Although the RDPSO algorithm was first proposed by the authors in [6], a more formal and detailed definition is done in the algorithm presented in Fig. 4.

Beside the several characteristics inherent to the RDPSO, its performance highly depends on the values of α and ρ_i, $i = 1, 2, 3, 4$, since it is a parameterized swarm algorithm. Therefore, an attraction domain in which parameters may be defined to ensure the convergence of the RDPSO needs to be found. In addition, a suitable relation between the several parameters needs to be defined to contemplate real world constraints.

4 Convergence Analysis

The above presented RDPSO is a stochastic procedure in which Eq. (1) describes the discrete-time motion of a robot with four external inputs $\chi_i[t]$, $i = 1, 2, 3, 4$. The main problem when analyzing this kind of algorithms lies in the fact that external inputs vary in time. However, one can consider that each robot converges to an equilibrium point defined by the limit values of the attractor points χ_i. Therefore, assuming that the

Fig. 4 RDPSO algorithm for robot n

Wait for information about initial pose $\langle x_n[0], \varphi_n[0] \rangle$ and $swarmID$
Loop:
- If $swarmID \neq 0$ // it is not an excluded robot
  - Evaluate its individual solution $h_n[t]$
  - If $h(x_n[t]) > h_{best}$ // robot has improved
    - $h_{best} = h(x_n[t])$ // Section 2
    - $\chi_1[t] = x_n[t]$
  - Exchange information with teammates about the individual solution $h_n[t]$ and current position $x_n[t]$
  - Build a vector $H[t]$ containing the individual solution of all robots within $swarmID$
  - If $\max H[t] > H_{best}$ // swarm has improved
    - $H_{best} = \max H[t]$ // Section 2
    - $\chi_2[t] = x_n[t]$
    - If $SC_s > 0$
      - $SC_s = SC_s - 1$ // stagnancy counter
    - If $SC_s = 0$ // the swarm can be rewarded
      - If $N_S < N_{max}$ and $rand() \frac{1}{N_S^{kill}+1} > rand()$ // small probability of calling a new robot
        - Broadcast the need of a new robot to any available excluded robot // Table 1
        - If $N_s^{kill} > 0$
          - $N_s^{kill} = N_s^{kill} - 1$ // excluded robots counter
      - If $rand() \frac{N_S}{N_{max}} > rand()$ // small probability of creating a new swarm
        - Broadcast the possibility of creating a new swarm to any available excluded robot // Table 1
        - If $N_s^{kill} > 0$
          - $N_s^{kill} = N_s^{kill} - 1$ // excluded robots counter
  - Else // swarm has not improved
    - $SC_s = SC_s + 1$ // stagnancy counter
    - If $SC_s = SC_{max}$ // punish swarm
      - If $N_S > N_{min}$ // it is possible to exclude the worst performing robot
        - $N_s^{kill} = N_s^{kill} + 1$ // excluded robots counter
        - $SC_s = SC_{max}\left[1 - \frac{1}{N_S^{kill}+1}\right]$ // reset search counter
        - If $h_{best} = \min H[t]$ // this is the worst performing robot
          - $swarmID = 0$ // exclude this robot
      - Else // delete the entire swarm
        - $swarmID = 0$ // exclude this robot
  - If $g(x_n[t]) \geq g_{best}$ // maximize distance to obstacles
    - $g_{best} = g(x_n[t])$ // Section 2.2
    - $\chi_3[t] = x_n[t]$
  - $[L_n, index_n] = sort\_ascending(L_{n,1:N_S})$ // sort the elements of line n from link matrix L in ascending order
    - For $i = 1{:}\,N_S$
      - If $index_n(i)$ has not yet chosen it as its nearest neighbor
        - $\chi_4[t] = x_i[t] + d_{max} \frac{x_i[t] - x_n[t]}{\|x_i[t] - x_n[t]\|}$ // the position of the nearest neighbor increased by d_{max} toward $x_n[t]$
        - Communicate to robot i that it was chosen by robot n // Section 2.3
        - break from For
  - $v_n[t+1] = w_n[t] + \sum_{i=1}^{4} \rho_i r_i (\chi_i[t] - x_n[t])$ // equation 1
  - $x_n[t+1] = x_n[t] + v_n[t+1]$ // equation 2
- Else // it is an excluded robot
  - Wandering algorithm // *e.g.*, [28]
  - Evaluate its individual solution $h_n[t]$
  - If $h(x_n[t]) > h_{best}$ // robot has improved
    - $h_{best} = h(x_n[t])$
  - Exchange information with teammates about the individual solution $h_n[t]$ and current position $x_n[t]$
  - Build a vector $H[t]$ containing the individual solution of all N_X robots within the excluded swarm ($swarmID = 0$)
  - If $\max H[t] > H_{best}$
    - $H_{best} = \max H[t]$
    - If $h_{best} = \max_{N_I} H[t]$ // this is one of the best N_I performing robot of the excluded swarm
      - If $N_X \geq N_I$ and $rand() \frac{N_X}{N_T} > rand()$ // small probability of creating a new swarm
        - $swarmID = swarmID\_new$ // include this robot in the new active swarm
        - Broadcast the need of $N_I - 1$ robots to any available excluded robot // Table 1
      - Else
        - If receives information about the need of a new robot
          - $swarmID = swarmID\_received$ // include this robot in the active swarm
          - $N_S = N_S + 1$
          - Exchange information with teammates about N_S
        - If receives information about the need of creating a new swarm
          - $swarmID = swarmID\_new$ // include this robot in a new active swarm
          - $N_S = N_I$ // reset number of robots in the swarm
          - $N_s^{kill} = 0$ // reset number of excluded robots
          - $SC_s = 0$ // reset search counter

until stopping criteria (convergence/time)

algorithm converges, this section presents the stability analysis of the RDPSO. Although this topic was previously addressed in [8], this paper presents a more detailed description on how to obtain the global attraction domain and how this may be reduced to a particular attraction domain based on robot constraints. Moreover, the influence of each RDPSO parameter is assessed by means of several experiments with a swarm of two physical robots.

4.1 Problem Statement

Consider a population of N_T robots wherein each robot needs to cooperatively find the optimal solution of a given mission within its swarm. The goal is to find the attraction domain $\mathscr{A}$ such that, if coefficients $\alpha, \rho_i \in \mathscr{A}, i = 1, 2, 3, 4$, the global asymptotic stability of the system defined by (1) and (2) is guaranteed. In other words, the attraction domain $\mathscr{A}$ represents the region wherein RDPSO parameters may be defined in such a way that robots can find the optimal solution while avoiding obstacles and ensuring MANET connectivity.

4.2 General Approach

Knowing that $v_n[t-k] = x_n[t-k] - x_n[t-(k+1)]$ with $k \in \mathbb{N}_0$, and considering Eqs. (1) and (2), one can rewrite the RDPSO equation as a nonhomogeneous five-order difference equation:

$$\begin{aligned} &x_n[t+1] + \left(-1-\alpha+\sum\nolimits_{i=1}^{4}\rho_i r_i\right) x_n[t] \\ &+ \left(\frac{1}{2}\alpha\right) x_n[t-1] + \left(\frac{1}{3}\alpha + \frac{1}{6}\alpha^2\right) x_n[t-2] + \\ &\quad + \left(-\frac{1}{24}\alpha^3 - \frac{1}{24}\alpha^2 + \frac{1}{12}\alpha\right) x_n[t-3] \\ &+ \left(\frac{1}{24}\alpha^3 - \frac{1}{8}\alpha^2 + \frac{1}{12}\alpha\right) x_n[t-4] \\ &\quad = \sum\nolimits_{i=1}^{4}\rho_i r_i \chi_i[t]. \end{aligned} \tag{7}$$

The equilibrium point x_n^* can be defined as a constant position solution of (7), such that, when each robot reaches x_n^*, the velocity $v_n[t+k]$ is zero, i.e., robots will stop at the equilibrium point x_n^*. Supposing that χ_i are constants, i.e., the algorithm does converge, the particular solution x_n^* of each robot can be obtained replacing $x_n[t+1-k]$ in Eq. (7) by x_n^*:

$$\begin{aligned} &x_n^* + \left(-1-\alpha+\sum\nolimits_{i=1}^{4}\rho_i r_i\right) x_n^* + \left(\frac{1}{2}\alpha\right) x_n^* \\ &+ \left(\frac{1}{3}\alpha + \frac{1}{6}\alpha^2\right) x_n^* + \left(-\frac{1}{24}\alpha^3 - \frac{1}{24}\alpha^2 + \frac{1}{12}\alpha\right) x_n^* + \\ &+ \left(\frac{1}{24}\alpha^3 - \frac{1}{8}\alpha^2 + \frac{1}{12}\alpha\right) x_n^* = \sum\nolimits_{i=1}^{4}\rho_i r_i \chi_i[t] \Leftrightarrow x_n^* \\ &= \frac{\sum_{i=1}^{4}\rho_i r_i \chi_i}{\sum_{i=1}^{4}\rho_i r_i}. \end{aligned} \tag{8}$$

In other words, each robot will converge to the particular solution x_n^*, based on the following theorems [33]:

Theorem 1 [33] *All solutions of* (7) *converge to* x_n^* *as* $t \to \infty$, *if and only if the homogeneous difference equation of* (7) *is asymptotically stable.*

Theorem 2 [33] *The homogeneous difference equation of* (7) *is asymptotically stable if and only if all roots of the corresponding characteristics equation have modulus less than one.*

In order to study the homogeneous difference Eq. (7) *stability, let us consider the following characteristic equation:*

$$\begin{aligned} &p(\lambda) \equiv \lambda^5 + \left(-1-\alpha+\sum\nolimits_{i=1}^{4}\rho_i r_i\right)\lambda^4 + \left(\frac{1}{2}\alpha\right)\lambda^3 \\ &+ \left(\frac{1}{3}\alpha + \frac{1}{6}\alpha^2\right)\lambda^2 + + \left(-\frac{1}{24}\alpha^3 - \frac{1}{24}\alpha^2 + \frac{1}{12}\alpha\right)\lambda \\ &\quad + \left(\frac{1}{24}\alpha^3 - \frac{1}{8}\alpha^2 + \frac{1}{12}\alpha\right) = 0. \end{aligned} \tag{9}$$

Due to the complexity in obtaining the roots of the characteristics equation of homogeneous difference Eq. (7), a result based on Jury-Marden's Theorem [34] is established which ensures that all roots of the real polynomial $p(\lambda)$ have modulus less than one.

Theorem 3 [34] *Consider the real polynomial* $p(y) = a_0 y^n + a_1 y^{n-1} + \cdots + a_{n-1} y + a_n$, $a_0 > 0$. *Construct an array having initial rows:*

$$\begin{aligned} &\{c_{11}, c_{12}, \ldots, c_{1,n+1}\} = \{a_0, a_1, \ldots, a_n\}, \\ &\{d_{11}, d_{12}, \ldots, d_{1,n+1}\} = \{a_n, a_{n-1}, \ldots, a_0\}, \end{aligned}$$

and subsequent rows defined by:

$$c_{\beta\gamma} = \begin{vmatrix} c_{\beta-1,1} & c_{\beta-1,\gamma+1} \\ d_{\beta-1,1} & d_{\beta-1,\gamma+1} \end{vmatrix}, \beta = 1, 2, \ldots, n+1$$

$$d_{\beta\gamma} = c_{\beta,n-\gamma-\beta+3}$$

All roots of the polynomial $p(y)$ have modulus less than one if and only if $d_{21} > 0$, $d_{\tau 1} < 0$ ($\tau = 3, 4, \ldots, n+1$).

Therefore, let us present the following result.

Proposition 1 *All roots of $p(\lambda)$ have modulus less than one if and only if the following conditions are met.*

$$\begin{cases} 0 < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2 & , 0 < \alpha \leq 0.6 \\ \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2 & , 0.6 < \alpha \leq 1 \end{cases}. \tag{10}$$

Proof The real polynomial $p(\lambda)$ described in Eq. (9) can be rewritten as:

$$a_0\lambda^5 + a_1\lambda^4 + a_2\lambda^3 + a_3\lambda^2 + a_4\lambda + a_5 = 0, \tag{11}$$

Furthermore, one can construct an array having initial rows defined as:

$$\begin{aligned} &c_{11}, c_{12}, \ldots, c_{16} = a_0, a_1, \ldots, a_5, \\ &d_{11}, d_{12}, \ldots, d_{16} = a_5, a_4, \ldots, a_0, \end{aligned} \tag{12}$$

and subsequent rows defined by:

$$c_{\beta\gamma} = \begin{vmatrix} c_{\beta-1,1} & c_{\beta-1,\gamma+1} \\ d_{\beta-1,1} & d_{\beta-1,\gamma+1} \end{vmatrix}, \tag{13}$$

$$d_{\beta\gamma} = c_{\beta,8-\gamma-\beta}, \tag{14}$$

where $\beta = 2, 3, 4, 5, 6$ and $\gamma = 0, 1, 2, 3$.

By Theorem 3, we consider that all roots of polynomial $p(\lambda)$ have modulus less than one if and only if $d_{21} > 0, d_{\tau 1} < 0$, for $\tau = 3, 4, 5, 6$. Hence,

$$\begin{cases} d_{21} > 0 \\ d_{31} < 0 \\ d_{41} < 0 \\ d_{51} < 0 \\ d_{61} < 0 \end{cases} \Leftrightarrow \begin{cases} 1 - a_5^2 > 0 \\ (1 - a_5a_1)^2 - d_{21}^2 < 0 \\ ((1-a_5a_1)(a_1-a_5a_4)-d_{21}(a_3-a_5a_2))^2 - d_{31}^2 < 0 \\ c_{41}^2 - d_{41}^2 < 0 \\ c_{51}^2 - d_{51}^2 < 0 \end{cases} \tag{15}$$

Solving (15) we obtain:

$$\begin{cases} 0 < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2 & , 0 < \alpha \leq 0.6 \\ \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2 & , 0.6 < \alpha \leq 1 \end{cases}. \tag{16}$$

□

Consequently, by Proposition 1, Theorem 1 and Theorem 2, the conditions in (10) are obtained so that all solutions of (7) converge to x_n^* resulting in a global attraction domain $\mathscr{A} = \{\alpha, \rho_i : 0 < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2, 0 < \alpha \leq 0.6 \wedge \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4} \rho_i r_i \leq \alpha + 2, 0.6 < \alpha \leq 1; i = 1, 2, 3, 4\}$ (cf., Fig. 5). Although it was possible to define a relatively small attraction domain, next section further explores particular conditions of the algorithm, by redefining and adjusting parameters values.

4.3 Robot Constraints

One way to improve the convergence analysis of the algorithm consists of adjusting the parameters based on physical mobile robots constraints, such as acceleration and deceleration states inherent to their dynamical characteristics. These states are usually unaddressed in the literature while analyzing the traditional PSO and its main variants, since virtual agents (i.e., particles) are not constrained by such behaviors. Let us then suppose that a robot is traveling at a constant velocity such that $v_n[t-k] = v$ with $k \in \mathbb{N}_0$ and it is able to find its equilibrium point in such a way that $x_n[t] = \chi_i$, $i = 1, 2, 3, 4$. In other words, the best position of the cognitive, social, obstacle and MANET components is the same. As a result, the robot needs to decelerate until it stops, i.e., $v > v_n[t+1] \geq \cdots \geq v_n[t+k] \geq \cdots \geq 0$.

Consequently, Eqs. (1) and (4) can be rewritten as:

$$0 \leq v\left(\alpha + \frac{1}{2}\alpha + \frac{1}{6}\alpha(1-\alpha) + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)\right) < v, \tag{17}$$

thus resulting in

$$0 < \alpha \leq 0.632. \tag{18}$$

Let us now consider the opposite scenario, i.e., a robot that has stopped $v_n[t-k] = 0$ with $k \in \mathbb{N}_0$ starts to move since $x_n[t] \neq \chi_i$, $i = 1, 2, 3, 4$. The robot needs to accelerate until it reaches the maximum velocity defined by Eq. (1), taking into account that $w_n[t] = 0$.

Similarly to the procedure presented in (7), but considering the previously described conditions, the following nonhomogeneous first-order difference equation results:

$$x_n[t+1]+\left(\sum_{i=1}^{4}\rho_i r_i - 1\right)x_n[t] = \sum_{i=1}^{4}\rho_i r_i \chi_i[t]. \quad (19)$$

Hence, the characteristic equation associated to (19) is

$$p_1(\lambda) \equiv \lambda + \left(\sum_{i=1}^{4}\rho_i r_i - 1\right) = 0. \quad (20)$$

Proposition 2 *The homogeneous difference equation of* (19) *is asymptotically stable if and only if*

$$0 < \sum_{i=1}^{4}\rho_i r_i < 2. \quad (21)$$

Proof Based on Theorem 2 one can consider that the homogeneous difference Eq. (19) is asymptotically stable if and only if the root of $p_1(\lambda)$ have modulus less than one. Therefore,

$$\begin{aligned} p_1(\lambda) = 0 &\Leftrightarrow \lambda + \left(\sum_{i=1}^{4}\rho_i r_i - 1\right) \\ &= 0 \Leftrightarrow \lambda = -\left(\sum_{i=1}^{4}\rho_i r_i - 1\right) \end{aligned} \quad (22)$$

Then,

$$\begin{aligned} |\lambda| < 1 &\Leftrightarrow \left|-\left(\sum_{i=1}^{4}\rho_i r_i - 1\right)\right| \\ &< 1 \Leftrightarrow 0 < \sum_{i=1}^{4}\rho_i r_i < 2. \end{aligned} \quad (23)$$

□

Consequently, by Proposition 2 and Theorem 1, the conditions in (21) are obtained so that all solutions of (19) converge to x_n^* resulting in a particular attraction domain $\mathscr{A}_p$.

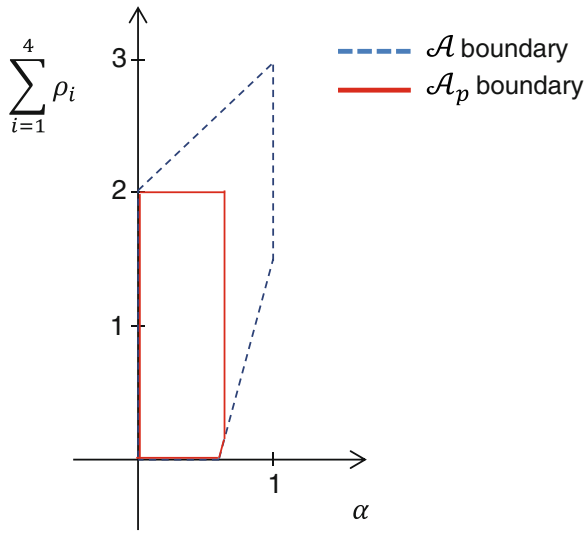

Fig. 5 Global attraction domain $\mathscr{A} = \{\alpha, \rho_i : 0 < \sum_{i=1}^{4}\rho_i r_i \leq \alpha + 2, 0 < \alpha \leq 0.6 \wedge \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4}\rho_i r_i \leq \alpha + 2, 0.6 < \alpha \leq 1; i = 1, 2, 3, 4\}$ and particular attraction domain $\mathscr{A}_p = \{\alpha, \rho_i : 0 < \sum_{i=1}^{4}\rho_i \leq 2, 0 < \alpha \leq 0.6 \wedge \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4}\rho_i \leq 2, 0.6 < \alpha \leq 0.632; i = 1, 2, 3, 4\}$ of the asymptotic stability of the RDPSO

However, since r_i randomly varies between 0 and 1, such that $\max r_i = 1, i = 1, 2, 3, 4$, condition (21) can be rewritten as:

$$0 < \sum_{i=1}^{4}\rho_i < 2. \quad (24)$$

Hence, the particular attraction domain $\mathscr{A}_p = \{\alpha, \rho_i : 0 < \sum_{i=1}^{4}\rho_i \leq 2, 0 < \alpha \leq 0.6 \wedge \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4}\rho_i \leq 2, 0.6 < \alpha \leq 0.632; i = 1, 2, 3, 4\}$ is represented by the parameter region, i.e., attraction domain, of the asymptotic stability depicted in Fig. 5.

As a result of the above analysis, the RDPSO can be conceived in such a way that the system's convergence can be controlled by taking into account obstacle avoidance and MANET connectivity, without resorting to the definition of any arbitrary or problem-specific parameters. However, the influence of each individual parameter ρ_i and fractional coefficient α in the performance of the algorithm needs to be further explored in order to systematically adjust the collective behavior of the swarm.

4.4 Preliminary Evaluation

To evaluate the parameters' impact within the RDPSO algorithm, a swarm of two physical robots was used in the following set of experiments (Fig. 6). The robots consisted on eSwarBots differential ground platforms recently developed and presented in [23] for swarm robotics applications. Although the platforms present a limited odometric resolution of 3.6 ° while rotating and 2.76 mm when moving forward, their low cost and high autonomy enable the performance of experiments with large number of robots.

As described in [35] and [36], a swarm behavior can be divided into two activities: i) exploitation; and ii) exploration. If the exploitation level is too high, then the algorithm may get stuck on sub-optimal solutions. However, if the exploration level is too high, the algorithm may take too much time to find the optimal solution. In order to understand the relation between the fractional coefficient α and exploitation/exploration capabilities of the RDPSO, the center-of-mass trajectory in phase space of the swarm of 2 eSwarBots, for various values of α, while fixing $\rho_i = 0.49$, was analyzed. Both robots were randomly deployed in the vicinity of the solution in $(0, 0)$ with a fixed distance of 0.5 m between them. The solution was defined by an illuminated spot which was sensed using the overhead light sensors (LDR) equipped on the platforms (cf, Fig. 6).

As it may be perceived in Fig. 7a, the behavior of the swarm is susceptible to changes in α. When α is too small, i.e., $\alpha = 0.01$, the exploitation level is too high and likely to get stuck in a sub-optimal solution.

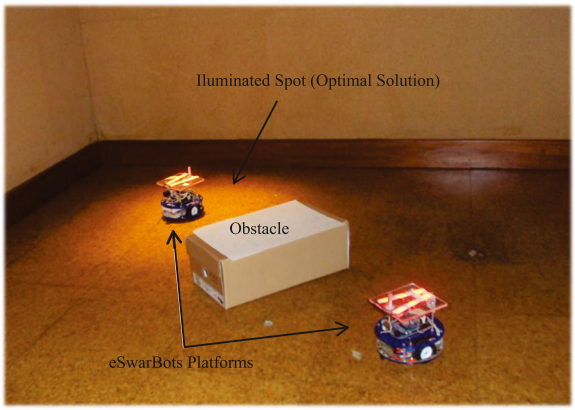

Fig. 6 Illustrative example of the preliminary results to study the RDPSO algorithm on two eSwarBot platforms

However, the intensification of the algorithm convergence is improved - it presents a quick almost linear convergence. When α is at the boundary of the particular attraction domain $\mathscr{A}_p$, i.e., $\alpha = 0.632$, the trajectory of the swarm is cyclical and presents a good balance between exploitation and exploration. In this case, robots exhibit a level of diversification adequate to avoid local solutions and yet at a considerable level of intensification to converge to the optimal solution - the trajectory of the swarm is represented by a spiral convergence toward a nontrivial attractor. When α is too high and outside $\mathscr{A}_p$, i.e., $\alpha = 0.99$, despite the cyclical trajectory of the swarm toward the global solution, the swarm presents an oscillatory behavior, thus having a high exploration level but being more unstable and sometimes unable to converge - it presents a difficult convergence. The values of the cognitive and social factors ρ_1 and ρ_2 are not critical for the algorithm convergence, but the selection of proper values may result in superior performance, both in terms of speed of convergence and sub-optimal solutions avoidance. To further understand the cognitive and social components of the RDPSO, the 2 eSwarBots were placed near the local and global solution uniquely identifiable by controlling the brightness of the light - the brighter site (optimal solution) was considered better than the dimmer one (sub-optimal solution), and so the goal of the swarm was to collectively choose the brighter site (cf., Fig. 8 from next section visualizes the experimental setup). At the beginning, robots were at a distance of 1.60 m from each other. Also, the fractional coefficient α was fixed at 0.632 (i.e., the threshold stability) and $\rho_3 = \rho_4 = 0.1$ for multiple (ρ_1, ρ_2) combinations while keeping the same absolute value $\rho_T = 1$ with $\rho_T = \rho_1 + \rho_2$. Figure 7b presents the Euclidean distance in phase space between the 2 robots, thus depicting the evolution and convergence of the distance between them. As expected, increasing the social weight ρ_2 decreases the Euclidean distance between robots - the distance between robots inclines to only a few centimeters when using $(\rho_1, \rho_2) = (0.1, 0.9)$ and near 1 meter using $(\rho_1, \rho_2) = (0.9, 0.1)$. However, the relation between the inter-robot final distance and (ρ_1, ρ_2) weights is not linear. It can also be observed that, after increasing the social weight ρ_2, the robot initially located at the sub-optimal solution converges in a more intensive way, i.e., the radius of the spiral at

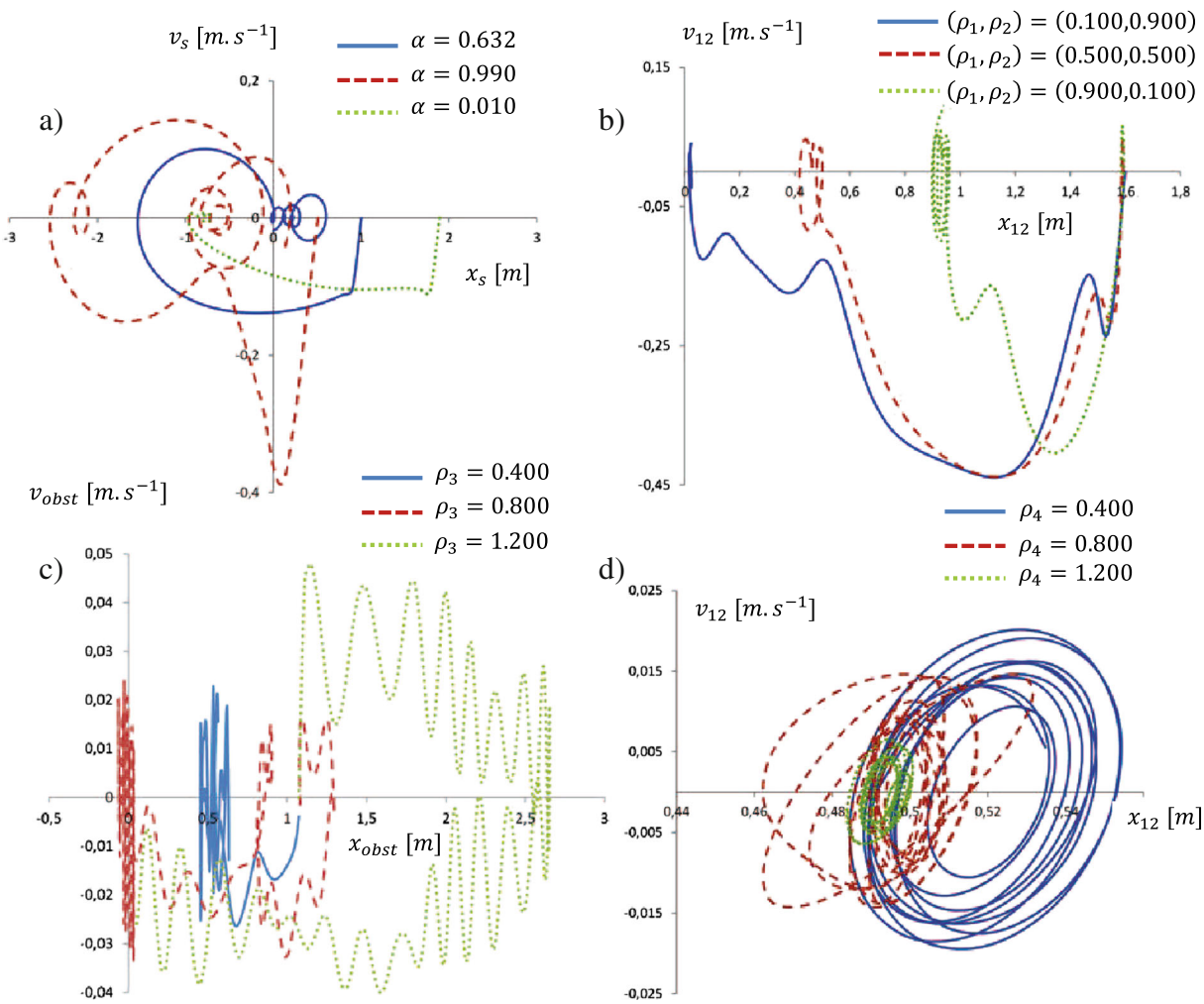

Fig. 7 Analysis of real robot constraints in a swarm of 2 robots. **a** center-of-mass trajectories in phase space to evaluate α; **b** Euclidean distance between robots in phase space to evaluate the relation between ρ_1 and ρ_2; **c** distance from the worst performing robot to the obstacle in phase space to evaluate ρ_3; **d** Euclidean distance between robots in phase space to evaluate ρ_4

the convergence point is smaller for higher ρ_2 values. Hence, the exploitation behavior increases as the distance between robots decreases, thus compromising the performance of the swarm. Moreover, robots' velocities do not directly depend on the relation between ρ_1 and ρ_2, since the relative velocity between robots reached a maximum velocity of approximately 0.45 $m.s^{-1}$ in all three (ρ_1, ρ_2) combinations.

However, independently of (ρ_1, ρ_2) combinations, robots may be unable to complete their main mission if they cannot efficiently avoid obstacles. The presence or absence of obstacles can affect the efficiency of the RDPSO - one set of parameters may result in a faster convergence but may fail in the presence of obstacles or it may increase obstacles susceptibility though swarms may be more resilient. As previously explained, a robot is able to avoid obstacles due to a repulsive force based on a monotonic and positive sensing function $g(x_n[t])$ that depends on the distance between the robot and the obstacle. To better understand the relation between the obstacle susceptibility weight ρ_3 and the other RDPSO parameters, one of the robots was placed in the optimal position (i.e., the brighter site) and the other robot was placed 1 meter away from it. Also, an obstacle was placed halfway the path described by the latter robot toward the optimal position (Fig. 6). Robots were programmed to detect obstacles 0.5 m away from them, i.e., $r_s = 0.5$. In

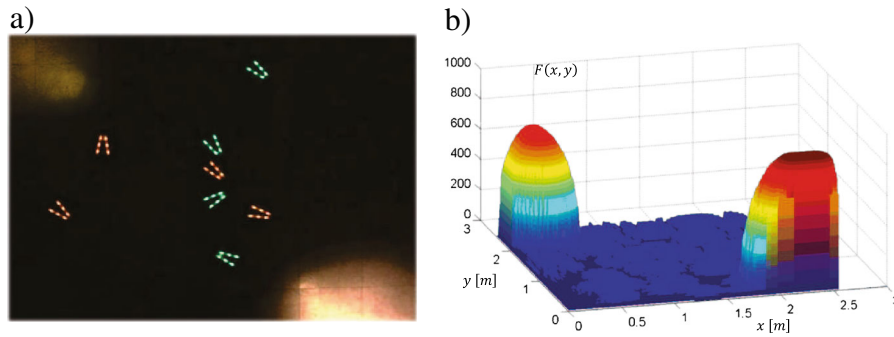

Fig. 8 Experimental setup. **a** Enclosed arena with 2 swarms (*different colors*) of 4 eSwarBots each; **b** Virtual representation of the target distribution (i.e., intensity of light) retrieved sweeping the scenario with a single robot

order to allow the manipulation of ρ_3 within a larger range while respecting condition (21), $\rho_4 = 0.1$ and $\rho_T = 0.7$ with $(\rho_1, \rho_2) = (0.2, 0.5)$, i.e., the social component influence is stronger than the cognitive one.

The obstacle susceptibility of the robot was evaluated fixing $\alpha = 0.632$ for $\rho_2 = 0.4, 0.8, 1.2$. Observing Fig. 7c, one can conclude that the worst performing robot gets stuck (and even sometimes collides) in the vicinity of the obstacle, only for an obstacle susceptibility weight of $\rho_3 = 0.4$. For any of the other two situations, the robot is able to circumvent the obstacle, thus reaching the optimal solution. However, notwithstanding the same final results for both $\rho_3 = 0.8$ and $\rho_3 = 1.2$, as ρ_3 increases the robot presents a more chaotic behavior, i.e., more oscillatory. For $\rho_3 = 1.2$ the robot first moves 1 m and a half away from its current location avoiding the obstacle in an inadequate way.

Finally, and to completely fulfill MRS requirements, a way to ensure the MANET connectivity needs to be considered. Robots within the same swarm should spread out as much as possible in order to improve the convergence rate of the RDPSO algorithm. However, they must keep a certain range between themselves. Therefore, one needs to find a suitable tradeoff between the enforcing communication component ρ_4 and the mission parameters (i.e., ρ_1 and ρ_2), since each robot has to plan its moves while maintaining the MANET connectivity. As previously presented, the RDPSO takes use of the adjacency matrix A that directly depends on link matrix $L = \{l_{ij}\}$ to identify the minimum/maximum distance/signal quality of each line, thus returning the position of the nearest neighbor in which a robot needs to ensure connectivity. In this experiment, one of the robots was placed at a distance of 0.5 m away from the global best position, thus being attracted to it. The other robot was placed 0.5 m away from the first robot (1 m away from the optimal solution, i.e., global best position) and was not allowed to move, thus simulating an internal failure (e.g., mechanical flaw). To maintain the MANET connectivity, the robots must keep a maximum distance between them of $d_{max} = 0.5$ meter. The distance between robots, x_{12}, is evaluated by manipulating ρ_4 while satisfying condition (24), with $\rho_3 = 0.1$, $(\rho_1, \rho_2) = (0.2, 0.5)$ and $\alpha = 0.632$. The enforcing communication component was set as $\rho_4 = 0.4, 0.8, 1.2$. It may be observed in Fig. 7d that, for any ρ_4, the robot presents a spiral convergence in d_{max} vicinities. However, as increases, the convergence of the robot toward d_{max} also increases (the center of the spiral approximates d_{max}). For $\rho_4 = 0.4$ the robot converges toward a distance superior to d_{max} with a larger spiral radius while trying to get closer to the solution. For $\rho_4 = 1.2$ the robot ignores the solution and hardly moves from its initial position.

Given the above convergence analysis, next section presents experimental results obtained using both real and simulated robots to evaluate the RDPSO.

5 Experimental Evaluation

The previous section shows the theoretical convergence of the RDPSO without considering the mechanism of social exclusion. Nevertheless, this mechanism allows for a larger diversity of the solution, thus providing an escape to sup-optimal solutions that could be wrongly taken as the optimal one (i.e., robots would think that they reached the equilibrium point x_n^*). Therefore, to further validate the claims around the RDPSO and based on the convergence analysis

previously presented, this section provides experimental results obtained using both real and simulated robots.[1]

5.1 Evaluation on eSwarBots

In this section, the effectiveness of using the RDPSO is explored on swarms of eSwarBots, while performing a collective foraging task with local and global information under communication constraints. Since the RDPSO is a stochastic algorithm, it may lead to a different trajectory convergence whenever it is executed. Therefore, test groups of 20 trials of 180 seconds each were considered for $N_T = 8, 12$ eSwarBots. A minimum, initial and maximum number of 1, 2 and 3 swarms were used. The maximum travelled distance between iterations was set as 0.15 m, i.e., $\max |x_n[t+1] - x_n[t]| = 0.15$.

It is noteworthy that trying to maintain the network connectivity by only taking into account the communication range d_{max} does not match reality since the propagation model is more complex, i.e., the signal depends not only on the distance but also on the multiple paths from walls and other obstacles. However, in simulation or small and obstacle free scenarios, the communication distance is a good approach and it is easier to implement. Therefore, for the sake of simplicity and without lack of generality, the distance criterion d_{max} was used to model communication constraints, with $d_{max} = 0.5$ m.

The experimental environment was the same used to assert the relation between parameters in Section 4.4 and is represented in Fig. 8a. eSwarBots are equipped with RGB-LEDs that allow representing a wide range of different colors to depict different swarms. Active swarms are identified using the primary colors red, green and blue in which the last one only appears when a third swarm is created (cf., Table 1 in Section 3.5). Robots within the socially excluded swarm are identified with a white light. Despite being an obstacle free scenario of 2.55 × 2.45 m, the robots themselves act as dynamic obstacles - note that a maximum number of 12 robots correspond to a population density of approximately $2\ robots \times m^{-2}$. Inter-robot communication to share positions and individual solutions were carried out using ZigBee 802.15.4 wireless protocol. Since eSwarBots are equipped with XBee modules that allow a maximum communication range larger than the whole scenario, robots were provided with a list of their teammates' addresses in order to simulate the ad-hoc multi-hop network communication with limited range. At each trial, robots were manually deployed on the scenario in a spiral manner (as previously presented in Section 3.4 and [7]) while preserving the maximum communication distance d_{max}.

As Fig. 8b depicts, the objective function is represented by a sub-optimal and an optimal solution. The main objective of robots is to find the brighter site (optimal solution). The intensity values $F(x, y)$ represented in Fig. 8b were obtained sweeping the whole scenario with a single robot in which the light sensor was connected to a 10-bit analog input, thus offering a resolution of approximately 5 mV. To improve the interpretation of the algorithm performance, results were normalized in a way that the objective of the robotic teams was to find the optimal solution of $f(x, y) = 1$.

The algorithm parameters were chosen in order to satisfy the conditions presented previously. However, to further explore the relation between parameters and population size, two different sets of values were analyzed (Table 2).

Each set of parameters $S_j, j = 1, 2$, is defined by the tuple $\{\alpha, \rho_1, \rho_2, \rho_3, \rho_4\}$. Hence, the only difference between both sets S_1 and S_2 is how RDPSO parameters are defined within the particular attraction domain $\mathscr{A}_p$ previously represented in Fig. 5. The first set (S_1) is more conservative with higher ρ_3 and ρ_4 than S_2, thus allowing robots to preserve the MANET connectivity and avoid obstacles collision at any cost. The second set (S_2) is greedier with higher ρ_1 and ρ_2 than S_1, wherein robots' primary concern is to find the optimal solution (even if some collisions or MANET ruptures occur). Figure 9 depicts the performance of the algorithm, by changing the total number of robots N_T and the set of parameters. Since these experiments represent a search task, it is necessary to evaluate the completeness (i.e., final solution) and speed of the mission (i.e., runtime). Therefore, the median of the best solution in the 20 trials was taken as a final output for each different condition.

[1] Videos from real and simulated experiments are available at http://paloma.isr.uc.pt/$\sim$micaelcouceiro/media/media.htm

Table 2 Two sets of RDPSO parameters inside the particular attraction domain $\mathscr{A}_p = \{\alpha, \rho_i : 0 < \sum_{i=1}^{4} \rho_i \leq 2, 0 < \alpha \leq 0.6 \wedge \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4} \rho_i \leq 2, 0.6 < \alpha \leq 0.632; i = 1, 2, 3, 4\}$

Parameters	α	ρ_1	ρ_2	ρ_3	ρ_4
S_1	0.632	0.100	0.300	0.790	0.790
S_2	0.632	0.200	0.400	0.690	0.690

Analyzing Fig. 9, it is clear that the proposed mission can be accomplished by any number of robots between 8 and 12. In fact, independent of on the number of robots, teams converge to the solution in approximately 90 % of the experiments. The charts also show that, independently of the set of parameters, increasing the number of robots from 8 to 12 decreases the time needed to find the solution. A population of 8 robots, for both sets of parameters, takes approximately 119 (S_1) and 138 (S_2) seconds to converge to the optimal solution. On the other hand, a population of 12 robots takes approximately 56 (S_1) and 112 (S_2) seconds to converge. Yet, either with 8 or 12 robots, robots seems to perform better using the first set of parameters. In other words, using a greedier behavior (S_2) over a conservative one (S_1), in which robots prioritize finding the optimal solution over maintaining the MANET connectivity and obstacles avoidance, decreases the RDPSO performance.

Another important factor is that some robots of a given swarm are unable to converge to the final solution when one robot of the same swarm finds it. This issue is related with odometry limitations of the platforms which results in the accumulation of positioning errors. The use of encoders, such as the ones implemented in these robots, is a classical method for their low-cost and simplicity of use. However, the errors, inherent to the use of such encoders, are cumulative, which makes it difficult for the robots to complete the proposed odometry objectives accurately. However, it is noteworthy that all robots within the same swarm agree with the best solution.

Nevertheless, analyzing swarm algorithms, as the RDPSO, within small populations of 8 or 12 robots may not represent the required collective performance. In fact, how well can a population of up to 12 robots reveal the RDPSO performance if swarm algorithms usually need a large number of agents (i.e., dozens, hundreds, or even thousands) for the collective intelligence to emerge (cf., [37])? Also, is it enough to assess the RDPSO performance within the small proposed scenario under only one target distribution?

To overcome such weaknesses, next section presents computational experiments using a larger population of simulated robots within larger scenarios under different kinds of benchmark functions (i.e., target distribution) commonly used in optimization algorithms.

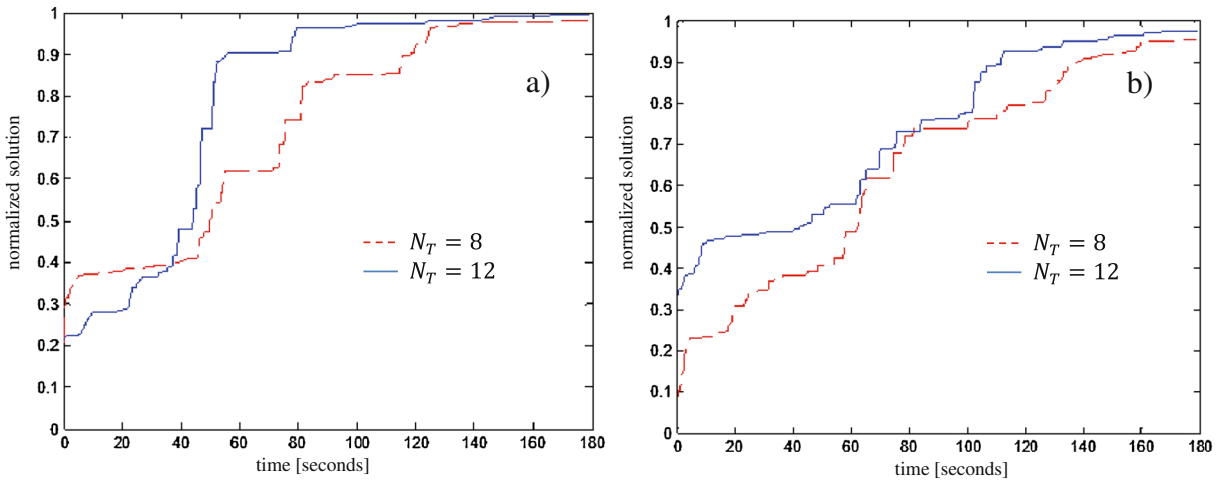

Fig. 9 RDPSO evaluation changing the number of robots N_T and set of parameters: **a** S_1; **b** S_2

5.2 Computational Experiments

The use of simulated robots instead of the physical ones was necessary to further evaluate the RDPSO with arbitrarily large populations of robots. All of the experiments were carried out in a simulated scenario of 300 × 300 m with obstacles randomly deployed at each trial, in which three 2-dimensional benchmark cost functions $F_\epsilon(x, y)$, $\epsilon = 1, 2, 3$, were defined where x and y-axis represent the planar coordinates in meters (Fig. 10): i) Gaussian ($\epsilon = 1$); ii) Rastrigin ($\epsilon = 2$); and iii) Rosenbrock ($\epsilon = 3$) [38].

In order to improve the interpretation of the algorithm performance, results were once again normalized in a way that the objective of robotic teams was to maximize $f_\epsilon(x, y)$, i.e., minimize the original benchmark functions $F_\epsilon(x, y)$, thus finding the optimal solution of $f_\epsilon(x, y) = 1$, while avoiding obstacles and ensuring the MANET connectivity:

$$f_\epsilon(x, y) = \frac{F_\epsilon(x, y) - \max F_\epsilon(x, y)}{\min F_\epsilon(x, y) - \max F_\epsilon(x, y)}. \quad (25)$$

Test groups of 100 trials of 500 iterations each were considered for $N_T = 25, 50, 100$ robots. Also, a minimum, initial and maximum number of 2, 5 and 8 swarms were used. The maximum travelled distance between iterations was set as 0.5 m, i.e.,

$$\max |x_n[t + 1] - x_n[t]| = 0.5$$

while the maximum communication distance between robots was set to $d_{max} = 30$ m. The maximum range was considered to be 30 m since it is inside ZigBee typical range [7] and equivalent to the maximum communication distance in urban environment (i.e., with obstacles) of XBee OEM RF modules from Maxstream used in eSwarBots platforms [23]. The same two sets of parameters S_1 and S_2 presented in Table 2 were chosen.

Figure 11 depicts the performance of the algorithm by changing the total number of robots N_T, the objective function $f_\varepsilon(x, y)$ and the set of parameters S_1 and S_2. Similarly to the experiments with real robots, the median of the best solution in the 100 simulation was taken as a final output for each different condition.

In the Gaussian function $f_1(x, y)$, robots seem to perform well (Fig. 11a–b). The reason may be that $f_1(x, y)$ only presents 2 sub-optimal regions that are far apart from the optimal solution (Fig. 10a).

In the Rastrigin function $f_2(x, y)$ robots also seem to have a good performance (Fig. 11c–d). It is noteworthy that this function presents a difficult problem due to the relation between the size of the search space

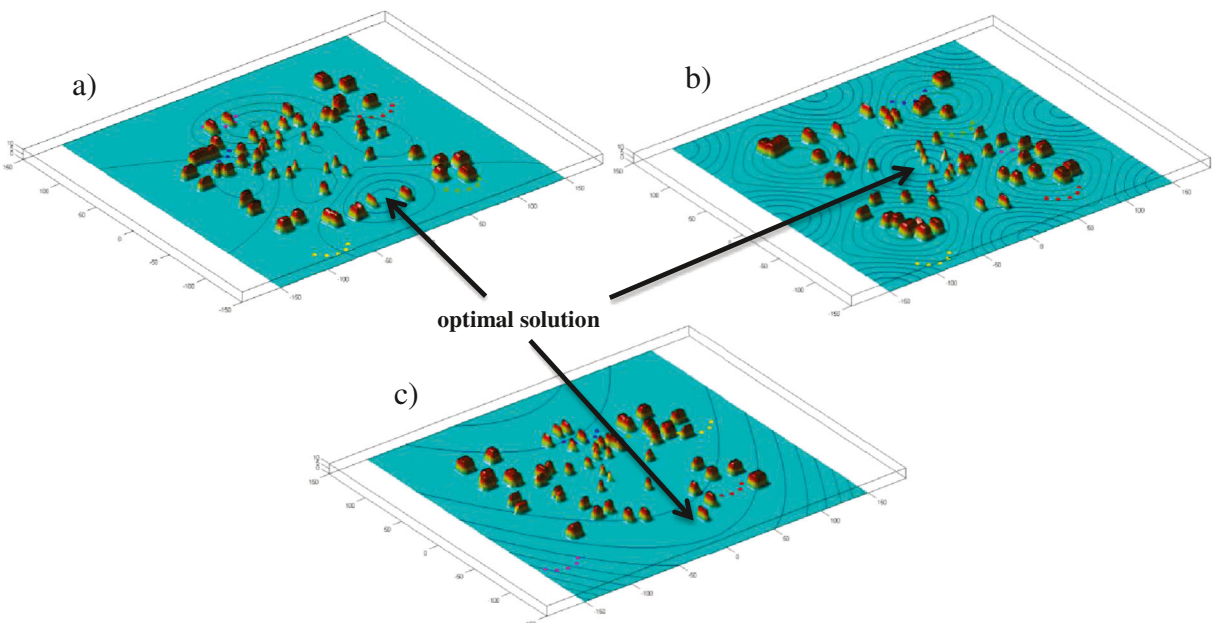

Fig. 10 Virtual scenario with obstacles and 25 robots divided into 5 swarms. **a** Gaussian $f_1(x, y)$; **b** Rastrigin $f_2(x, y)$; and **c** Rosenbrock $f_3(x, y)$ [38]

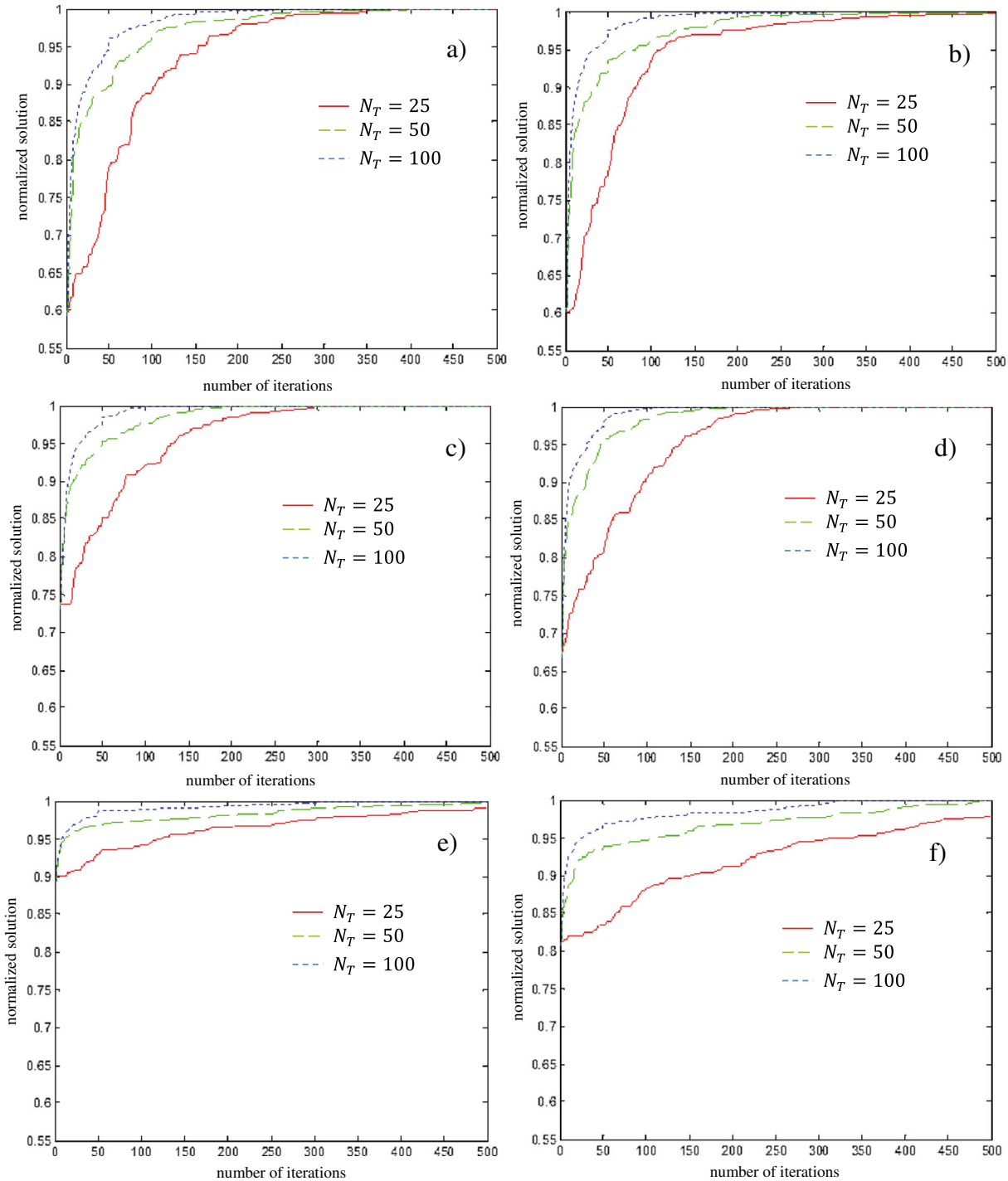

Fig. 11 RDPSO evaluation changing the number of robots N_T for each objective function and set of parameters: **a** $f_1(x, y)$ and S_1; **b** $f_1(x, y)$ and S_2; **c** $f_2(x, y)$ and S_1; **d** $f_2(x, y)$ and S_2; **e** $f_3(x, y)$ and S_1; and **f** $f_3(x, y)$ and S_2

and the number of sub-optimal solutions (Fig. 10b). In fact, when attempting to solve the Rastrigin function, most optimization or foraging algorithms easily fall into sub-optimal solutions. However, as the RDPSO is capable of maintaining a large diversity, it returns better results than expected.

Finally, in the Rosenbrock function $f_3(x, y)$, the optimal solution is inside a long, narrow, parabolic shaped flat valley (Fig. 10c). Robots are able to easily discover the valley (values above 0.9) but they seem to have some minor problems in finding the optimal solution (Fig. 11e–f). Contrarily to the previous functions (i.e., $f_1(x, y)$ and $f_2(x, y)$) in which the target distribution is very common, in foraging tasks requiring of robots to find confined target locations (e.g., toxic waste cleanup), the Rosenbrock function $f_3(x, y)$ is more like olfactory-based swarming. In olfactory-based swarming a plume is subject to diffusion and airflow, which increases the difficulty in finding the initial source (e.g., detection of hazardous gases) [39]. Nevertheless, independent of of the target distribution, the median value of the solution was always greater than 0.95 (i.e., optimal solution vicinities) regardless of the number of robots.

It should also be noted that the second set (S_2) presents worse results than the first one (S_1) in most situations (e.g., Fig. 11e–f). In other words, a greedy behavior wherein robots give too much importance on finding the optimal solution may jeopardize the performance of the team since some collisions or communication ruptures may delay or even interfere with the collective intelligence.

This phenomenon may be explained due to the prioritization of the robot's objectives. Although robotic teams are designed for specific applications (e.g., find a gas leak), the requirements to fulfill such applications (e.g., ensuring MANET connectivity) need to be ensured for collective cooperation to emerge. Nevertheless, it is impossible to withdraw more specific conclusions about the influence of using a different parameterization and/or population on the algorithm performance by only looking at the median of the best solution over time. Therefore, a more exhaustive statistical analysis needs to be carried out so as to assist the design of robotics network dynamic partitioning algorithms for similar scenarios.

A previous study [40] presented a statistical analysis of the RDPSO using the Multivariate Analysis of Variance technique (MANOVA) [41, 42] in order to evaluate the relationship between the population of robots and their maximum communication distance. Similarly, we herein present a MANOVA analysis to further understand the impact of using a conservative and greedy behavior (i.e., set of parameters) while increasing the number of robots within the population.

Therefore, the significance of the set of parameters and the number of robots (i.e., independent variables) to the final solution and the runtime (i.e., dependent variables) was analyzed using a two-way MANOVA for each target distribution (i.e., objective function) after checking the assumptions of multivariate normality and homogeneity of variance/covariance. The assumption of normality for each of the univariate dependent variables was examined using univariate tests of Kolmogorov-Smirnov ($p - value < 0.05$). The univariate normality of each dependent variable in the three objective functions has not been verified. However, since the number of samples is over 30, this statement was assumed[42, 43] using the Central Limit Theorem [41, 42]. Consequently, the assumption of multivariate normality was validated [41, 42].

The assumption about the homogeneity of variance/covariance matrix in each group was examined with the Box's M Test in the three objective functions. Although the homogeneity of variance/covariance matrices has not been verified for the three objective functions (i.e., $p - value = 0.001$), the MANOVA technique is robust to this violation because all the samples have the same size [41, 42]. When the MANOVA detected significant statistical differences, we proceeded to the ANOVA for each dependent variable followed by the Tukey's HSD Post Hoc. The estimation of the effect size (i.e., the proportion of the variance in the dependent variables that can be explained by the independent variables) was done according to Maroco [41] and Pallant [42]. This analysis was performed using IBM SPSS Statistics for a significance level of 5 %. As Table 3 depicts, the MANOVA revealed that the number of robots had a medium effect and significant on the multivariate composite independently on the objective function (with $p - value = 0.001$ and $Power = 1.000$). This indicates that the population of robots, as expected, has a crucial influence in the RDPSO performance.

Table 4 shows that the set of parameters had a small effect, yet significant, on the multivariate composite except for the Gaussian function ($f_1(x, y)$). In

Table 3 Multivariate test for the number of robots

	$f_1(x, y)$	$f_2(x, y)$	$f_3(x, y)$
Pillai's Trace	0.139	0.201	0.203
Partial Eta Squared η_p^2	0.069	0.101	0.117

Table 4 Multivariate test for the set of parameters

	$f_1(x, y)$	$f_2(x, y)$	$f_3(x, y)$
Pillai's Trace	0.006	0.015	0.045
Partial Eta Squared η_p^2	0.006	0.015	0.045
p-value	0.192	0.011	0.001
Power	0.350	0.800	0.998

this last one, the different set of parameters does not present any statistically significant differences. Hence, the proposed set of parameters seems to have a minor influence over the RDPSO performance. However, at this point, it is still not clear if it has a positive or negative influence.

Finally, the interaction between the two independent variables only had a small effect, yet significant, on the multivariate composite in Rosenbrock function ($f_3(x, y)$) (Pillai's Trace $= 0.032$; $\eta_p^2 = 0.016$; $p - value = 0.001$; $Power = 0.957$).

After observing the multivariate significance in the number of robots and set of parameters, an univariate ANOVA for each dependent variable followed by the Tukey's HSD Test was carried out (Fig. 12). It can be concluded that, in general, increasing the number of robots from 50 to 100, does not significantly improve the final solution of the RDPSO for such applications and target distribution. In other words, the algorithm is able to find the optimal solution with 50 robots in most situations. However, the runtime improves significantly as the number of robots increases (Fig. 12d–f). There is an almost inverse linear relationship between the runtime and the number of robots.

As for the set of parameters, the performance of the RDPSO, for both final solution and runtime, does not follow any tendency. For instance, a greedy behavior (S_2) decreases the runtime for functions $f_1(x, y)$ and $f_2(x, y)$ and increases it for function $f_3(x, y)$ (Fig. 12d–f). Despite using similar parameter values inside the previously defined particular attraction domain $\mathscr{A}_p$ (Fig. 5), it can be observed that small differences between both sets may result in considerable differences, mainly, in the algorithm convergence rate (i.e., runtime).

On the other hand, in most situations, one can slightly overcome the negative effect of a poor choice of parameters (within $\mathscr{A}_p$), increasing the population of robots. However, for the second set of parameters (S_2), robots happen to collide and sometimes they cannot maintain the maximum communication distance between them. Even within the second set, this could be avoided if parameters were not constant values throughout the search. In other words, there are some situations in which robots should adapt their behavior (e.g., [44]). For instance, if a robot is near collision, the obstacle susceptibility weight ρ_3 should instantaneously increase, hence ignoring the mission and communication constraints.

To go a step further into evaluating the parameterized RDPSO, next section presents simulation experiments so as to experimentally assess and compare its performance with four state-of-the-art algorithms in an exploration task.

5.3 Benchmark on MRSim

The Multi-Robot Simulator (MRSim)[2] was used to compare the RDPSO algorithm with four state-of-the-art swarm techniques, namely, the Extended Particle Swarm Optimization (EPSO) [14, 15], the Physically-embedded Particle Swarm Optimization (PPSO) [17, 46], the Glowworm Swarm Optimization (GSO) [47, 48], and the Aggregations of Foraging Swarm (AFS) [49, 50]. MRSim was initially created to evaluate simulation experiments under search and rescue robotics. As such, it has been successively improved considering several real-world phenomena such as radio frequency (RF) propagation. MRSim is an evolution of the Autonomous mobile robotics toolbox SIMROBOT (SIMulated ROBOTs) previously developed for an obsolete version of MatLab.[3] The simulator was completely remodelled for the newer MatLab version and new features were included, such as mapping and inter-robot communication. Besides that, MRSim also allows adding a monochromatic bitmap as a planar scenario changing its properties (e.g., obstacles, size, among others), adding features of each swarm robotic technique (e.g., robotic population, maximum communication range, among others) and edit robots' model (e.g., maximum velocity, type

[2] http://www.mathworks.com/matlabcentral/fileexchange/38409-mrsim-multi-robot-simulator-v1-0

[3] http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot-en.html

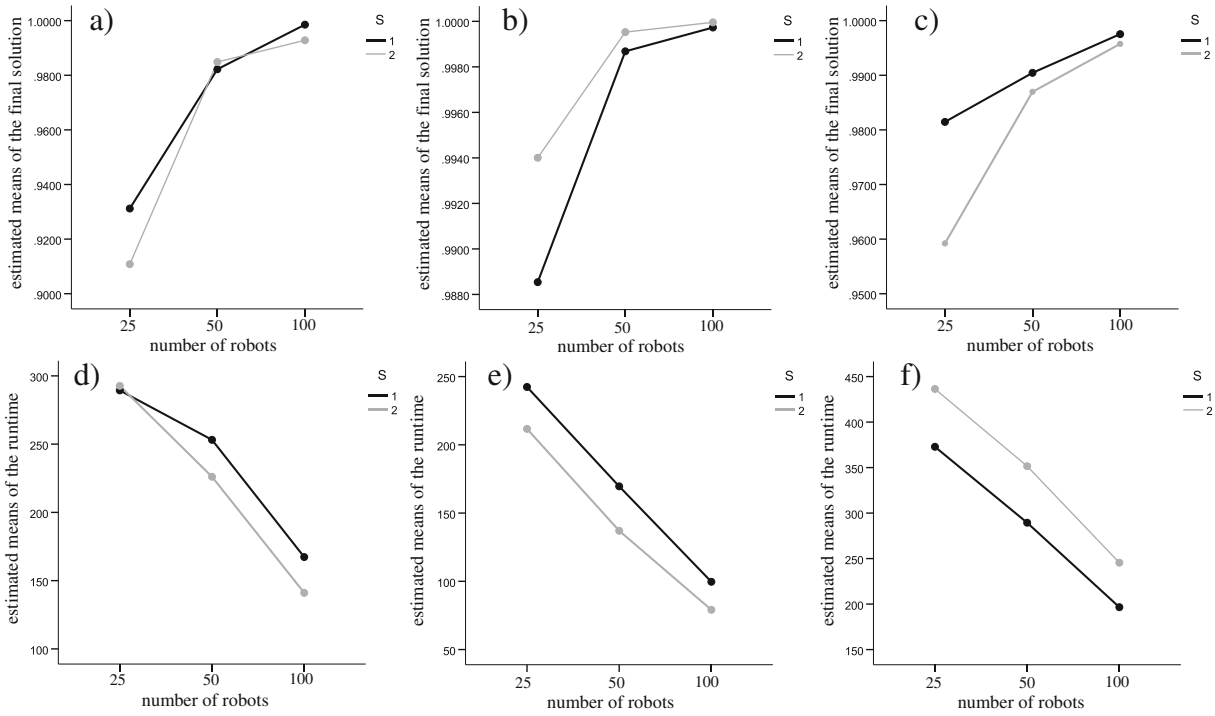

Fig. 12 Estimated marginal means of the RDPSO performance for the: **a** final solution using $f_1(x, y)$; **b** final solution using $f_2(x, y)$; **c** final solution using $f_3(x, y)$; **d** runtime using $f_1(x, y)$; **e** runtime using $f_2(x, y)$; **f** runtime using $f_3(x, y)$

of sensors, among others). This simulator was first evaluated in the context of the CHOPIN project, thus comparing decentralized and centralized versions of both RDPSO for exploration purposes [51].

Figure 13 depicts the MRSim interface with a simulation trial with robots using the RDPSO algorithm to collectively explore the whole scenario of a large basement garage environment - the garage of the Institute of Systems and Robotics at University of Coimbra, with an area of $A = 2975m^2$.

All algorithms were evaluated while changing the number of robots within the population $N_T = 10, 20, 30$ and the maximum communication range $d_{max} = 30, 100\ m$. The communication range was based on common values presented in the literature for both ZigBee and WiFi communication [7]. To significantly test and compare the different algorithms, 30 trials of 500 iterations for each (N_T, d_{max}) combination were conducted. Also, to perform a straightforward comparison between the algorithms, robots were randomly deployed in the scenario presented in Fig. 13. Exploring and building a map of the scenario was used as mission objective to evaluate the five algorithms. Hence, the objective function of the team of robots was defined as a cost function in which robots need to minimize the map's entropy, i.e., the uncertainty about the map. Please refer to Rocha et al. for a more detailed description [52]. Each robot n computes its best frontier cell as:

$$m_i^s = argmax_{(m_i \in \Re(x_n[t], R_w)} \times [\Psi(x_n[t], m_i) \parallel \overrightarrow{\nabla} H(m_i) \parallel], \quad (26)$$

wherein $\Re(x_n[t], R_w)$ represents the set of frontier cells located in the neighbourhood of robot n with sensing radius R_w. The coefficient $\Psi(x_n[t], m_i) \in [0, 1]$ measures if the cell m_i is in line-of-sight from a position $x_n[t]$, which also implies that cell m_i is likely to be empty. Moreover, the entropy of the cell m_i is represented by $H(m_i)$ and may be calculated as:

$$H(m_i) = -p(m_i)log[p(m_i)] - (1 - p(m_i))log_2[1 - p(m_i)], \quad (27)$$

being $p(m_i)$ the probability that a grid cell is occupied. The performance metric used is the exploration ratio of the scenario over time (number of iterations).

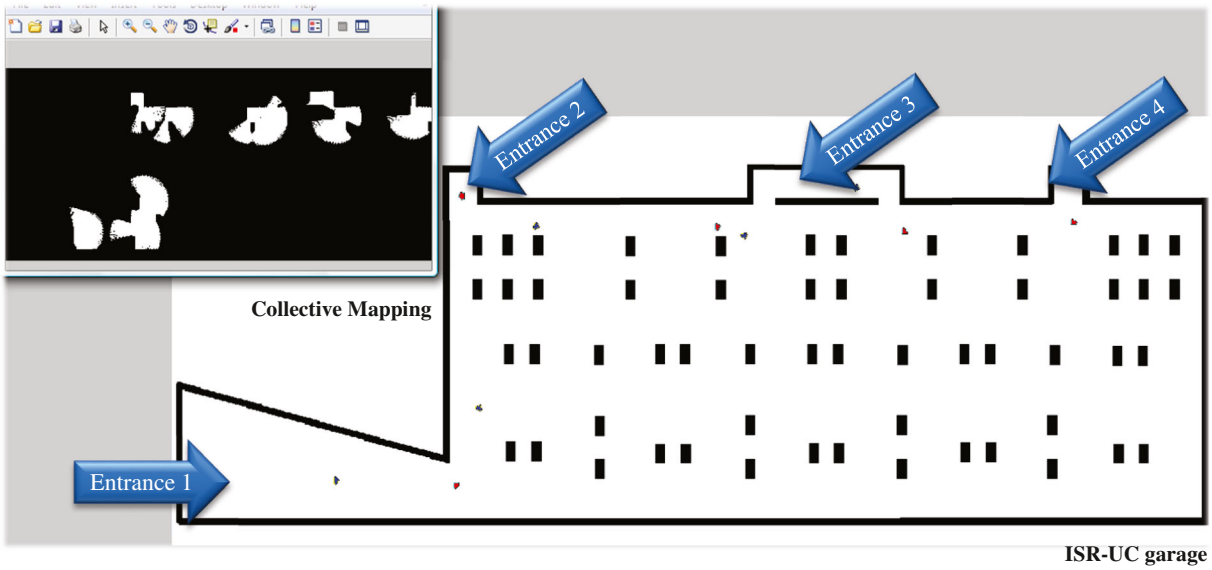

Fig. 13 MatLab Multi-Robot Simulator (MRSim). Illustration of one trial with 10 robots performing the collective mapping of an unknown scenario under the influence of the RDPSO algorithm [51]

The exploration ratio may be obtained by normalizing the mapped scenario as it follows:

$$\eta_{exp}[t] = \frac{A_e[t]}{A_a}, \tag{28}$$

wherein A_a is the useful area of the scenario only considering free cells, while $A_e[t]$ is the scenario explored up to time, or iteration, t.

As Fig. 14 depicts, the median of the best solution over the 500 trials was taken as the final output for each (N_T, d_{max}) combination. As it is possible to observe, the RDPSO outperforms the other methods for all (N_T, d_{max}) configurations tested. Nevertheless, such difference decreases especially as the population of robots increases when compared to the AFS and the GSO. For instance, for the configuration of $(N_T, d_{max}) = (30, 30)$, i.e., Fig. 14e, the GSO presents a better performance than the RDPSO during the first iterations while the AFS closely follows the same performance as the RDPSO.

To facilitate a straightforward comparison and since some of the algorithms present a similar performance, the area under the curve (AUC) may be used. This is a common measure used to analyse the accuracy of receiver operating characteristic (ROC) curves that represent the performance of classifiers. As the exploration ratio $\eta_{exp}[t]$ is a discrete function with $t \in \mathbb{N}_0$, the AUC may be calculated by the sum of each value over the 500 iterations. Moreover, one can normalize the AUC by dividing it by 500, thus resulting in a representation of the probability that a team of robots under a given algorithm has to explore the whole scenario. Hence, the normalized AUC may be calculated as:

$$AUC = \frac{1}{500} \sum_{k=0}^{500} \eta_{exp}[k], \tag{29}$$

The AUC of each set of trials is represented using boxplot charts. As one may observe in Fig. 15, the influence of the population is more significant than the communication range. This should be expected as swarm intelligent algorithms perform well for larger population of robots, i.e., it is possible to observe a higher degree of collective emergent behaviours as the population grows [37]. Nevertheless, it is still possible to observe that, in most methods, an increase in the maximum communication range results in a minor improvement in the exploration ratio accuracy and a significant one in its precision, i.e., smaller interquartile range. In other words, the outcome becomes more predictable and regular as the maximum communication range increases. Regarding the comparison between algorithms, it is possible to observe that both PPSO and EPSO present a similar performance with a probability of successfully exploring the whole scenario of almost 70 % for a population of 30 robots. The same may be observed for both AFS and GSO

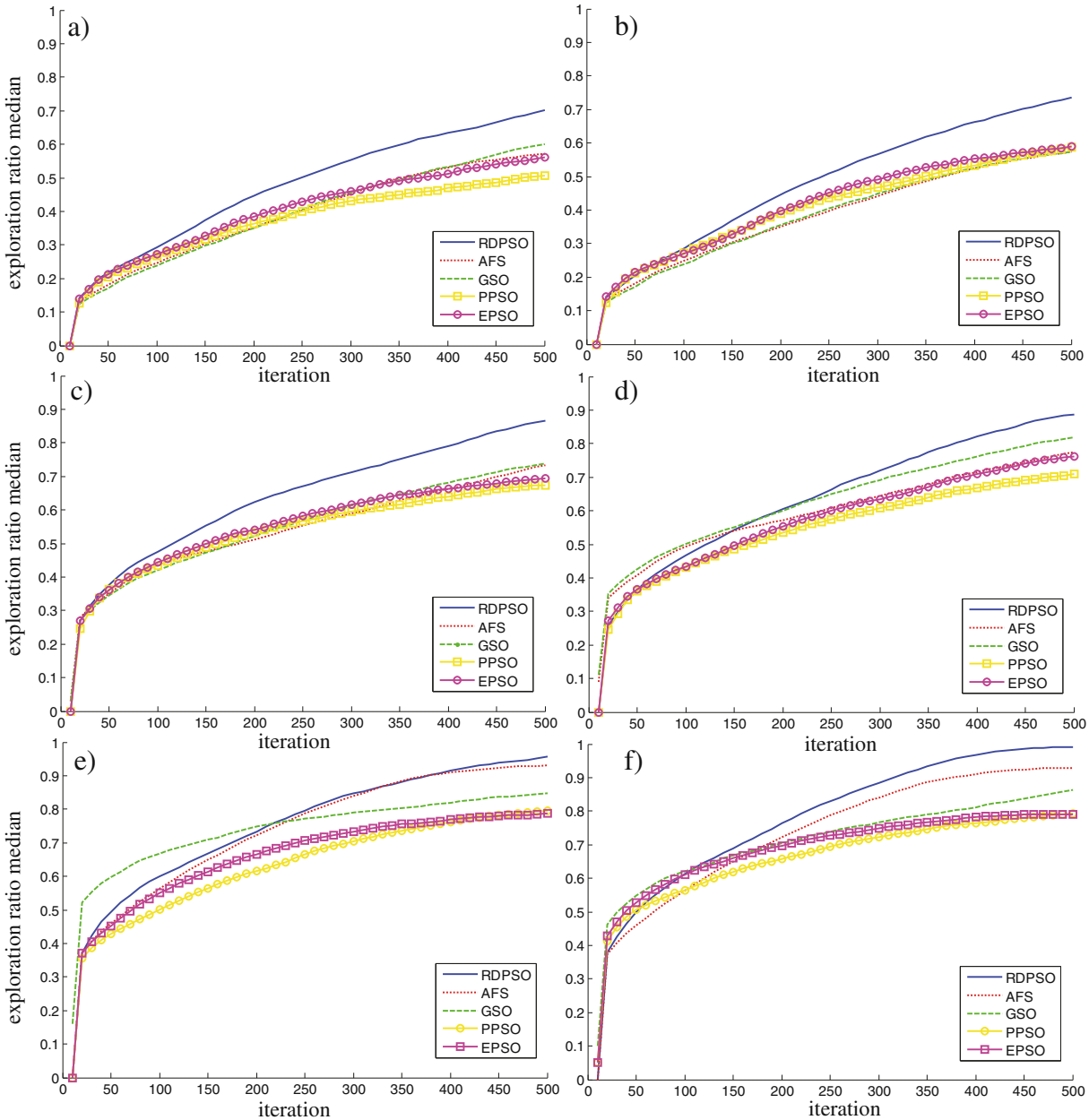

Fig. 14 Median of the exploration ratio $\eta_{exp}[t]$ over the 500 iteration for each method. **a** $(N_T, d_{max}) = (10, 30)$; **b** $(N_T, d_{max}) = (10, 100)$; **c** $(N_T, d_{max}) = (20, 30)$; **d** $(N_T, d_{max}) = (20, 100)$; **e** $(N_T, d_{max}) = (30, 30)$; **f** $(N_T, d_{max}) = (30, 100)$

algorithms, in which a superior performance of almost 75 % may be observed for such population. Finally, the RDPSO outperforms the other methods depicting a probability of successfully exploring the whole scenario of approximately 80 % for the maximum population. This 5 % difference may be generalized for all other (N_T, d_{max}) configurations tested. Nevertheless, such a difference is not linear and although the GSO presents a slightly better performance than the AFS for smaller populations, it seems that the AFS is able to overcome the GSO as the number of robots increases. Also, and as Fig. 15 depicts, the

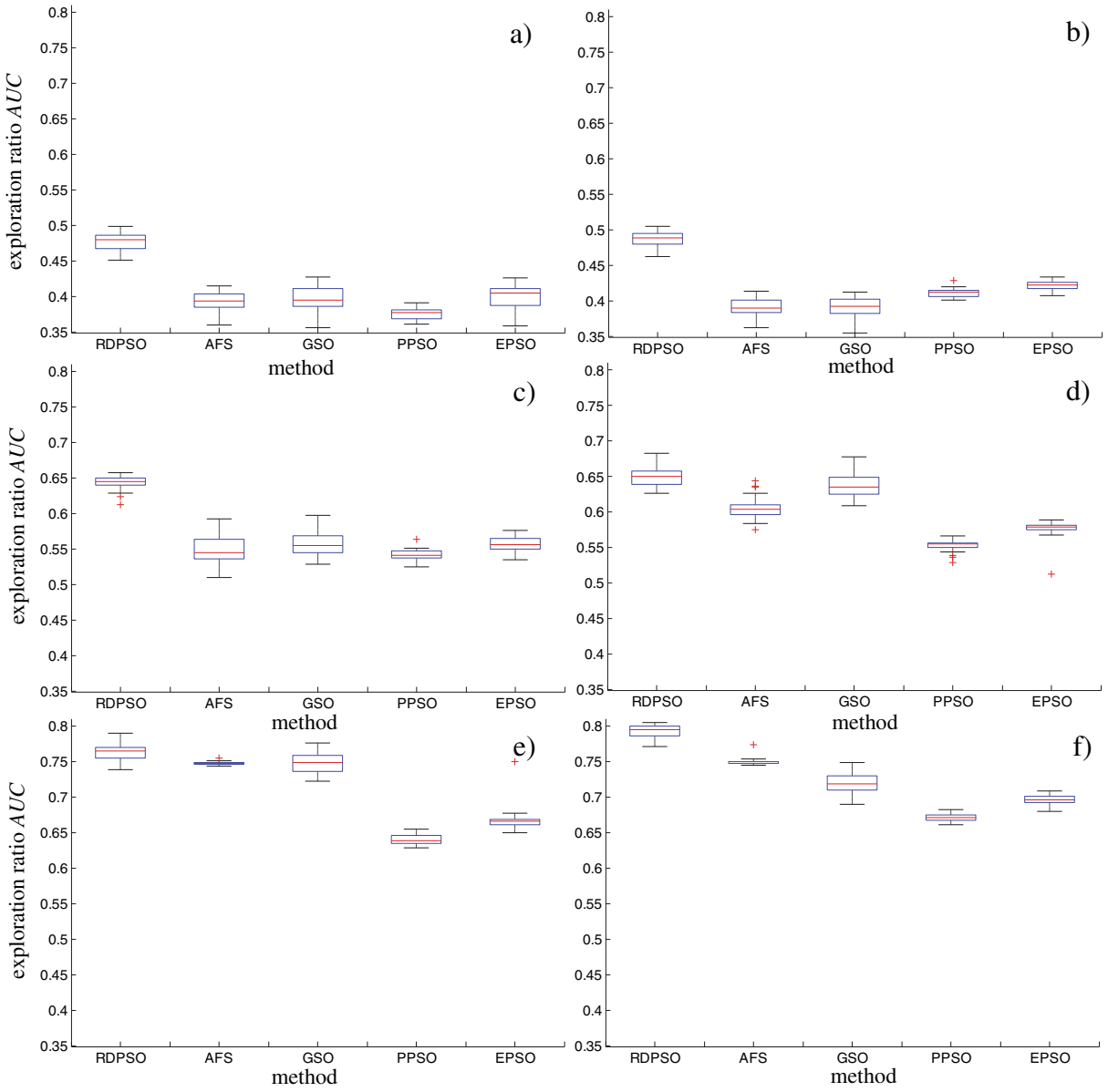

Fig. 15 AUC of the exploration ratio $\eta_{exp}[t]$ over the 500 iteration for each method. **a** $(N_T, d_{max}) = (10, 30)$; **b** $(N_T, d_{max}) = (10, 100)$; **c** $(N_T, d_{max}) = (20, 30)$; **d** $(N_T, d_{max}) = (20, 100)$; **e** $(N_T, d_{max}) = (30, 30)$; **f** $(N_T, d_{max}) = (30, 100)$

AFS presents a similar performance to the RDPSO for larger populations of robots.

6 Discussion

This paper intended to promote the tuning of the RDPSO collective behavior by presenting a rationale behind its parameterization. To that end, Section 4 focused on studying the stability of the RDPSO algorithm so as to define a set of conditions where the convergence of robots toward the solution is guaranteed. By doing so, it was possible to obtain an attraction domain that, for all intents and purposes, simply confines the relationship between the RDPSO parameters to a small region (Fig. 5). This is highly important since it significantly reduces the complexity on settling the RDPSO algorithm, without resorting

to arbitrary parameters that would not ensure its convergence. The results obtained in the previous section allow for the proposal of some guidelines in the process of designing robotics network dynamic partitioning algorithms for similar scenarios. For instance, a more conservative behavior with special attention to obstacle avoidance and communication constraints (S_1) may lead to better results in both terms of performance and runtime as the collective performance highly depends on the information shared between robots (S_2). Regarding the number of robots, it was expected that this would be a crucial variable in designing swarm algorithms. A better performance is achieved in a short amount of time as the number of robots increases. Moreover, a larger population of robots does not significantly disturb the communication network as the RDPSO is endowed with dynamic partitioning properties. However, if the main objective resides in fulfilling the mission regardless on the time needed, then a rationale on the size of the population of robots needs to be carried out. For instance, for the three simulated scenarios of 300 × 300 m, a number of 50 robots proves to be enough regardless of the target distribution and obstacles' location (Section 5.2). Going deeper into the "rabbit hole", a large number of simulation experiments was conducted to study the effect of the number of robots and the communication constraints in the RDPSO, thus comparing it with four state-of-the-art alternatives (Section 5.3). The mission consisted of exploring and mapping a 2000 m^2 scenario in which robots needed to minimize the map's entropy [52]. More than to just state the obvious phenomenon that a larger population of robots improves the overall performance, those experiments were useful to understand the influence of a more constrained communication network on the five swarm algorithms. Through Fig. 15 it was possible to observe a lower variability of the exploration ratio for a larger maximum communication distance feasible between robots, i.e., the outcome became more consistent for a less constrained communication network. Such phenomenon was more perceptible using the EPSO and PPSO algorithms, thus suggesting their higher susceptibility over the communication constraints. Associating this aspect to the fact that both algorithms work on a broadcast communication basis, the authors dissuade the use of those algorithms on applications that may require a larger number of robots (above 20 in the experiments in Section 5.3) or too limited communication constraints (bellow an inter-robot distance of 100 m in the experiments in Section 5.3).

7 Conclusion

The proposed RDPSO algorithm is a sociobiologically inspired parameterized swarm algorithm that takes into account real-world MRS characteristics. This paper presented the convergence analysis of the algorithm, studying its stability in such a way that parameters may be configured within a small attraction domain. Furthermore, an extended convergence analysis based on real robot constraints is conducted, thus decreasing the size of the attraction domain. A swarm of two physical platforms was used to evaluate constraints such as robot dynamics, obstacles and communication. Experimental results show that the algorithm converges in most situations regardless of the number of robots and set of parameters used. Also, the distribution of target locations, i.e., main objective function, does not greatly affect the algorithm performance. However, the algorithm exhibits some minor collisions and communication ruptures between robots when their behavior is greedier as this manifests in an inferior performance in some situations. To further evaluate the herein proposed strategy, this paper outlined an initial benchmark of the RDPSO by comparing it with the outcome from other swarm robotic algorithms under different configurations (e.g., number of robots). Such results can be used to apply swarm robotic concepts to real world applications such as search-and-rescue. One of the future improvements will be to extend the RDPSO with adaptive parameterization since robots may need to change dynamically their behavior during the search missions, based on contextual information. Therefore, by systematically adjusting the parameters within the defined attraction domain, the RDPSO should be extended in order to control the swarm susceptibility to the main mission, obstacle avoidance and communication constraint.

Acknowledgments This work was supported by a PhD scholarship (SFRH/BD /73382/2010) granted to the first author by the Portuguese Foundation for Science and Technology (FCT), the Institute of Systems and Robotics (ISR) and the Institute of Telecommunications (IT-Covilh) also under regular funding by FCT.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
2. Kennedy, J., Eberhart, R.: A new optimizer using particle swarm theory. In: Proceedings of the IEEE 6th International Symposium on Micro Machine and Human Science, pp. 39–43. Nagoya (1999)
3. Tillett, J., Rao, T.M., Sahin, F., Rao, R., Brockport, S.: Darwinian particle swarm optimization. In: Proceedings of the 2nd Indian International Conference on Artificial Intelligence, pp. 1474–1487 (2005)
4. Liu, Y., Nejat, G.: Robotic urban search and rescue: a survey from the control perspective. J. Intell. Robot. Syst. **72**(2), 147–165 (2013)
5. Parker, L.E.: Multiple mobile robot systems. Springer Handbook of Robotics, pp. 921–941 (2008)
6. Couceiro, M.S., Rocha, R.P., Ferreira, N.M.F.: A novel multi-robot exploration approach based on particle swarm optimization algorithms. In: Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011, pp. 327–332. Kyoto (2011)
7. Couceiro, M.S., Rocha, R.P., Ferreira, N.M.F.: Ensuring ad hoc connectivity in distributed search with robotic darwinian swarms. In: Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011, pp. 284–289. Kyoto (2011)
8. Couceiro, M.S., Martins, F.M.L., Rocha, R.P., Ferreira, N.M.F.: Analysis and parameter adjustment of the RDPSO - towards an understanding of robotic network dynamic partitioning based on Darwin's theory. Int. Math. Forum, Hikari Ltd. **7**(32), 1587–1601 (2012)
9. Couceiro, M.S., Ferreira, N.M.F., Machado, J.A.T.: Fractional order darwinian particle swarm optimization. In: Proceedings of the 3th Symposium on Fractional Signals and Systems, FSS'2011, Coimbra (2011)
10. Abd-El-Wahed, W.F., Mousa, A.A., M, A.E.-S.: Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. J. Comput. Appl. Math. **235**(5) (2011)
11. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58-73 (2002)
12. Kadirkamanathan, V., Selvarajah, H., Fleming, P.J.: Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans. Evol. Comput. **10**(3), 245–255 (2006)
13. Rapaic, M.R., Kanovic, Z., Jelicic, Z.D.: A theoretical and empirical analysis of convergence related particle swarm optimization. WSEAS Trans. Syst. Control **4**(11), 541–550 (2009)
14. Pugh, J., Martinoli, A.: Multi-robot learning with particle swarm optimization. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (2006)
15. Pugh, J., Martinoli, A.: Inspiring and modeling multi-robot search with particle swarm optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium (2007)
16. Saikishan, D., Prasanna, K.: Multiple robot co-ordination using particle swarm optimisation and bacteria foraging algorithm. Department of Mechanical Engineering, National Institute of Technology BTech thesis (2010)
17. Hereford, J., Siebold, M.: Multi-robot search using a physically-embedded particle swarm optimization. Int. J. Comput. Intell. Res. **4**(2), 197–209 (2008)
18. Jatmiko, W., Sekiyama, K., Fukuda, T.: A PSO-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement. IEEE Comput. Intell. Mag. **2**(2), 37–51 (2007)
19. Podlubny, I.: Fractional differential equations, 198th edn, vol. 198. Academic, San Diego, California (1999)
20. Couceiro, M.S., Luz, J.M.A., Figueiredo, C.M., Ferreira, N.M.F.: Modeling and Control of Biologically Inspired Flying Robots. Journal of Robotica - Press, Cambridge University (2011)
21. Couceiro, M.S., Martins, F.M.L., Rocha, R.P., Ferreira, N.M.F.: Introducing the fractional order robotic Darwinian PSO. In: Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences - ICNPAA'2012, Vienna (2012)
22. Tenreiro Machado, J.A., Silva, M.F., Barbosa, R.S., Jesus, I.S., Reis, C.M., Marcos, M.G., Galhano, A.F.: Some applications of fractional calculus in engineering. Hindawi Publ. Corp. Math. Probl. Eng. **1**(34), 1-34 (2010)
23. Couceiro, M.S., Figueiredo, C.M., Luz, J.M.A., Ferreira, N.M.F., Rocha, R.P.: A low-cost educational platform for swarm robotics. International Journal of Robots, Education and Art (2011)
24. Williams, R.L., Wu, J.: Dynamic obstacle avoidance for an omnidirectional mobile robot. J. Robot. **2010**(901365), 14 (2010)
25. Magnenat, S., Rétornaz, P., Bonani, M., Longchamp, V., Mondada, F.: ASEBA: a modular architecture for event based control of complex robots. IEEE/ASME Trans. Mechatron. (99), 1-9 (2009)
26. Crispin, Y.J.: Cooperative control of multiple swarms of mobile robots with communication constraints, optimization and cooperative control. In: Hirsch, M. J. et al. (eds.) ch. 381, pp. 207–220, Berlin, Heidelberg: Springer (2009)
27. Tewolde, G.S., Wu, C., Wang, Y., Sheng, W.: Distributed multi-robot work load partition in manufacturing automation. 4th IEEE Conference on Automation Science and Engineering, Key Bridge Marriott, Washington DC (2008)
28. Hahn, H.K., Schoenberger, K.: The ordered distribution of natural numbers on the square root spiral. The Journal of Business **2007**, 1–35 (2007)
29. Kulkarni, R.V., Venayagamoorthy, G.K.: Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. IEEE Trans. Syst., Man Cybernet. (**40**)6 (2010)
30. Burchardt, T.: Social exclusion: concepts and evidence. Breadline Europe: the measurement of poverty (2000)
31. Craighead, J.J., Sumner, J.S., Mitchell, J.A.: The Grizzly Bears of Yellowstone: Their Ecology in the Yellowstone Ecosystem, pp. 1959–1992. Island Press, Washington, D.C. (1995)

32. Zhang, K., Collins, E.G., Barbu, A.: An efficient stochastic clustering auction for heterogeneous robotic collaborative teams. J. Intell. Robot. Syst. **72**(3–4), 541–558 (2013)
33. Elaydi, S.: An Introduction to Difference Equations, 3rd edn, pp. 257–258. Springer, Science and Business Media, Inc., (2005)
34. Barnett, S.: Polynomials and Linear Control Systems. Marcel Dekker Inc., New York (1983)
35. Yasuda, K., Iwasaki, N., Ueno, G., Aiyoshi, E.: Parti1629 cle swarm optimization: a numerical stability analysis and parameter adjustment based on swarm activity. vol. 3, pp. 642-659. EEJ Transactions on Electrical and Electronic Engineering, Wiley InterScience. Wiley InterScience (2008)
36. Wakasa, Y., Tanaka, K., Nishimura, Y.: Control-theoretic analysis of exploitation and exploration of the PSO algorithm. In: IEEE International Symposium on Computer-Aided Control System Design, IEEE Multi-Conference on Systems and Control, Yokohama (2010)
37. Beni, G.: From swarm intelligence to swarm robotics. In: Proceedings of the Swarm Robotics Workshop pp. 1–9, Heidelberg (2004)
38. Molga, M., Smutnicki, C.: Test functions for optimization needs. In: Xin-She Yang (ed) Engineering Optimization: An Introduction with Metaheuristic Applications (2005)
39. Marques, L., Nunes, U., Almeida, A.d.: Particle swarm-based olfactory guided search. Auton. Robot **20**(3), 277–287 (2006)
40. Couceiro, M.S., Ferreira, N.M.F., Rocha, R.P., Martins, F.M.L., Clemente, F.: Statistical significance analysis of the R-DPSO - towards an understanding of the relationship between the population of robots and the maximum communication distance. In: Proceedings of the International Symposium on Computational Intelligence for Engineering Systems (ISCIES'2011), Coimbra (2011)
41. Maroco, J.: Análise Estatística com utilização do SPSS. Lisboa: Edições Silabo (2010)
42. Pallant, J.: SPSS survival manual, kindle edition ed., 4th edn. Open University Press (2011)
43. Pedrosa, A.C., Gama, S.M.A.: Introdução Computacional à Probabilidade e Estatística. Porto Editora, Portugal (2004)
44. Liu, W., Winfield, A.F.T.: Modeling and optimization of adaptive foraging in swarm robotic systems. Int. J. Robot. Res. IJRR **29**(14), 1743–1760 (2010)
45. Pugh, J., Martinoli, A.: Inspiring and modeling multi-robot search with particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium. Honolulu, HI (2007)
46. Hereford, J.M., Siebold, M.A.: Bio-inspired search strategies for robot swarms. In: Swarm Robotics, From Biology to Robotics. pp. 1–27 (2010)
47. Krishnanand, K.N., Ghose, D.: A glowworm swarm optimization based multi-robot system for signal source localization. In: Design and Control of Intelligent Robotic Systems - Studies in Computational Intelligence, pp. 49–68 (2009)
48. Krishnanand, K.N., Ghose, D.: Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell. **3**(2), 87–124 (2009)
49. Gazi, V., Passino, K.M.: Stability analysis of swarms. IEEE Trans. Autom. Control **48**(4), 692–697 (2003)
50. Gazi, V., Passino, K.M.: Stability analysis of social foraging swarms. IEEE Trans. Syst. Man Cybernet. **34**(1), 539–557 (2004)
51. Couceiro, M.S., Portugal, D., Rocha, R.P.: A collective robotic architecture in search and rescue scenarios. In: Proceedings of the 28th Symposium On Applied Computing (SAC2013). Coimbra (2013)
52. Rocha, R.P., Ferreira, F., Dias, J.: Multi-robot complete exploration using hill climbing and topological recovery. In: Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2008), Nice (2008)