Robotica (2015) volume 33, pp. 157–180. © Cambridge University Press 2014
doi:10.1017/S0263574714000058

Understanding the communication complexity of the robotic Darwinian PSO

Micael S. Couceiro†,‡*, Amadeu Fernandes†,
Rui P. Rocha† and Nuno M. F. Ferreira‡

†*Institute of Systems and Robotics (ISR), Department of Electrical and Computer Engineering, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal*
‡*RoboCorp, Engineering Institute of Coimbra, Quinta da Nora, 3030-199 Coimbra, Portugal*

(Accepted December 24, 2013. First published online: February 13, 2014)

SUMMARY

An extension of the well-known *Particle Swarm Optimization* (PSO) to multi-robot applications has been recently proposed and denoted as *Robotic Darwinian PSO* (RDPSO), benefited from the dynamical partitioning of the whole population of robots. Although such strategy allows decreasing the amount of required information exchange among robots, a further analysis on the communication complexity of the RDPSO needs to be carried out so as to evaluate the scalability of the algorithm. Moreover, a further study on the most adequate multi-hop routing protocol should be conducted. Therefore, this paper starts by analyzing the architecture and characteristics of the RDPSO communication system, thus describing the dynamics of the communication data packet structure shared between teammates. Such procedure will be the first step to achieving a more scalable implementation of RDPSO by optimizing the communication procedure between robots. Second, an ad hoc on-demand distance vector reactive routing protocol is extended based on the RDPSO concepts, so as to reduce the communication overhead within swarms of robots. Experimental results with teams of 15 real robots and 60 simulated robots show that the proposed methodology significantly reduces the communication overhead, thus improving the scalability and applicability of the RDPSO algorithm.

KEYWORDS: Distributed search; Swarm robotics; Scalability; MANET; Communication complexity; Routing protocol.

1. Introduction

Communication constitutes one of the most important resources for more effective cooperation among robots and improved robust collective performance.[1] The way robots communicate can be basically divided into the following three most common techniques:

- *Implicit communication* "through the world" (i.e., stigmergy) – robots sense the effects of teammates' actions through their effects on the world.[2–5]
- *Passive action recognition* – robots use sensors to directly observe the actions of their teammates.[6]
- *Explicit (intentional) communication* – robots directly and intentionally communicate relevant information through some active means (e.g., *WiFi*).[7,8]

Within the three techniques described above, the use of explicit communication is the most appealing method because of its directness and ease with which robots can become aware of the actions and/or goals of their teammates. The main uses of explicit communication in multi-robot teams are to synchronize actions, exchange information, and to negotiate between robots. Furthermore, explicit communication is a way of dealing with the hidden state problem, in which limited sensors cannot distinguish between different states of the world that are important for task performance.

* Corresponding author. E-mail: micaelcouceiro@gmail.com

However, the development of robot teams for unstructured scenarios, such as rescue missions, require that robots are able to maintain communication among them without the aid of a communication infrastructure. In other words, robots need to be able to deploy and maintain a *Mobile Ad hoc NETwork* (MANET) in order to explicitly exchange information within multi-hop network paths without unnecessarily restricting the team's range.[9]

MANETs have attracted much attention in the last years within mobile robotics community. The efficient information shared between agents belonging to a *multi-robot system* (MRS) would allow the coordination and cooperation necessary to fulfill collective tasks such as search and rescue (SaR). However, such networks typically consist of a large number of distributed nodes (i.e., robots) that organize themselves into multi-hop wireless networks. Therefore, robots may cooperate and route messages for each other,[9] i.e., robots can perform the roles of both hosts and routers.

Usually, within the context of MRS, a node corresponds to a robot with embedded processor, low-power radio, and is typically battery-operated. In order for MANETs to be cost efficient, the onboard processing, the wireless communication capabilities, and the battery power of each robot are highly limited. Moreover, since robots have mobility nature, the topology of the distributed networks is time varying and the strength of the connection can rapidly change or even completely disappear.

1.1. Our preliminary works

It was based on the assumptions that the authors proposed a strategy in Couceiro *et al.*[9] to ensure the MANET connectivity based on attraction–repulsion mechanisms evaluated on the *Robotic Darwinian Particle Swarm Optimization* (RDPSO) (cf., Section 2 for a brief explanation about the RDPSO algorithm). The problem was stated as having a population of N robots, divided into several swarms of N_S robots, $s \in \mathbb{N}$, where each robot would be both an exploring agent of the environment and a mobile node of a MANET that performs packet forwarding, according to a paradigm of multi-hop communication. The goal was to ensure that robots would explore an unknown environment, while ensuring that the MANET regarding their swarm would remain connected throughout the mission. For this purpose, the connectivity between robots was described by means of a *link matrix* $L = \{l_{ij}\}$ for a N_S-node network, where each entry represents the link between robots i and j. The link was defined as either the *link distance*[10] or the *link quality*[11] (e.g., *Received Signal Strength Indicator* (RSSI)) between pairs of robots. Simulation results showed that the influence inherent to communication's limitations can be attenuated by increasing the number of robots or the communication range/quality.

Recently, the authors proposed a natural extension of Couceiro *et al.*,[9] focusing on a fault-tolerance strategy to guarantee $k-$connected MANETs within each swarm, $k \epsilon \mathbb{N}$ and $k \leq N_S - 1$.[12] Hence, a given robot would choose its k-nearest neighbors, and the virtual force to maintain the MANET connectivity was represented by the vector sum of k-virtual forces. A population of 15 physical robots, denoted as *eSwarBots*,[13] was used to evaluate such strategy. Experimental results showed that the proposed fault-tolerance RDPSO would enable the overcoming of several robot failures such as energy depletion. This work follows the same principles that have been addressed previously in other works, such as Sabattini *et al.*[14] and Casteigts *et al.*,[15] regarding the need to maintain a pervasive MANET.

Nevertheless, only by securing that each robot may communicate with its teammates does not ensure an efficient group communication. Besides studying the necessary information to be exchanged between teammates, routing protocols should be designed based on the mission-related contextual information, i.e., based on the behavior that one should expect from the MRS.

1.2. Prior works

Bearing in mind such assumptions, many works on MRS have been focused on efficiently sharing information between teammates.[16–18] Rocha[16] addressed the problem of building volumetric maps efficiently sharing the necessary information based on mutual information minimization. To that end, the author presented a distributed architecture model with efficient information sharing, and wherein entropy was used to define a formal information-theoretic background to reason about the mapping and exploration process. This allows sharing only information that may be relevant for the team. It was with the same principle that Hereford and Siebold[17] proposed a swarm exploration strategy wherein robots only shared their position if their own solution was the best solution in the whole

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

swarm. Although this is an interesting strategy, robots still need to share information concerning their own solution and a global assessment of the collective performance needs to be carried out. Similarly, Shah and Meng[18] proposed a communication-efficient dynamic task scheduling algorithm for MRS. This algorithm avoided unnecessary communication by broadcasting global information only to the robots interested in it, thus reducing the communication overhead. Simulation experiments showed that the proposed strategy was able to reduce the communication cost to almost half when compared with a common broadcast approach.

Besides exploiting the necessary information that robots should share, routing protocols, such as the well-known *Ad hoc On-demand Distance Vector* (AODV), have been successively extended based on the mobile network requirements.[19–21] For instance, Abedi *et al.*[19] extended the AODV routing protocol based on the Manhattan mobility model, thus making it more fitted for *Vehicular Ad hoc NETwork* (VANET) applications. Such strategy allowed the establishing of more stable routes, especially in applications demanding a high mobility of nodes, thus reducing the communication overhead of the network. More generally, Asenov and Hnatyshin[20] extended the AODV based on the geographical position of nodes retrieved with Global Positioning Systems (GPS). This improves the performance of the *route discovery* process in AODV routing (cf., Section 4.1 for a description about this mechanism). Nevertheless, such strategy assumes that each robot in the network is aware of all teammates' position, thus increasing the communication complexity. Similarly, Ayash *et al.*[21] proposed two GPS-based strategies, namely the AODV Location Aided Routing (LAR) protocol and the AODV*Line* protocol, to minimize the control overhead of the AODV protocol, thus limiting the flooding area of AODV. While the first protocol limits the route discovery to a small area of the network, the second one uses node location information to restrict route search area to be only near the line connecting source and destination nodes. However, both strategies still present the same disadvantage as in Asenov and Hnatyshin[20], i.e., the knowledge about the current position of all robots.

1.3. Statement of contributions and paper outline

Although this work revolves around the RDPSO first presented in Couceiro *et al.*[22] and briefly described in Section 2, the same analysis may be conducted to other behavior-based architectures. Following are the main contributions of this work:

(i) The data exchanged between robots of the same swarm, i.e., network, are studied in depth and a rationale is presented for each different situation within the RDPSO context so as to minimize the communication overhead (Section 3).
(ii) The traditional AODV reactive routing protocols are extended based on the RDPSO dynamics to minimize the number of updates regarding the routes connecting pairs of robots, thus avoiding unnecessary flooding (Section 4).
(iii) Based on the proposed approaches, the communication complexity of the RDPSO is evaluated using both physical and virtual robots in a large indoor environment (Section 5).

Sections 6 and 7 outline the discussion and main conclusions respectively.

2. Robotic Darwinian PSO

This section briefly presents the RDPSO algorithm proposed in Couceiro *et al.*[22] The Darwinian PSO (DPSO) was originally presented by Tillett *et al.*[23] for optimization problems, being an evolutionary algorithm that extends the well-known PSO[24] using natural selection, or survival-of-the-fittest, to enhance the ability to escape from sub-optimal solutions. The RDPSO is an extension of the DPSO to multi-robot applications presented for the first time in Couceiro *et al.*[22] and further improved in several subsequent publications such as Couceiro *et al.*,[25,26] thus presenting the following features:

- *Social exclusion and inclusion*: The RDPSO is represented by *multiple swarms* (i.e., group of robots from the same network) wherein each swarm individually performs just as a PSO-like robotic algorithm in search of the solution, and some rules govern the whole population of robots. The socially excluded robots randomly wander in the scenario instead of searching for the objective

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

function's global optimum as the other robots in the active swarms do. However, they are always aware of their individual solution and the global solution of the socially excluded group.

- *Obstacle avoidance*: A new cost or fitness function is defined in such a way that it guides the robot in performing the main mission while avoiding obstacles. For this purpose it is assumed that each robot is equipped with sensors capable of sensing the environment for obstacle detection within a finite sensing radius r_s. A monotonic and positive *sensing function*, $g\left(x_n\left[t\right]\right)$, at each discrete time, or iteration, $t \in \mathbb{N}$, is defined. This function depends on the sensing information, i.e., distance from the robot to an obstacle.
- *Ensuring MANET k-connectivity* : Robots' position needs to be controlled in order to maintain the communication based on constraints such as maximum distance and minimum signal quality. The way to preserve the network connectivity depends on the characteristics of the communication. 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. Considering that nodes are mobile, it is necessary to guarantee the communication between all nodes. The robots' position is updated by means of the ensuring MANET connectivity algorithm first presented in ref. [9], and further extended in ref. [12], to consider k-fault tolerance, i.e., each pair of robots from the same swarm is connected to, at least, k robot-disjoint paths.

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

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

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

where coefficients ρ_i, $i = 1, 2, 3, 4$, assign weights to the local best (i.e., cognitive component), the global best (i.e., social component), the obstacle avoidance component, and the network connectedness enforcement component when determining the new velocity, with $\rho_i > 0$. Parameters r_i are random vectors, wherein each component is generally a uniform random number between 0 and 1. $v_n\left[t\right]$ and $x_n\left[t\right]$ represent the velocity and the position vector of robot n respectively. $\chi_i\left[t\right]$ represents the best position of the cognitive, social, obstacle, and MANET components. The cognitive $\chi_1\left[t\right]$ and social components $\chi_2\left[t\right]$ are commonly presented in the classical PSO algorithm. $\chi_1\left[t\right]$ represents the local best position of robot n, while $\chi_2\left[t\right]$ represents the global best position of robot n. The other features $\chi_3\left[t\right]$ and $\chi_4\left[t\right]$ are novel and inherent to multi-robot applications. In brief, $\chi_3\left[t\right]$ represents the local best position of robot n regarding the sensed obstacles so far. Similarly, $\chi_4\left[t\right]$ represents the local best position of robot n that allows maintaining a connected MANET based on its closest neighbor, i.e., one-hop robot.

In the common PSO algorithm, the inertial component $w_n\left[t\right]$ is usually proportional to the inertial influence. The RDPSO uses *fractional calculus* (FC),[27,28] to describe the dynamic phenomenon of a robot's trajectory. As presented in Couceiro *et al.*,[25] the inertial component $w_n\left[t\right]$ may be defined as:

$$w_n\left[t\right] = \alpha v_n\left[t\right] + \frac{1}{2}\alpha v_n\left[t-1\right] + \frac{1}{6}\alpha\left(1-\alpha\right) v_n\left[t-2\right] + \frac{1}{24}\alpha\left(1-\alpha\right)\left(2-\alpha\right) v_n\left[t-3\right], \tag{3}$$

for the *eSwarBot* platforms,[13] where α represents the fractional coefficient.

Considering Eqs. (1)–(3), 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 the other requirements (i.e., avoid obstacles and maintain a certain distance between neighbors). In other words, robots within the same swarm do not physically converge to a given solution but instead reach a global consensus. Such consensus is related to the nature of the mission. For instance, if we have a group of mobile olfactory robots that are trying to find a gas leak in an indoor environment (cf., refs. [29 and 30]), each robot's state comprises its pose and the corresponding value of gas density. The swarm will reach a consensus every time the highest gas density is shared among teammates,

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

Table I. Punish–reward RDPSO rules.

Punish	Reward
If a swarm does not improve during a specific threshold, then the swarm is punished by excluding the worst performing robot.	If a swarm improves and its current number of robots is inferior to the maximum number of accepted robots to form a swarm, then it has a small probability of being rewarded with the best performing robot that was previously excluded.
If the number of robots in a swarm falls below the minimum number of accepted robots to form a swarm, then swarm is punished by being dismantled	If a swarm has been more often rewarded than punished, then it has a small probability of spawning a new swarm.

Header bit [0, 1]		Data byte(s)
0	Local Broadcast to neighbors	Number of bytes depends on specific data
1	Broadcast to whole swarm	

Fig. 1. General communication packet structure for a swarm of N_s robots.

thus affecting their local decision-making. To avoid swarms' stagnation, the RDPSO encompasses the rules presented in Table I, which are based on the principles of social exclusion and inclusion.

Nevertheless, to achieve a global consensus within each swarm, robots need to share a certain amount of information as described in the following section.

3. Sharing Information Within the RDPSO

It has generally been assumed in MRS that each robot has the ability to communicate with any other robot with small consideration for the quality and performance of the wireless communication network. Although being valid in particular situations, such an assumption does not generally hold. As previously described, the RDPSO ensures the connectivity of the network (cf., $\chi_4\,[t]$ term in the previous section and Couceiro *et al.*[12] for a more detailed description). Nevertheless, how this is carried out in practice without overloading the communication channel needs to be addressed. Moreover, the communication packet structure shared between robots needs to be specified and a rationale behind it should be introduced. Generally, the packet data structure may be illustrated as presented in Fig. 1.

It is noteworthy that the broadcast to the whole swarm should be avoided as it represents a high communication complexity. In brief, in order to broadcast to the whole swarm by multi-hop communication, the message needs to be addressed to each robot identity (ID). The number of bytes necessary for the main message, i.e., data byte(s), will depend on the message itself. For instance, if a robot wants to share its position and considering a planar scenario, then two bytes may be enough to represent the coordinate on each axis.

3.1. Ensuring connectivity

Since robots may move apart to further areas, it is important to have a pervasive networking environment for communication among robots. Furthermore, without a preexistent infrastructure, robots need to be able to act as intermediate nodes, i.e., routers, in order to relay information from one point to another, thus supporting multi-hop communication in a MANET.[31]

In a previous work, an initial deployment strategy denoted as *Extended Spiral of Theodorus* (EST) was presented.[12] The EST was introduced as an autonomous, realistic, and fault-tolerant initial deployment strategy based on the RSSI signal. Similar to Rybsky *et al.*'s work,[32] the initial deployment of robots was carried out hierarchically dividing the population of robots into *rangers* and *scouts*. Each ranger handled the initial deployment of an entire swarm of scouts allowing a distributed and autonomous transportation, thus sparing the need of a preprocessing procedure (e.g., topological features extraction using unmanned aerial vehicles). In other words, the initial deployment was able

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

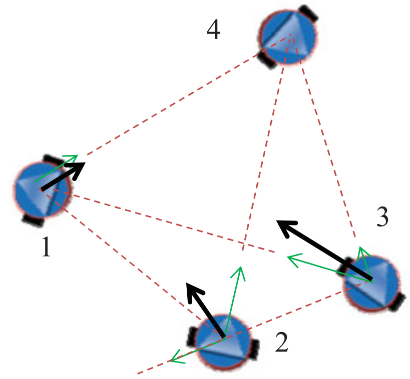


Fig. 2. (Colour online) Illustration of a MANET topology of a swarm. Dashed lines represent the link quality between the pairs of robots, thinner arrows represent the force vectors regarding each chosen neighbor, and larger arrows represent the resulting force vectors that ensure MANET biconnectivity.

Header bit	Data byte(s)
0	$x_n[t]$

Fig. 3. Communication packet structure that allows robots in maintaining the MANET k-connectivity within their swarm of N_s robots.

to ensure that each exploring robot would be able to communicate with k neighbors from the same swarm, $k \in \mathbb{N}$, thus ensuring that the MANET is k-connected.

After the initial deployment process is concluded, robots explore the environment while ensuring the same k-connectivity of the swarm by defining $\chi_4[t]$ as a set of attractive and repulsive forces.[12] Let us consider the following illustrative example presented in Fig. 2 in which it is necessary to guarantee a biconnected network ($k = 2$). As it is possible to observe, robot 1 chooses robot 2 and 4 as its nearest neighbors since they are the nearest ones or the ones that present the higher signal quality. The link between robot 1 and 2 corresponds to the ideal situation, such that any attractive or repulsive force is necessary. However, robot 4 is too far away from robot 1, thus resulting in an attraction virtual force toward it. Robot 2 chooses robot 3 and 4 as its nearest neighbors since robot 1 has first chosen robot 2. As robot 3 is too close to robot 2, a repulsive force is generated. On the other hand, as robot 4 is too far away from robot 2, an attractive force is generated. The resulting force will then allow robot 2 to move away from robot 3 while getting closer to robot 4. Finally, the two nearest neighbors of robot 3 that do not choose it as their nearest neighbor are robot 1 and 4, which are too far away, thus being affected by attractive forces toward them.

Based on the presented strategy, it is possible to ensure the k-connectivity of the network by simply sharing the position to the k neighbors. Therefore, only taking into consideration the information of the N_b robots within the one-hop path (i.e., neighbors) would allow ensuring the connectivity of the whole swarm. Figure 3 presents the packet structure of communication for this particular situation.

An alternative to broadcasting the position to the N_b neighbors would be the use of strategies to find the teammates' position under their visual range.[33] For instance, if robots are equipped with laser range finders, retro-reflective markers may be used for recognition. To that end, one should ensure that the sensing radius r_s is equal or superior to the maximum distance of neighbors, which depends on the minimum inter-robot signal quality RSSI.

3.2. *Converging to the optimal solution*

As previously presented in Section 2, $\chi_2[t]$ represents the best positions of the social component. Therefore, robots from the same active swarm, i.e., not in the socially excluded group, need to share their best cognitive solution $f_n[t]$ and current position $x_n[t]$ so as to compute the position of the robot that has the best social solution. For instance, if one wishes to find a gas leak, the best performing robot will be the one with the highest solution, i.e., $\max_{n \in N_s} f_n[t]$. Nevertheless, efficiently sharing this information may allow to drastically reduce the communication complexity of the RDPSO.

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

Header bit	Data byte(s)	
1	$x_n[t]$	$f_n[t]$

Fig. 4. Communication packet structure that allows robots from active swarms to cooperatively converge to the solution. This packet is only sent if a robot improves its best cognitive solution.

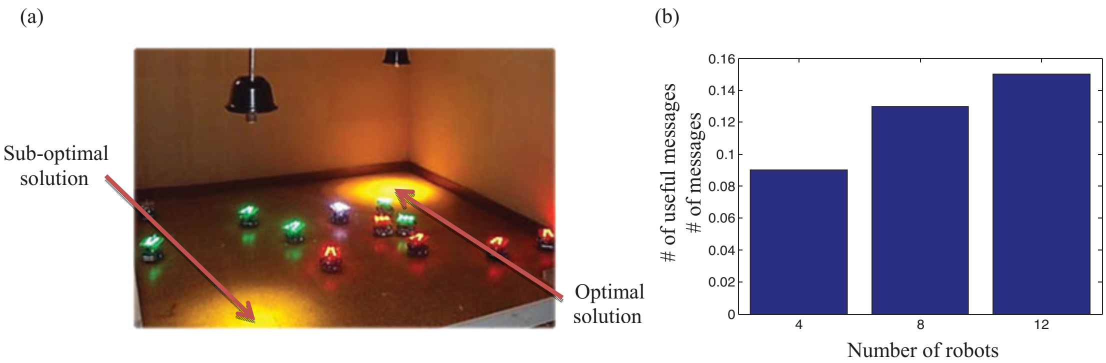


Fig. 5. (Colour online) (a) Experimental setup presented in Couceiro *et al.*[34] (b) Ratio between the number of useful messages and the total number of messages.

For instance, if a robot from the active swarm was unable to improve, then the information about its position and solution is irrelevant to the group, i.e., the collective behavior will not change. Therefore, and as a rule, a robot may only share its current solution and position if it is able to improve its best cognitive solution, i.e., $f_n[t+j] > f_n[t], j \in \mathbb{N}$. Otherwise, as robots are able to memorize the best solution of the swarm and the corresponding position so far, without significantly increasing the memory complexity, robots will simply continue computing their algorithm without communicating. Figure 4 represents the packet structure sent from a robot that was able to improve its solution.

Note that this significantly reduces the communication complexity, as this data need to be exchanged between all teammates, i.e., broadcasted to the whole swarm by means of multi-hop communication. For instance, in a previous work,[34] a setup of 4, 8, and 12 *educative Swarm Robots* (*eSwarBots*)[13] on a small scenario with one optimal and one sub-optimal solution was presented (Fig. 5(a)). As Fig. 5(b) depicts, using 12 robots represents the most critical situation tested regarding the chances that the swarm has to improve. Even so, in a population of 12 robots under 80 trials of 180 sec each, it was possible to observe that a robot is only able to improve in approximately 15% of the iterations, i.e., only approximately 15% of the information shared is useful to the collective performance. As the number of robots decreases for the same scenario, the probability that a robot has to improve also decreases slightly, thus decreasing slightly the amount of useful information (Fig. 5(b)).

It is noteworthy that the amount of useful information will vary depending on several conditions (e.g., number of robots, scenario, mission objectives, among others). Nevertheless, efficiently sharing information based on the strategy proposed here will always significantly reduce the communication complexity of the algorithm as robots will not always improve at each iteration. After this analysis on the data exchanged between robots from active swarms, next section shows an efficient way to share information between excluded robots, i.e., robots within the socially excluded group.

3.3. Avoiding sub-optimality

As presented previously in Section 2, the way the RDPSO handles sub-optimal avoidance is by socially excluding robots that have nothing to offer to the group, i.e., that are unable to improve for a certain stagnancy threshold (cf., ref. [22] and Table I for a more detailed description about this "punish"–"reward" mechanism). In brief, the number of times a swarm evolves without finding an improved objective is tracked with a search counter. If a swarm's search counter exceeds a maximum critical threshold, the swarm is punished by excluding the worst performing robot, which is added to a socially excluded group. Nevertheless, the behavior of the socially excluded robots differs from the ones in the active swarms. Instead of searching for the optimal solution (i.e., the main activity of

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

the society) as the other robots in the active swarms do, they randomly wander in the scenario while avoiding obstacles and maintaining the MANET connectivity with the other excluded robots. Note, however, that they are always aware of their best cognitive solution. That being said, the only regular information that excluded robots need to share is their current position to their neighbors so as to maintain the MANET connectivity (cf., Section 3.1).

However, if an active swarm continues to improve for a certain amount of time, there will be a probability to be rewarded with the best performing robot from the socially excluded group. Moreover, the swarm will also have a small probability of creating a new swarm from the best performing robots from the socially excluded group. Therefore, when excluded robots receive a calling from an active swarm, they will broadcast their best cognitive solutions and respective positions to the whole socially excluded group by means of multi-hop communication (cf., Section 3.2). Thereby they will be able to assess the best performing excluded robots so far and evaluate which ones would be a part of an active swarm.

Although one wishes to avoid broadcasting to the whole multi-hop network, this event will only occur from time to time since it depends on the constant improvement of swarms and a probability of successful calling. Furthermore, an adequate choice on the routing protocol may allow overcoming or, at least, minimizing the broadcast overhead.

4. Routing Protocol

In MANETs, the communication between source and destination nodes may require traversal of multiple hops. Since the introduction of such networks, a community of researchers has proposed a variety of routing algorithms, mainly divided into two classes: (i) proactive; and (ii) reactive. In the first class, every node maintains a list of destinations and their routes by processing periodic topology broadcasts originated by each node in the network. In reactive routing protocols, nodes maintain their routing tables on a need-to-use basis. For more information about those two classes, please refer to Natesapillai *et al*.[35]

Although many works have compared such routing protocols (e.g., refs. [36–38]), this has been mostly carried out in simulation and outside the scope of swarm robotic applications, wherein a large quantity of highly dynamic nodes need to be considered. Within such assumptions, the class of proactive routing protocols utterly falls apart. Besides being unsuitable to use in highly mobile nodes, proactive routing requires a high communication cost to constantly maintain all topological information.

Therefore, as swarm robotics aims for scalability under an increasing number of robots and mobility rate within the network, this work will focus on reactive routing protocols. One of the most well-known reactive protocols is the AODV.

4.1. Ad hoc On-demand Distance Vector

The AODV routing protocol is one of the most adopted reactive MANET routing protocols.[39] This protocol exhibits a good performance on MANETs, thus accomplishing its goal of eliminating source routing overhead. Nevertheless, at considerably high rates of node mobility, it requires the transmission of many routing overhead packets. Despite this limitation, the AODV has been extensively applied in most wireless equipments, such as the one used on the robotic platforms *eSwarBots*[13] (Fig. 6(a)); and the Original Equipment Manufacturers *RF* (*OEM-RF*) *Xbee Series 2* from Digi International[40] (Fig. 6(b)).

Under the AODV protocol, when a robot A needs to communicate with robot B, it broadcasts a route discovery message to its neighbors (i.e., local broadcast), including the last known sequence number for that destination.[41] The route discovery is flooded through the network until it reaches a robot that has a route to the destination. Each robot that forwards the route discovery creates a reverse route for itself back to robot A. When the route discovery reaches a robot with a route to robot B, that robot generates a *route reply* that contains the number of hops necessary to reach robot B and the sequence number for robot B most recently seen by the robot generating the route reply. Each robot that participates in forwarding this route reply back toward robot A creates a forward route to robot B. Hence, each robot remembers only the next hop and not the entire route.

In order to maintain routes, AODV normally requires that each robot periodically transmits a *hello message*. Within the RDPSO algorithm, this may be accomplished at each step of the algorithm,

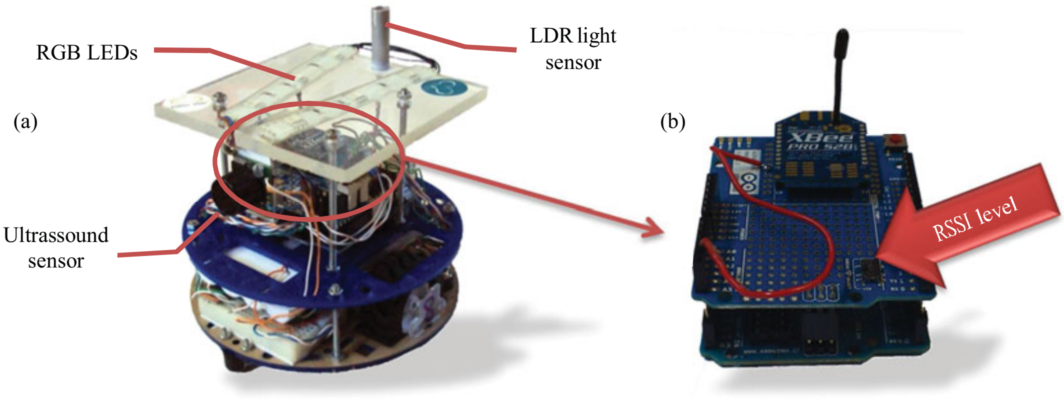


Fig. 6. (Colour online) (a) *eSwarBot* platforms presented in Couceiro *et al.*[13] (b) Electrical modification of *XBee Series 2* from Digi International[40] to provide the RSSI signal output.

i.e., after reaching the desired position $x_n\,[t+1]$, thus benefiting from the need to share its current position in order to ensure MANET connectivity (Section 3.1). A previously defined link may be considered to be broken if a robot does not receive three consecutive hello messages from a neighbor. Under that condition, any upstream robot that has recently forwarded packets to a destination using that link is notified via an *unsolicited* route reply containing an infinite metric for that destination. Upon receipt of such a route reply, a robot must acquire a new route to the destination using the route discovery once again.

4.2. *RDPSO-based AODV*

Although the mechanics of the AODV are quite transparent for users in wireless technology (e.g., *OEM-RF XBee Series 2*), one may need to extend the original AODV features so as to further adapt it to the application itself (e.g., ref. [19]). In this work, the AODV is extended based on two key elements: (i) as the teams of robots begin connectivity by means of the EST initial deployment (cf., ref. [12]), a *node discovery* functionality was introduced; and (ii) the mobility of robots within the RDPSO behavior is taken into account so as to establish more stable routes.

The node discovery basically allows discovering the IDs of all robots that have joined the network. Each robot will then broadcast a node discovery command throughout the network. All robots that receive the command will send a response that includes its own address. A timeout is defined by the node discovery sender, thus allowing specifying an amount of time a robot will spend in discovering its teammates. In other words, the node discovery functionality is highly suitable as the RDPSO handles multiple swarms and it may be difficult to predefine a population of specific robots to form a swarm in advance. Moreover, such strategy avoids the need to configure the address of each robot independently as each robot will acquire the default ID of its teammates in the beginning of the mission. Therefore, after each swarm is deployed within the scenario, the very first action that robots must perform is the node discovery command. Afterwards, the route discovery will be carried out (cf., previous section) and the mission will start.

Subsequently, it is possible to improve the AODV based on the mobility of robots by first understanding how robots may generally behave within the RDPSO algorithm. As previously stated in Section 2, the RDPSO model depends on the sensed information (1), both cognitive and social, and the inertial coefficient based on the approximate fractional difference of order α (3). That being said, a robot may estimate where a neighbor, i.e., one-hop robot, will be in the next iteration by knowing its previous positions, its best position so far, and the social solution of the group.

The later situation is the simplest one as each robot is always aware of the best solution of the whole group so far (Section 3.2). Hence, this requirement does not increase the memory complexity of the algorithm at all.

Similarly, a robot may know the best position of its neighbors as it is intrinsic to the communication packet structure shared when robots improve their individual solution (Section 3.2). For this situation, each robot will need to keep the position received by robots when they are able to improve, i.e.,

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

$f_n\,[t+j] > f_n\,[t]$, $j \in \mathbb{N}$. Nevertheless, the position of non-neighbor robots may be discarded as this is a distributed strategy that only considers information from one-hop nodes. Therefore, this results in an addition of the memory complexity per robot equal to the number of neighbor robots, i.e., $\mathcal{O}\,(N_b)$. Note, however, that this only represents memorizing twice N_b bytes necessary to represent the planar best position of each neighbor robot.

The most memory demanding situation will be inevitably memorizing the position of neighbors over time. Based on Eq. (3), one may compute the motion of robots with the information of four last steps, i.e., $v_n\,[t-j]$, $j=0,\ldots,3$. As neighbor robots share their current position $x_b\,[t]$, $b \in N_b$, a robot needs to memorize the two consecutive positions, $x_b\,[t]$ and $x_b\,[t-1]$, of all its neighbors so as to calculate their current velocity $v_b\,[t]$ (cf., Eq. (2)). In other words, a robot will need to keep track of positions of all its N_b neighbor robots for the last five steps to estimate their position, i.e., $\mathcal{O}\,(5N_b)$.

In sum, to extend the AODV based on the RDPSO behavior, one needs to increase the memory complexity of robots by $\mathcal{O}\,(6N_b)$. Note that this is a small increment to the memory complexity of each robot when compared with the benefit that this novel mechanism may provide in reducing the communication complexity of the whole swarm.

Having the information described above, each robot may be able to estimate all neighbors' next position $x_b\,[t+1]$ by means of Eqs. (1)–(3). Nevertheless, as the RDPSO is endowed with a stochastic effect, i.e., r_i, $i=1,2,3,4$, it is almost impossible for a robot to estimate the neighbors' exact next position accurately. However, one may improve the precision of such estimate by considering the expected values of the uniform random parameters. In other words, for the position estimate of the neighbors, a deterministic simplified version of the RDPSO is considered. The deterministic simplified RDPSO is obtained by setting the random numbers to their expected values:

$$E\,(r_i) = \frac{1}{2},\ \ i = 1,\ 2. \tag{4}$$

Thus, for the deterministic simplified RDPSO, replacing the random factors r_i by $\frac{1}{2}$, Eqs. (1)–(3) may be rewritten in a single equation as

$$\begin{aligned} x^e_{n,b}[t+1] = &\left(-1-\alpha+\frac{1}{2}\sum\nolimits_{i=1}^{4}\rho_i\right)x_{n,b}[t] + \left(\frac{1}{2}\alpha\right)x_{n,b}[t-1] + \left(\frac{1}{3}\alpha+\frac{1}{6}\alpha^2\right)x_{n,b}[t-2] \\ &+\left(-\frac{1}{24}\alpha^3-\frac{1}{24}\alpha^2+\frac{1}{12}\alpha\right)x_{n,b}[t-3] + \left(\frac{1}{24}\alpha^3-\frac{1}{8}\alpha^2+\frac{1}{12}\alpha\right) \\ &\times x_{n,b}[t-4] + \frac{1}{2}\rho_1\chi_{1n,b}[t] + \frac{1}{2}\rho_2\chi_{2n,b}[t], \end{aligned} \tag{5}$$

in such a way that $x^e_{n,b}\,[t+1]$ represents the position of robot b estimated by its neighbor n. Note that the remaining parameters in Eq. (5) are explained in Section 2. Although the estimated position is unlikely to be exactly the same as the real position, i.e., $x^e_{n,b}\,[t+1] \neq x_b\,[t+1]$, a good approximation may be enough to select if robot b may be a candidate to be the intermediate in route between source and destination robots.

Therefore, to improve the AODV routing protocol, when a source robot wants to send a packet to a destination robot, it will first estimate the next position of neighbor robots. Then it will recognize the intermediate robot that can participate in the routing of the message. The robot can be selected as the next hop if its estimated position is the closest to the destination robot, i.e., the one with the smallest Euclidean distance,

$$\mathrm{ID}_b = \underset{b\,\in\,N_b}{\operatorname{argmin}}\, d\left(x^e_{n,b}\,[t+1]\,,x^e_f\,[t+1]\right), \tag{6}$$

where ID_b represents the ID of robot n's neighbors that has the smallest distance to the destination robot, and $x^e_f\,[t+1]$ is the estimated position of the destination robot. After the message reaches the selected robot, the same process is carried out to assess the neighbor robot that would yield the next

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

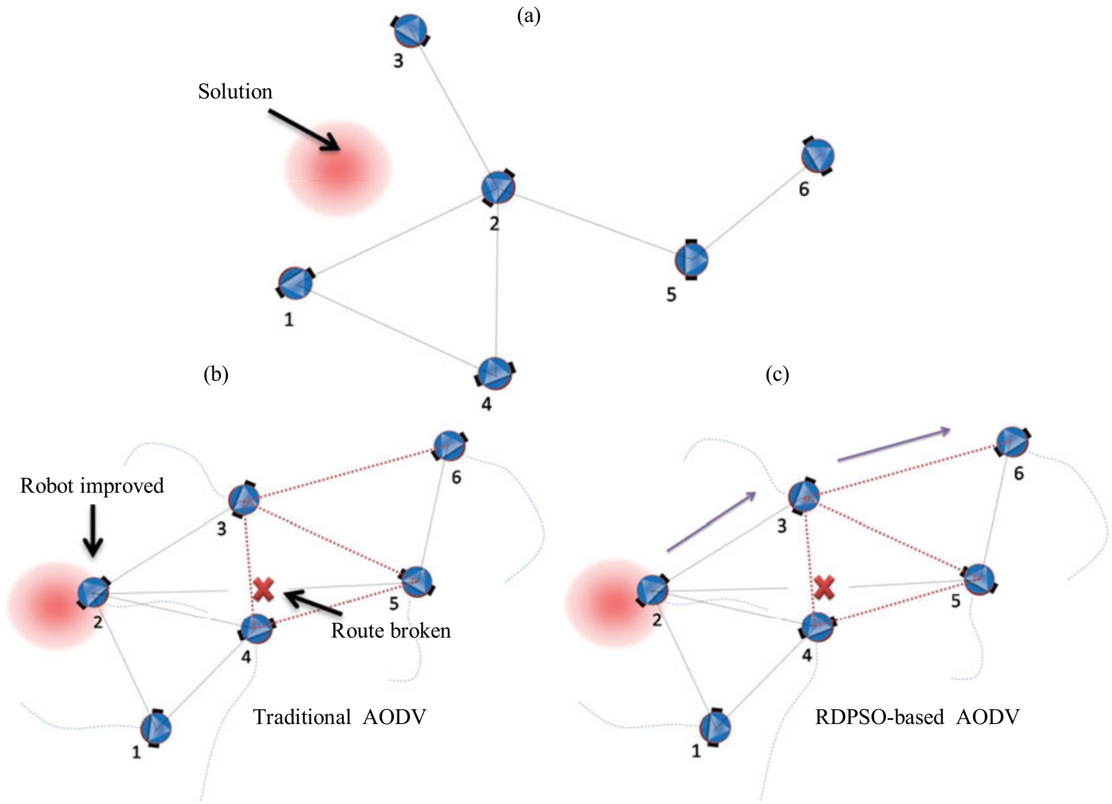


Fig. 7. (Colour online) The RDPSO-based AODV routing protocol. Red bolder lines between robots represent that there exists a possible link between them, but the AODV protocol is unaware of it. (a) The robots start connected by means of the EST initial deployment strategy, thus enforcing the MANET connectivity of the whole swarm.[12] The node discovery and the route discovery allow to retrieve the ID of all robots and build the routes between them (blue thin lines). (b) After a while, robot 2 improves and tries to broadcast its new solution and position to the whole network. However, as robot 2 is unable to communicate with robot 6 by means of the route built previously using AODV, a new route discovery needs to be sent (red thick lines). (c) Using the RDPSO-based AODV will allow robot 2 to choose the neighbor that is near robot 6, i.e., robot 3, which will forward the message to its destination, i.e., robot 6.

most fitted hop. Hence, source, destination, and candidate robots for next hop are the inputs of the herein proposed strategy for each robot. It is noteworthy that the information that will be used from the destination robot will be the last known information obtained from the broadcast to the whole swarm (cf., Section 3.2). Although the destination robot is likely to have changed its position in the meantime, the idea is to have an estimate of the region as to send the message and choose the most adequate path.

Routes established within such strategy are more stable and have less overheads than the original AODV routing method. Nevertheless, this is a greedy distributed strategy, and it may happen that a robot cannot find any intermediate node as the next best hop. For instance, the source robot may choose an incorrect neighbor robot based on its location without knowing that it may not have any other neighbors at all beside itself. In this situation, i.e., when a message returns to a robot that has forwarded it or to the source robot, then the common AODV mechanism of route discovery is used between that robot and the destination robot (cf., Section 4.1).

To easily understand the strategy proposed herein, Fig. 7 presents an illustrative example of a swarm under the RDPSO algorithm. In the beginning (Fig. 7(a)), and due to RDPSO main mechanisms,[12] robots are able to communicate between themselves, thus guaranteeing the MANET connectivity. Since the AODV routing protocol is the one adopted in this work, its main mechanism to retrieve all routes between robots is fulfilled, i.e., route discovery, as presented in Section 4.1. The routes between robots are represented by blue thin lines that connect them. Because of the particularities of RDPSO,

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

the node discovery is carried out so as to retrieve the IDs of all robots within the same swarm. While any robot improves, they continue exploring the scenario informing its neighbors about its position to maintain the MANET connectivity (Section 3.1). After a while (Fig. 7(b)), robot 2 is able to improve its cognitive solution, thus informing all other robots within the swarm (Section 3.2). Since robot 2 cannot communicate with robot 6, and considering the traditional AODV, a new route needs to be found, i.e., the route discovery needs to be fulfilled once again. These new routes are represented by red thick lines that connect the robots. Nevertheless, the route discovery mechanism requires successive local broadcasts that may overload the communication channel. Figure 7(c) depicts the mechanism inherent to the RDPSO-based AODV. Within such strategy, robot 2 will choose the nearest neighbor that presents the smallest distance to robot 6 (cf., Eq. (6)). As robot 2 is able to directly communicate with robot 6, it will forward the message to robot 6.

The whole RDPSO communication procedure for a robot n may be briefly summarized as presented in Algorithm I. Note that Algorithm I only focuses on the shared information between robots and the routing protocol. For a detailed description of the RDPSO main behavior, please refer to Couceiro *et al.*[42]

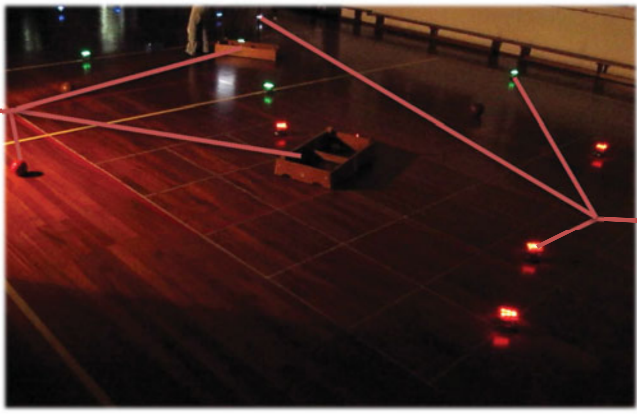
Algorithm 1. Sharing information within the RDPSO.

$starting\,()$// wait for information about initial position $x_n\,[0]$ and $swarmID$
$fullIDs = node\_discovery\,(swarmID)$// full list of robot IDs from the same swarm $swarmID$
$routesIDs = route\_discovery\,(fullIDs)$// list of routes within the same swarm $swarmID$
$Main\_Loop$
  If $swarmID \neq 0$ // it is not an excluded robot
    $send\,(0, x_n\,[t])$ // local broadcast may be avoided applying recognition techniques in visual range (Section 3.1)
    $f_n\,[t] = sense\,()$ // evaluate individual solution $f_n\,[t]$
    If $robot\_improved\,(f_n\,[t-1]\,,\, f_n\,[t])$ // robot n will globally broadcast its current solution and position (Section 3.1)
      $listIDs = send\,(1, x_n\,[t]\,,\, f_n\,[t])$ // $listIDs$ is an array of robot IDs that did not received the message
      $resend\,(listIDs)$ // use the RDPSO-based AODV
      If $call\_robot\,()$ // robot n may call a new robot from the excluded group to its swarm
        $send\,(0, swarmID\_call)$ // broadcast the possibility to receive a new robot
      If $create\_swarm\,()$ // robot n may create a new swarm from the excluded group
        $send\,(0, swarmID\_new)$ // broadcast the possibility to create a new swarm
    If $received\_fwdmsg\left(ID_f, x_n\,[t]\,,\, f_n\,[t]\,, routesIDs\right)$ // ID_f represents the ID of the destination robot
      $resend\left(ID_f\right)$ // use the RDPSO-based AODV
  Else // it is an excluded robot
    $wander\,()$ //Section 3.3
    $send\,(0, x_n\,[t])$ // local broadcast may be avoided applying recognition techniques in visual range (Section 3.1)
    $f_n\,[t] = sense\,()$ // evaluate individual solution $f_n\,[t]$
    If $received\,(swarmID\_call)$ Or $received\,(swarmID\_new)$ // call for a new robot or swarm received
      $listIDs = send\,(1, x_n\,[t]\,,\, f_n\,[t])$ // $listIDs$ is an array of robot IDs that did not received the message
      $resend\,(listIDs)$ // use the RDPSO-based AODV
End // until stopping criteria (e.g., convergence, time)
$resend\,(listIDs)$ // RDPSO-based AODV function
  For $i = 1$ to len $(listIDs)$ // check unreached robots one by one from $listIDs$
    For $j = 1$ to N_b // estimate position of its N_b neighbors (Eq. (5))
      $b = fullIDs\,(j)$
      $x^e_{n,b}\,[t+1] = estimate\_pos\left(x_{n,b}\,[t]\,,\ldots, x_{n,b}\,[t-4]\,,\, \chi_{1,2_{n,b}}\,[t]\right)$
    $ID_b = \min\limits_{b \in N_b} d\left(x^e_{n,b}\,[t+1]\,,\, x^e_{listIDs(i)}\,[t+1]\right)$ // find closest neighbor to robot $listIDs\,(i)$ (Eq. (6))
    If $ID_b = listIDs\,(i)$ // the unreached robot $listIDs\,(i)$ is a neighbor
      $send\,(ID_b, x_n\,[t]\,,\, f_n\,[t])$ // send message directly to robot ID_b
    Else
      If $find\,(ID_b, routesIDs)$ // the robot ID_b already exists in the route necessary to reach $listIDs\,(i)$
        $routesIDs = route\_discovery\,(fullIDs)$ // necessary as it is unable to reach the destination robot
        $send\,(listIDs\,(i)\,,\, x_n\,[t]\,,\, f_n\,[t])$ // send message to robot $listIDs\,(i)$
      Else
        $routesIDs = update\_route\,(n, ID_b, listIDs\,(i))$ // update the route necessary to reach $listIDs\,(i)$
        $send\,(ID_b\,[listIDs\,(i)]\,,\, x_n\,[t]\,,\, f_n\,[t]\,, routesIDs)$ // send message to robot ID_b so as to reach $listIDs\,(i)$
End

Next section evaluates the communication complexity of RDPSO with and without the strategies proposed herein.

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

(a)



(b)

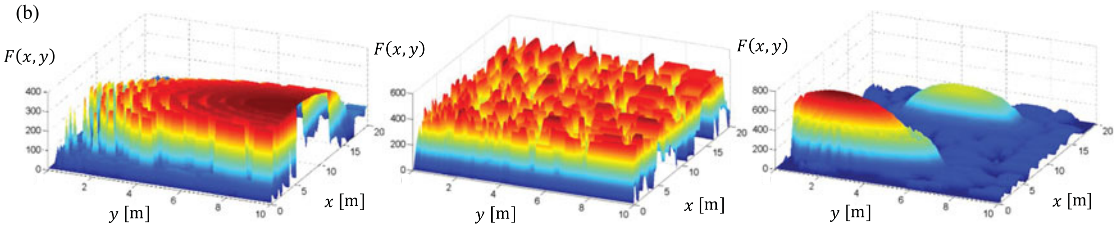


Fig. 8. (Colour online) Experimental setup. (a) Arena with 3 swarms (different colors) of 5 *eSwarBots* each. (b) Virtual representation of target distribution over day under different lighting conditions.

5. Experimental Results

This section is divided into three sections exploring and comparing the properties of the "regular" version of the RDPSO (previously presented) to its counterpart version proposed in this paper – the "optimized" RDPSO.

5.1. Real-world experiments

In this section, the effectiveness of the proposed communication methodology on a group of 15 *eSwarBots*,[13] i.e., $N = 15$ is explored, performing a distributed exploration task under the RDPSO behavior (Fig. 8). As this paper emphasizes on the analysis of the communication complexity of the RDPSO, the convergence of the algorithm itself was neglected. This may only be considered as the communication methodology proposed here does not affect the decision-making of robots since the same useful information is always shared between teammates. Therefore, as *eSwarBots* are equipped with light-dependent resistor (LDR) light sensors that allow sensing the brightness of light, their solution was affected by the current room lighting conditions, either natural or not. Just for the purpose of illustrating the variability of light over time, Fig. 8(b) represents the intensity of light $F(x, y)$ over a day. Such data were obtained sweeping the whole scenario with a single robot with the light sensor connected to a 10-bit analog input resulting in a resolution of approximately 5 mV.

Since the RDPSO is a stochastic algorithm, it may lead to a different trajectory convergence whenever executed, thus resulting in a different amount of information exchanged between robots. Therefore, two sets of 20 trials of 360 sec each were considered. In other words, the "regular" RDPSO (first set of trials) was compared with the "optimized" RDPSO (second set of trials), i.e., the extension of the RDPSO based on the strategies presented in Sections 3 and 4. At each trial, the robots were deployed in a 20×10 m indoor scenario (Fig. 8(a)), ensuring the initial connectivity of each swarm in a spiral manner (cf., Section 2 or Couceiro *et al.*[12] for a more detailed description).

The inter-robot communication was carried out using *ZigBee* 802.15.4 wireless protocol. Although the *XBee Series 2* modules allow a maximum communication range of approximately 30 m in indoor/urban environments (cf., ref. [13]), the signal quality of the received data is highly susceptible to obstacles and other phenomena (e.g., communication reflection and refraction), thus resulting in

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

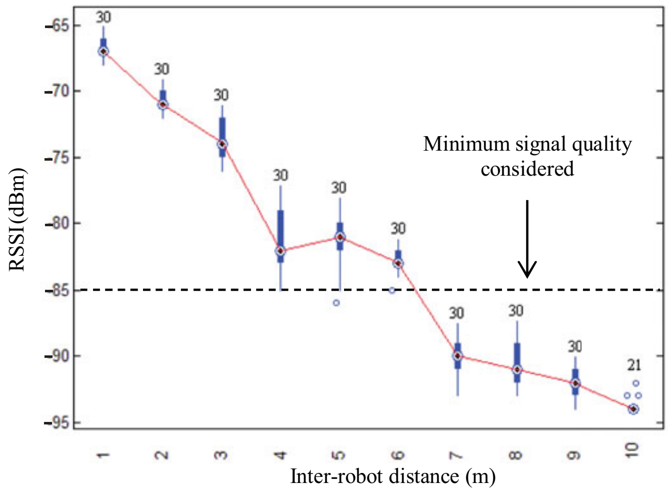


Fig. 9. (Colour online) Measured RSSI versus distance from two robots located in the experimental scenario.

the loss of packets with increase in inter-robot distance. In fact, preliminary experiments to test the *XBee* modules on the same scenario showed that the connectivity starts failing above 10 m (Fig. 9). Therefore, to allow a more realistic and conservative approach, the connectivity between robots was maintained using the received signal quality. To this end, the *XBee* modules were modified to provide the RSSI signal output (cf., Fig. 6(b)). This RSSI output is available as a pulse width modulation (PWM) signal of 120 Hz, where the duty cycle DC varies according to the signal level relative to the receiver sensitivity it follows:

$$\mathrm{DC} \approx 38 + 0.1 \times \mathrm{RSSI}, \tag{7}$$

in which the parameters of the straight-line equation were obtained in the equipment data sheet.[13] For instance, a 30% duty cycle (i.e., 1.5 V) is equivalent to approximately the receiver sensitivity of –94 dBm. In order to choose a minimum signal threshold that would ensure MANET connectivity, Fig. 9 presents the relation between the RSSI and the distance between two robots randomly wandering in the same scenario presented in Fig. 8(a) while sending 30 periodic messages every 2 sec to each other at each different distance. The RSSI versus the inter-robot distance was represented using a box-plot chart, in which the ends of blue thicker lines and the circle in-between correspond to the first and third quartiles and the median values respectively. The numbers on top of each set of measures correspond to the number of messages received at each different distance.

As expected, in an indoor scenario endowed with obstacles, the signal quality is not proportional to the inter-robot distance. In fact, even the inverse relationship between distance and signal quality considered in many works 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. Moreover, for a distance of above 10 m, a robot is only able to receive approximately 2/3rd of the messages. Therefore, to avoid the possible loss of packets due to the distance between robots, the minimum allowed receiver power was set to –85 dBm, i.e., for distances less than 6 m. This allows avoiding the possible loss of packets due to low levels of signal quality.

A minimum initial and maximum number of 0, 3, and 4 swarms were used, thus representing an initial swarm size of $N_S = 5$ *eSwarBots*. The maximum traveled distance between iterations was set as 0.25 m, i.e., $\max |x_n[t+1] - x_n[t]| = 0.25$. In other words, each robot could only travel a maximum of 0.25 m without considering the position of its neighbor robots so as to ensure the MANET connectivity.

As stated previously, by employing the optimized communication strategies given in Sections 3 and 4, it is expected to significantly reduce the communication cost of the RDPSO algorithm. One of the methods to evaluate the communication cost consists in counting the average number of packets sent and the processing time to handle the communication procedure, i.e., pause time, for each robot over 360 sec of each trial. The number of packets sent was easy to retrieve since a robot under the

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

Table II. Communication cost.

	AVG ± STD number of packets	AVG ± STD pause time (sec)
“Regular” RDPSO	742 ± 24	126 ± 4
“Optimized” RDPSO	415 ± 37	39 ± 7

“regular” RDPSO communicates after each iteration step with its own swarm, i.e., if it is a swarm of five robots, then the robot will send four packets, while in the “optimized” RDPSO the robot follows the rules presented in Section 2. Regarding the pause time inherent to the whole communication procedure, a timer was used to count the time before entering the function that allows for a robot to send and receive the data packets from its own swarm. It is noteworthy that during that time the robot is unable to perform any other action. Table II compares the average (AVG) and standard deviation (STD) communication cost of the RDPSO with and without the proposed strategy.

As it is possible to observe, the number of messages decreases significantly using the proposed methodology. This is highly valuable as the number of exchanged messages has a high influence on the power consumption of each robot. On the other hand, reducing the number of times each robot needs to share its information allows reducing the time allocated for such task. Note that this is not proportional since that, in the “optimized” RDPSO, robots communicate only with their neighbors at each iteration step (since *eSwarBots* are not equipped with sensing capabilities that allow retrieving teammates’ position). Communication with the whole swarm is constrained by how each robot improves over time. In other words, while each robot allocates approximately 35% of the mission time to exchange information within the “regular” RDPSO, this novel approach allows reducing this value to approximately 10%, thus increasing robot’s mobility. This is due to both requiring less data to be exchanged (Section 3) and also minimization of route discovery messages inherent to the RDPSO-based AODV (Section 4). In other words, the approach proposed herein would be more power-efficient and allow each robot to spend less time without moving than the “regular” one.

Nevertheless, the efficiency of a communication paradigm cannot be measured by only comparing the total number of exchanged packets. One of the most well-known performance metrics to evaluate the network throughput is the packet delivery ratio. The packet delivery ratio is calculated by dividing the number of packets received by a robot by the number of packets sent to it. This allows specifying the packet loss rate, which limits the maximum throughput of the network. Therefore, the average packet delivery ratio was evaluated based on the number of robots within the same swarm (either active swarm or the socially excluded group). As previously mentioned in Section 3, and further detailed in Couceiro *et al.*,[22] the RDPSO uses a “punish”–“reward” mechanism to avoid sub-optimality by socially excluding and including robots within active swarms. In other words, at some point over 360 sec of each trial, i.e., 7,200 sec for each set, a swarm may be formed by only two robots or even by 15 robots from the population. In other words, Fig. 10 depicts the average packet delivery ratio when swarms are formed by a specific number of robots, although some of these cases, namely swarms formed by less than three robots or by more than 10 robots, may only occur at some occasions (around 5% of the whole time).

As one may observe, there is a sharp decrease in the packet delivery ratio for the “regular” RDPSO when a swarm is formed by more than 10 robots, dropping down to approximately 65% for a maximum network load of 15 robots. This is explained by the high number of exchanged messages that, for a network load of more than 10 robots, does not satisfy the capacity of the buffer, or the packet buffering time exceeds the time limit. As the “optimized” RDPSO significantly decreases the number of exchanged messages (cf., Table II), robots are still capable of receiving more than 90% of the data even within a swarm of 15 robots.

The first key contribution of this paper, i.e., the efficient way to share information within the RDPSO algorithm (Section 3), is the major reason for such a significant reduction in both communication cost (Table II) and number of dropped packets (Fig. 10). Although the adapted AODV improves the communication efficiency of the RDPSO algorithm, it is still not clear how advantageous this specific extension may be so far.

The routing overhead has been frequently used in the literature to evaluate routing algorithms, being commonly represented by the ratio between the number of route discovery messages and the

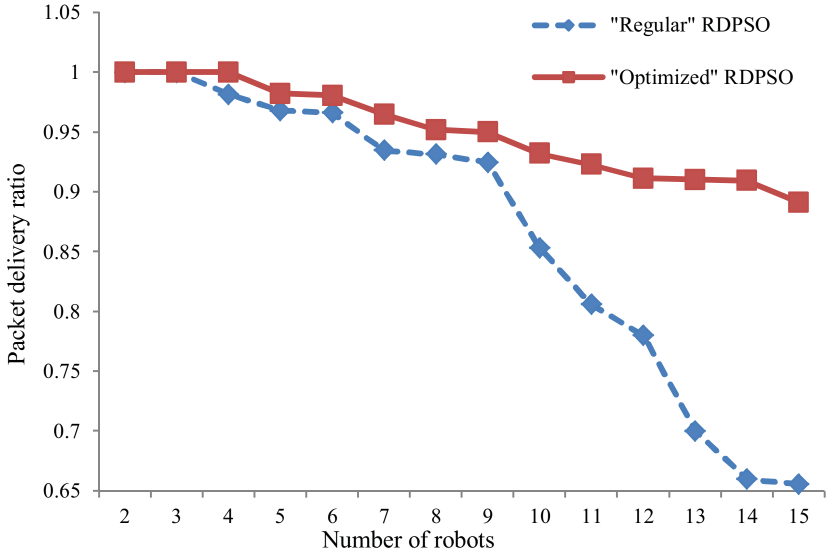


Fig. 10. (Colour online) Packet delivery ratio within robots from the same swarm.

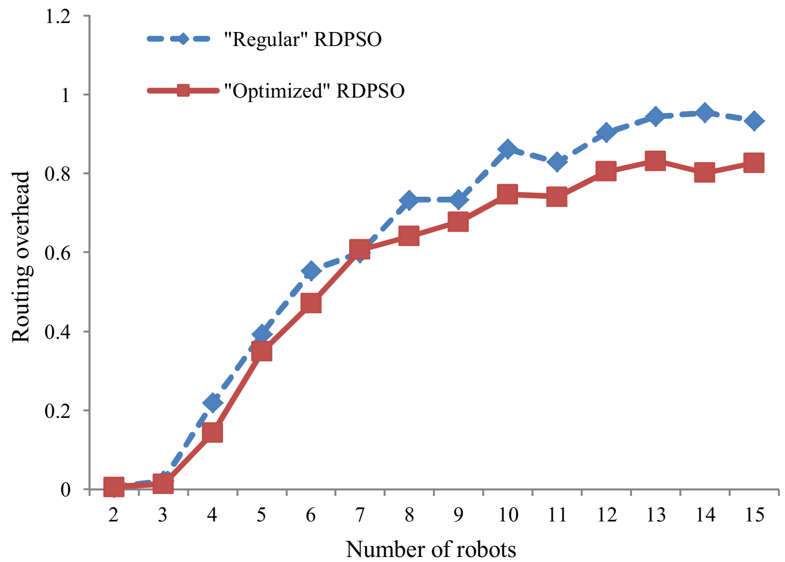


Fig. 11. (Colour online) Routing overhead within robots from the same swarm.

number of data packets. Once again, let us compare the routing overhead of the "regular" RDPSO with the "optimized" RDPSO for different team sizes from two to 15 robots under 7,200 sec of each set of trials (Fig. 11).

Once again, the "optimized" RDPSO clearly overcomes the "regular" RDPSO for larger population of robots. Even though the number of data packets is reduced due to the efficient way to share information between robots (Section 3), the number of route discovery messages decreases more significantly (Section 4), thus resulting in a smaller routing overhead for a larger number of robots. It would be expected to have a worse routing overhead ratio when robots communicate less while moving since the routes would be completely outdated. Nevertheless, the RDPSO-based AODV is able to reduce the number of route discovery messages in such a way that it allows overcoming that issue. This is due to the proposed geographically based AODV that takes into account the dynamics of RDPSO, thus creating on-the-fly routes (Fig. 7). However, how better are these new routes when compared with the alternatives returned by the traditional AODV? To answer this question, one needs to analyze the number of hops forming such routes.

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

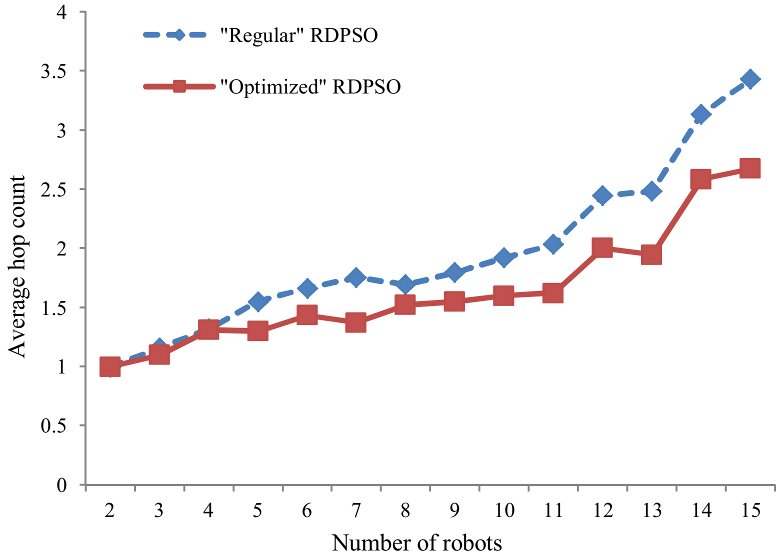


Fig. 12. (Colour online) Average hop count within robots from the same swarm.

The average hop count may be represented by the sum of the number of hops necessary to deliver the packets from their sources to destination divided by the total number of successfully delivered packets. The average hop count is measured by the number of hops.

As Fig. 12 depicts, the applicability of the novel AODV routing protocol may be observed for a swarm of at least five robots. For smaller swarms, the improvement of the RDPSO-based AODV is meaningless, which on the other hand turns out to be a worse alternative to the traditional AODV since it slightly increases the memory complexity of the algorithm (cf., Section 4). However, as analyzing swarm algorithms within small populations may not represent the required collective performance (cf., ref. [43]), let us focus on larger team sizes, i.e., above five robots. As it is possible to observe, in some situations, the RDPSO-based AODV reduces around 20% the number of required hops to deliver a packet. Although this may not seem relevant, this contributes to a smaller pause time, and consequently a higher mobility of robots. Moreover, reducing the number of hops necessary to deliver the packets also reduces the power consumption of each robot, thus increasing the autonomy of the whole swarm.

5.2. *Temporal analysis*

It is noteworthy that the two key contributions of this paper, i.e., the efficient way to share information within the RDPSO algorithm, and the adapted AODV routing protocol, result in significant differences compared with its “regular” counterpart. Moreover, such differences increase with the number of robots, thus improving the scalability of the RDPSO algorithm. Yet, in order to further explore inter-robot communication dynamics under the “optimized” RDPSO, let us analyze how such information is shared within different social statuses, i.e., within socially active and excluded swarms.

In order to achieve this, the number of local and global broadcasts within each swarm was analyzed. For a better understanding of how robots within the RDPSO evolve, let us take a look at one of the 20 trials in which the “optimized” RDPSO was evaluated, i.e., a single trial of 360 sec. Figure 13 depicts the distribution of robots (Fig. 13(a)) and highlights the respective total number of local (Fig. 13(b)) and global broadcasts (Fig. 13(c)) within each swarm over time. While the colored lines correspond to each socially active swarm, respectively *R* (red), *G* (green), and *B* (blue) swarms, the dark dashed line corresponds to the socially excluded swarm. The mission starts with five robots within each active swarm as stated previously. As one may observe, the number of workers in active swarms tends to decrease over time. This is an expected phenomenon as the resources begin to dwindle over time, i.e., in this specific case study robots are unable to find ever improving light intensities. At some point it is even possible to observe that swarms *B* and *R* extinguish, while swarm *G* proliferates, thus reaching a population of up to 11 robots. This happens right before the population in swarm *G* decreases to approximately seven robots. Consequently, this leads to an increase of socially excluded robots with

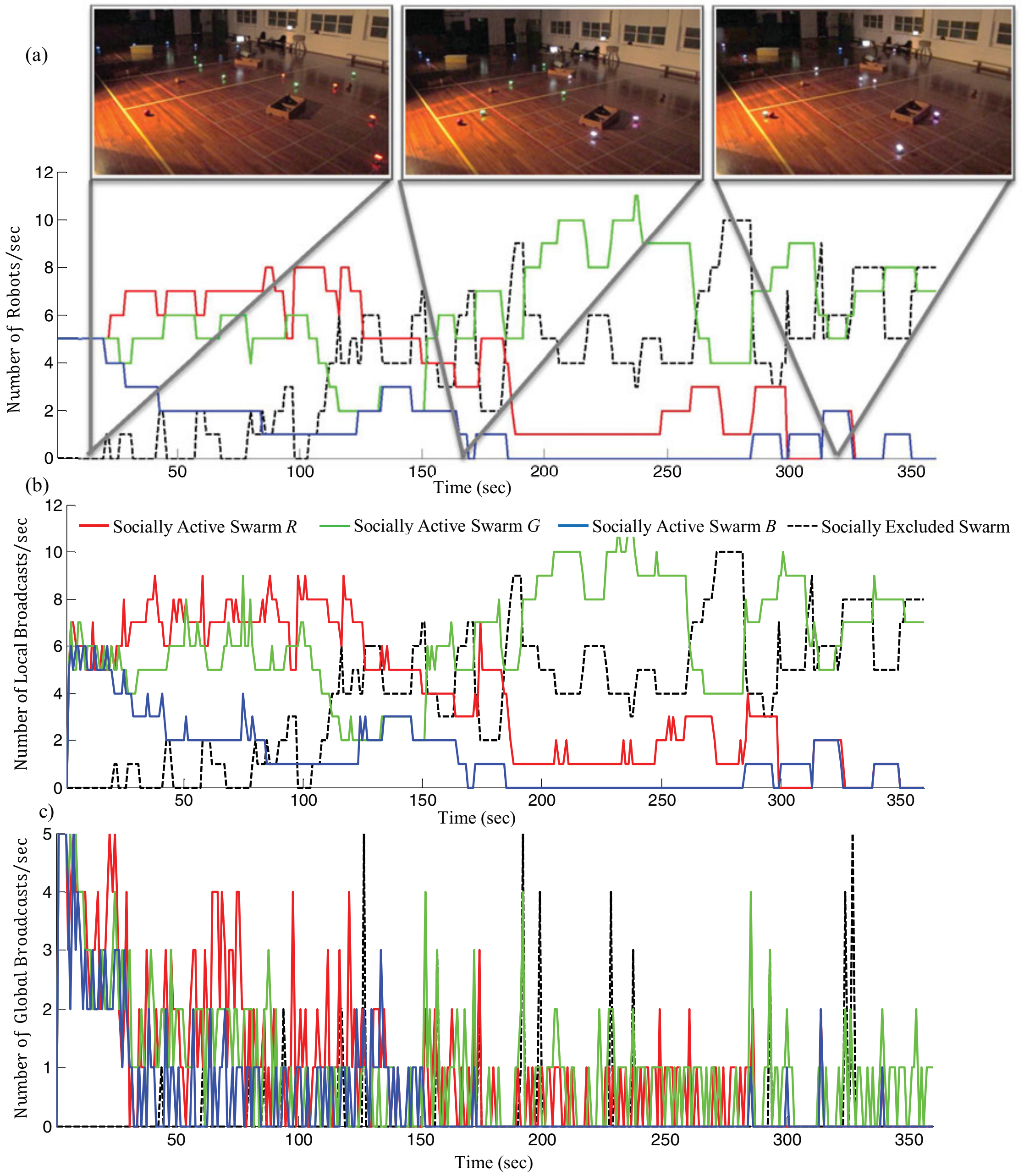


Fig. 13. (Colour online) Evolution of robotic swarms over a trial of 360 sec. (a) Population size; (b) number of local broadcasts; (c) number of global broadcasts.

a maximum of 10 robots after the 4th minute. Regarding local broadcasts, such temporal variations would be expected by considering the rules stated previously in Section 3. The local broadcasts necessary to maintain the network connectivity remain at each step of the algorithm, thus presenting a proportional amount to the number of robots within each swarm. Such proportionality is only broken when a socially active swarm claims a new robot or tries to create a new swarm (small peaks observed in colored lines). A rationale behind the global broadcasts is harder to achieve. As one may observe in general, socially active robots present a higher amount of messages flooded through the whole swarm. This is interesting to observe as such global broadcast is related to swarms' improvement that requires the global consent of the population. As a result, such global broadcasts diminish over time. This kind of global message seems to be significantly less recurrent in socially excluded swarms.

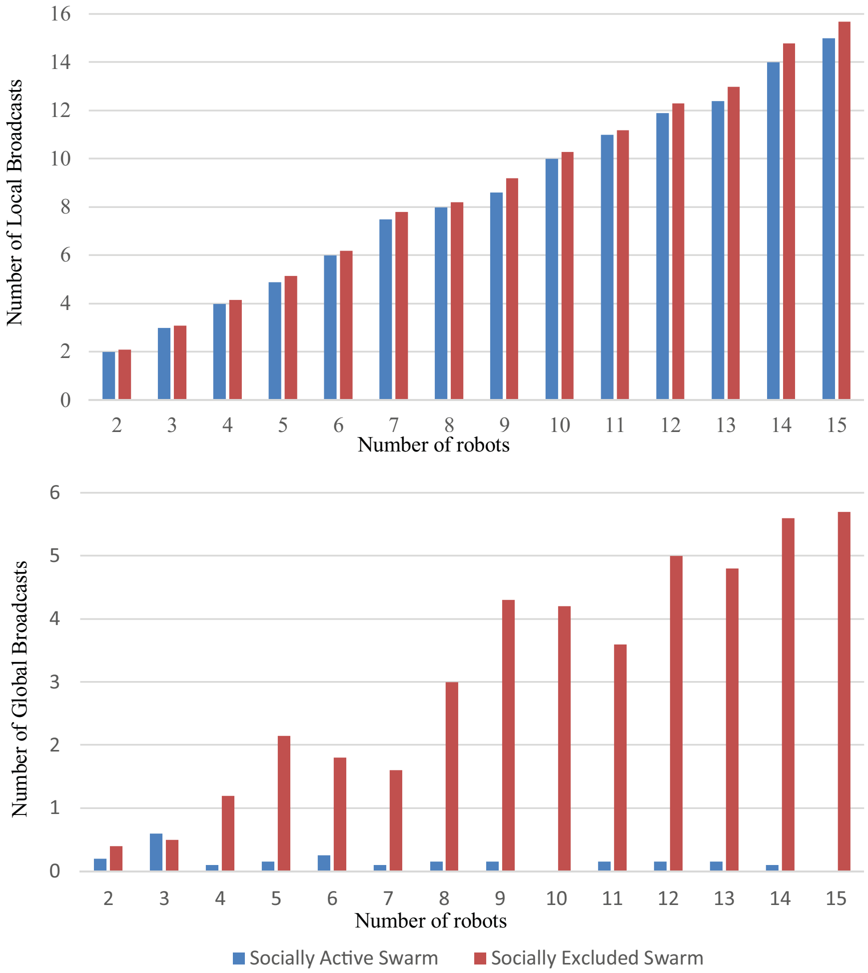


Fig. 14. (Colour online) Normalized temporal average number of local and global broadcasts.

As one may observe, the time a certain amount of robots are socially excluded may not correspond to the time for which the same amount of robots is socially active. Therefore, to further compare the information shared within the different social statuses over 7,200 sec of the whole set of experiment, a simple normalization of data over time was adopted. Figure 14 depicts the average number of local and global broadcasts within each swarm configuration. As a rule of thumb, the local broadcasts increase almost proportionally to the population of robots. This may be observed in both socially excluded and active swarms with a minor difference between both. The main difference between robots belonging to different social statuses may be seen in the number of global broadcasts. Socially excluded robots barely communicate with the whole group. In fact, such communication only depends on the improvement of socially active swarms if socially active swarms improve. Hence, as the overall amount of socially active robots decreases, the number of socially excluded robots increases and the probability of success (i.e., improving the current solution) also decreases. Consequently, this reduces the required number of global broadcasts from excluded swarms.

As the experiments presented so far are limited to a maximum number of 15 physical robots within the same swarm, it was necessary to perform simulation experiments to evaluate the scalability of the "optimized" RDPSO.

5.3. *Scalability evaluation through simulation*

The *Multi-Robot Simulator* (MRSim)[1] was used to evaluate the previously proposed "optimized" RDPSO. The MRSim is an evolution of the *Autonomous mobile robotics toolbox SIMROBOT*

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

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

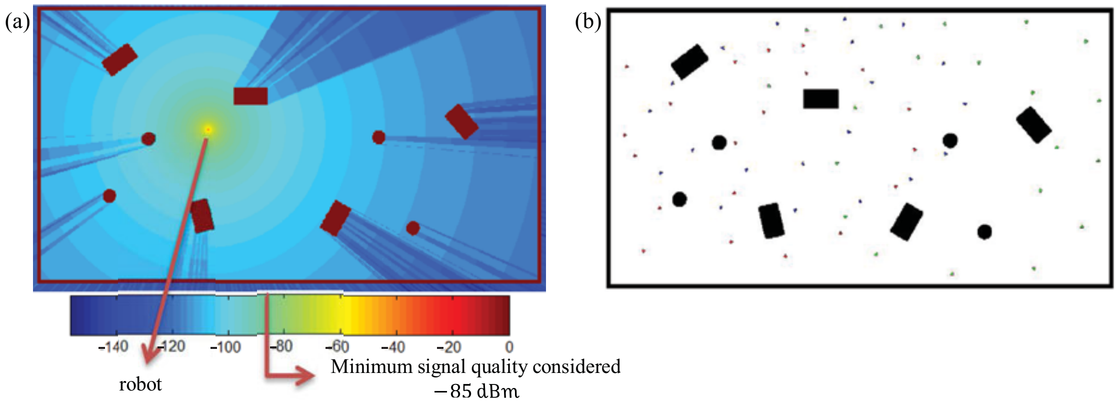


Fig. 15. (Colour online) Simulation experiments in a 20 × 10-m indoor scenario (sports pavilion). (a) *WiFi* communication propagation. (b) Setup with 3 swarms of 20 robots each (population of 60 robots) autonomously deployed based on the EST approach given in Couceiro *et al.*[12]

(*SIMulated ROBOTs*) previously developed for an obsolete version of *MatLab*.[44] The simulator was completely remodeled for the latest *MatLab* version and new features, such as mapping and inter-robot communication, were included.[45] In addition, MRSim also enables the addition of a monochromatic bitmap as a planar scenario and the configuration of its properties (e.g., obstacles and size, among others), as well as the implementation of features for each swarm robotic technique (e.g., robotic population and maximum communication range, among others) and configuration of robot's model (e.g., maximum velocity and type of sensors, among others).

Due to the lack of a preexistent model of *WiFi* propagation (radio frequency at 2.4 GHz) in MRSim, this work considered its implementation based on work by Luca *et al.*[46] The attenuation over the transmitter–receiver distance d (m) was calculated as:

$$L = l_c + 10\gamma \log d + \sum_{W} l_W, \tag{8}$$

where W represents the number of walls with attenuation l_W between the transmitter and the receiver. The constant factor l_c corresponds to the reference loss value at 1 m. This was defined as $l_c = 47.4$ dB, and experimentally validated in indoor scenarios by Luca *et al.*[46] The path loss exponent γ is usually defined between 2 and 4, wherein values near 2 correspond to propagation in free space and values near 4 represent lossy environments. The parameter γ was uniformly distributed between the interval 3 and 4, thus providing a stochastic effect on the communication propagation.[47]

In order to improve the understanding of how *WiFi* communication propagates in the scenario considered in this work, Fig. 15(a) depicts the range of communication power. Note that signal strength values are shown in decibel-milliwatt (dBm). As it is possible to observe, and considering the condition that the minimum receiver power allowed was set at –85 dBm (Fig. 9), a robot may be unable to communicate with its teammates in some zones due to occlusion by obstacles and distance.

As a mean of simplification and in line with the previous real experiments, the same 20 × 10-m indoor scenario (sports pavilion) was created on MRSim. Due to the computational cost of the simulator, which significantly increases with the number of robots, only experiments of up to 60 robots were possible.

As MRSim is a step-based simulator (without real-time iterations), the ratio between the number of packets exchanged within the "optimized" RDPSO and the "regular" RDPSO was analyzed. Note that this depends on the type of communication (i.e., local or global broadcast). For instance, in a swarm of 10 robots a global broadcast from a single robot corresponds to nine packets exchanged, i.e., one for each teammate. However, if the same robot has only four neighbors (one-hop robots), then a local broadcast will correspond to only four packets exchanged. Due to the stochastic nature of the RDPSO, box-plot charts were once again used to represent the ratio between the number of packets exchanged within the "optimized" RDPSO and the "regular" RDPSO over 30 trials with a maximum of 5,000 steps each (Fig. 16). To easily observe the differences, the ratio was averaged at

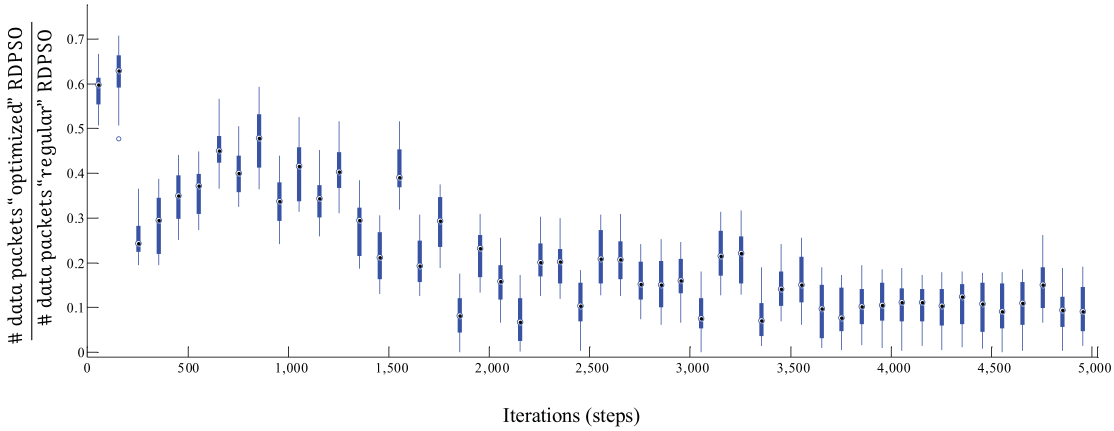


Fig. 16. (Colour online) Ratio between the number of packets exchanged using the "optimized" RDPSO and the "regular" RDPSO over the number of iterations in a population of 60 robots.

each 100 steps. Once again, note that the number of robots within the same swarm may vary from two robots to the total number of robots within the population (60 robots).

As one may observe, the difference between the "optimized" RDPSO and the "regular" RDPSO grows with time. The decreasing tendency observed in Fig. 16 is an expected phenomenon. As swarms' exploration within the "optimized" RDPSO advances, the number of global broadcasts necessary to converge to the optimal solution decreases (Section 3.2). After a certain amount of time (half the mission time), the "optimized" RDPSO is able to decrease the number of exchanged data packets to approximately 20% of the number of data packets exchanged under the "regular" RDPSO. In terms of communication cost this may be considered as a significant improvement. As an example, the *eSwarBot* platforms usually present battery autonomy of up to 4 hr without using the *XBee Series 2* modules. However, such autonomy drops to approximately 2 hr with constant data transmission.[13] Another example, such as the well-known *e-puck* robot, is even more significant.[48] The *e-puck*'s battery autonomy can drop from 3 hr to approximately 1 hr using the *WiFi* communication from the *Gumstix Overo COM*.

6. Discussion – Towards a Stigmergetic RDPSO

The RDPSO was proposed for the first time in 2011[22] by adapting DPSO[23] to swarm robotic applications. Although the communication between robots was initially studied in Couceiro *et al.*,[9] previous works have mainly focused on improving the evolutionary properties of the RDPSO, thus neglecting the scalability constraints that these may impose. Therefore, the authors would like to discuss the take-home message this paper brings forth and present the future expectations around the RDPSO algorithm.

The motivation behind this work was to explore a strategy for improving the scalability of RDPSO by optimizing its communication complexity. This was achieved by analyzing judiciously the Information to be explicitly exchanged between robots, and proposing a way to efficiently share it without decreasing the collective performance of the algorithm. Then, the well-known AODV was adapted based on RDPSO dynamics.

Real and simulation experiments were conducted to observe the effect of the proposed optimized strategy. The mission consisted of collectively exploring a 20×10−m scenario in which robots' cognitive solution was affected by the light sensed at their current position. The superiority of the "optimized" RDPSO over the "regular" RDPSO was especially visible in the number of packets exchanged between robots and the packet delivery ratio. Although the differences between the routing overhead and the required number of hops to deliver a packet were not significant for small groups of robots, the "optimized" RDPSO was still able to reduce both to approximately 20% less for swarms of 15 robots. Those differences were even more visible in the simulations with a swarm of 60 robots examining the ratio between the total number of packets exchanged within the "optimized" RDPSO and the "regular" RDPSO. Although in the beginning of the mission the "optimized" RDPSO

https://doi.org/10.1017/S0263574714000058 Published online by Cambridge University Press

presented a rather modest reduction of approximately 50% of the number of packets exchanged, as robots continuously explored the scenario, such differences increased to approximately 85%. To improve the analysis of the communication architecture within the RDPSO, the differences between the two social statuses were also represented, thus revealing that the principle of cooperation undergoes several phases that depend on more than just mission-related contextual information (e.g., sensed solution).

This dependency between the swarms gives rise to a *competitive evolutionary process* inherent to animal nature as described in the Darwinian principle of survival-of-the-fittest. On the other hand, as many other biological societies are involved in diverse survival conditions, the outcome of this competitive evolutionary process is reflected in social cooperation among the members from the same group. This is a highly recurrent process in nature denoted as *coopetition*.[49] For instance, certain birds are unable to reach parasites on some parts of their bodies, thus benefiting from preening one another. Hence, there is an entire flock of potential preeners that compete in the hope of establishing a beneficial cooperative relationship. To the similarity of the RDPSO, birds that try to be preened without preening others are excluded from this relationship as they do not compete.

These results paved the way toward an insightful reassessment and revolution of the RDPSO algorithm. Considering the recent advances in the control of aggregation behaviors without communication (e.g., ref. [5]), the most expected improvement would be the development of a *stigmergetic* RDPSO without significantly reducing the collective performance of swarms. In this case, the macroscopic capabilities of the RDPSO should be defined by spatial or dynamical conditions in the environment. In other words, the system and environment itself build a closed macroscopic feedback loop, which works in a collective way as a distributed control mechanism.[5] In this case, robots interact kinetically or through *stigmergy* effects.[50] For instance, emulating Darwin's survival-of-the-fittest without explicit communication would not only require robots to possess the capability of discerning collisions between obstacles and other robots but also between robots from different swarms. This could be attained by endowing robots with simple low-cost vision capabilities, such as the ArduEye vision sensor.[2]

All that being said, one may state that it is still difficult at this point to go from an algorithm sustained by explicit communication to a *stigmergetic* one. However, the authors argue that this paper provides an exhaustive rationale on the necessary explicit communication within the RDPSO that gives the first step in that direction.

7. Conclusions

An optimization of the communication procedure between robots under a collective swarm intelligence behavior, previously proposed and denoted as RDPSO, was presented in this paper. Moreover, the traditional AODV was improved considering robots' motion and behaviors inherent to the RDPSO. Such improvements were motivated by the need to use large teams of robots without significantly increasing the communication overhead. Several experimental results with up to 15 real robots and 60 virtual robots in a 20×10-m scenario clearly show the advantages of such an optimized strategy regarding the scalability of the algorithm, thus paving the way for future swarm applications of hundreds or thousands of robots. Therefore, in the future, and due to the flexibility of the herein proposed solution, this "optimized" RDPSO should be evaluated on larger teams of swarm robots under realistic applications such a multi-robot *Simultaneous Location and Mapping* (SLAM) that usually presents a communication bottleneck as the number of robots increase. Finally, we also intend to implement an estimation method to dimension the swarm of robots according to the environment topology and temporal constraints.

Acknowledgements

This work was supported by a PhD scholarship (SFRH/BD /73382/2010) granted to the first author, the research project CHOPIN (PTDC/EEA-CRO/119000/2010) by the Portuguese Foundation for Science and Technology (FCT), the Institute of Systems and Robotics (ISR), and the Engineering

[2] http://ardueye.com/

Institute of Coimbra (ISEC). The authors would also like to thank the referees for all their helpful feedback.

References

1. L. E. Parker, "Multiple Mobile Robot Systems," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, New York, NY, 2008), pp. 921–941.
2. M. J. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robot. Auton. Syst.* **16**, 321–331 (1995).
3. S. Onn and M. Tennenholtz, "Determination of social laws for multi-agent mobilization," *Artif. Intell.* **95**, 155–167 (1997).
4. B. B. Werger, "Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams," *Artif. Intell.* **110**(2), 293–320 (1999).
5. S. Kernbach, D. Häbe, O. Kernbach, R. Thenius, G. Radspieler, T. Kimura and T. Schmickl, "Adaptive collective decision-making in limited robot swarms without communication," *Int. J. Robot. Res.* **32**(1), 35–55 (2013).
6. M. J. Huber and E. Durfee, "Deciding When to Commit to Action During Observation-Based Coordination," *Proceedings of the First International Conference on Multi-Agent Systems* (1995) pp. 163–170.
7. M. Tambe, "Towards flexible teamwork," *J. Artif. Intell. Res.* **7**, 83–124 (1997).
8. L. E. Parker, "ALLIANCE: An Architecture for fault-tolerant multi-robot cooperation," *IEEE Trans. Robot. Autom.* **14**(2), 220–240 (1998).
9. M. S. Couceiro, R. P. Rocha and N. M. F. Ferreira, "Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms," *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, Kyoto, Japan (2011) pp. 284–289.
10. W. Sheng, Q. Yang, J. Tan and N. Xi, "Distributed multi-robot coordination in area exploration," *Robot. Auton. Syst.* **54**, 945–955 (2006).
11. D. Tardioli and J. L. Villarroel, "Real Time Communications Over 802.11: RT-WMP," *IEEE Internatonal Conference on Mobile Ad Hoc and Sensor Systems* (2007) pp. 1–11.
12. M. S. Couceiro, C. M. Figueiredo, R. P. Rocha and N. M. F. Ferreira, "Darwinian Swarm Exploration Under Communication Constraints: Initial Deployment and Fault-Tolerance Assessment," *Robot. Auton. Syst.* (2013; In Press).
13. M. S. Couceiro, C. M. Figueiredo, J. M. A. Luz, N. M. F. Ferreira and R. P. Rocha, "A low-cost educational platform for swarm robotics," *Int. J. Robots Educ. Art* **2**(1), 1–15 (Feb. 2012).
14. L. Sabattini, N. Chopra and C. Secchi, "On Decentralized Connectivity Maintenance for Mobile Robotic Systems," *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL (2011) pp. 988-993.
15. A. Casteigts, J. Albert, S. Chaumette, A. Nayak and I. Stojmenovic, "Biconnecting a Network of Mobile Robots Using Virtual Angular Forces," *IEEE 72nd Vehicular Technology Conference, Fall (VTC 2010-Fall)*, Ottawa, ON (2010) pp. 1038–1046.
16. R. P. Rocha, "Efficient Information Sharing and Coordination in Cooperative Multi-Robot Systems," *Proceedings of II European-Latin-American Workshop on Engineering Systems (SELASI'2006)*, Porto, Portugal (2006) pp. 1–6.
17. J. Hereford and M. Siebold, "Multi-robot search using a physically embedded particle swarm optimization," *Int. J. Comput. Intell. Res.* **4**(2), 197–209 (2008).
18. K. Shah and Y. Meng, "Communication-Efficient Dynamic Task Scheduling for Heterogeneous Multi-Robot Systems," *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL (2007) pp. 230–235.
19. O. Abedi, M. Fathy and J. Taghiloo, "Enhancing AODV routing protocol using mobility parameters in VANET," *IEEE/ACS International Conference on Computer Systems and Applications, (AICCSA 2008)*, Doha, Qatar (2008) pp. 229–235.
20. H. Asenov and V. Hnatyshin, "GPS-Enhanced AODV Routing," *Proceedings of the International Conference on Wireless Networks (ICWN'09)*, Las Vegas, NV (2009) pp. 1–7.
21. M. Ayash, M. Mikki and Y. Kangbin, "Improved AODV Routing Protocol to Cope with High Overhead in High Mobility MANETs," *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Palermo, Sicily (2012) pp. 244–251.
22. M. S. Couceiro, R. P. Rocha and N. M. F. Ferreira, "A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2011)*, Kyoto, Japan (2011) pp. 327–332.
23. J. Tillett, T. M. Rao, F. Sahin, R. Rao and S. Brockport, "Darwinian Particle Swarm Optimization," *Proceedings of the 2nd Indian International Conference on Artificial Intelligence* (2005) pp. 1474–1487.
24. J. Kennedy and R. Eberhart, "A New Optimizer Using Particle Swarm Theory," *Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science* (1995) pp. 39–43.
25. M. S. Couceiro, F. M. L. Martins, R. P. Rocha and N. M. F. Ferreira, "Introducing the Fractional Order Robotic Darwinian PSO," *Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (ICNPAA'2012)*, Vienna, Austria (2012) pp. 242–252.

26. M. S. Couceiro, J. A. T. Machado, R. P. Rocha and N. M. F. Ferreira, “A Fuzzified Systematic Adjustment of the Robotic Darwinian PSO,” *Robot. Auton. Syst.* **60**(12), 1625–1639 (2012).
27. I. Podlubny, *Fractional Differential Equations*, Mathematics in Science and Engineering, vol. 198 (Academic Press, San Diego, CA, 1999).
28. M. S. Couceiro, J. M. A. Luz, C. M. Figueiredo and N. M. F. Ferreira, “Modeling and control of biologically inspired flying robots,” *J. Robotica* **30**(1), 107–121 (2012).
29. W. Jatmiko, K. Sekiyama and T. Fukuda, “Modified particle swarm robotic odor source localization in dynamic environments,” *Int. J. Intell. Control Syst.* **11**(2), 176–184 (2006).
30. A. Marjovi and L. Marques, “Multi-robot olfactory search in structured environments,” *Robot. Auton. Syst.* **52**(11), 867–881 (2011).
31. L. E. Miller, “Multihop Connectivity of Arbitrary Networks,” *Project Report*, Wireless Communication Technologies Group (NIST), Gaithersburg, MD, webmaster@antd.nist.gov. (2001).
32. P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson and M. D. Erickson, “Enlisting rangers and scouts for reconnaissance and surveillance,” *IEEE Robot. Autom. Mag.* **7**(4), 14–24 (2000).
33. R. V. Kulkarni and G. K. Venayagamoorthy, “Bio-inspired algorithms for autonomous deployment and localization of sensor nodes,” *IEEE Trans. Syst. Man Cybern.* **40**(6), 663–675 (2010).
34. M. S. Couceiro, R. P. Rocha, C. M. Figueiredo, J. M. A. Luz and N. M. F. Ferreira, “Multi-Robot Foraging Based on Darwin’s Survival of the Fittest,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’2012)*, Vilamoura, Algarve (2012).
35. K. Natesapillai, V. Palanisamy and K. Duraiswamy, “A performance evaluation of proactive and reactive protocols using NS2 simulation,” *Int. J. Eng. Res. Ind. Appl.* **2**(11), 309–326 (2009).
36. S. J. Lee, M. Gerla and C. K. Toh, “A simulation study of table-driven and on-demand routing protocols for mobile ad hoc networks,” *Network. IEEE* **13**(4), 48–54 (1999).
37. F. Bertocchi, P. Bergamo, G. Mazzini and M. Zorzi, “Performance Comparison of Routing Protocols for Ad Hoc Networks,” *IEEE GLOBECOM*, San Fransisco, CA (2003) pp. 1033–1037.
38. X. Wu, H. Xu, H. R. Sadjadpour and J. J. Garcia-Luna-Aceves, “Proactive or Reactive Routing: A Unified Analytical Framework in MANETs,” *Proceedings of 17th International Conference on Computer Communications and Networks (ICCCN‘08)*, St. Thomas, VI (2008) pp. 1–7.
39. C. E. Perkins and E. M. Royer, “Ad Hoc On-Demand Distance Vector Routing,” **In**: *Mobile Computing Systems and Applications* (1999) pp. 90–100.
40. Digi International (Online) (2007). Available at: http://alumni.ipt.pt/∼lrafael/manual_XBee_Series2_OEM_RF-Modules_ZigBee.pdf. Accessed July 28, 2013.
41. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’98)*, Dallas, TX (1998) pp. 85–97.
42. M. S. Couceiro, F. M. L. Martins, R. P. Rocha and N. M. F. Ferreira, “Mechanism and convergence analysis of a multi-robot swarm,” *J. Intell. Robot. Syst.* (2013; In Press).
43. G. Beni, “From Swarm Intelligence to Swarm Robotics,” *Proceedings of the Swarm Robotics Workshop*, Heidelberg, Germany (2004) pp. 1–9.
44. University of Technology. (Online) (2001). Available http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot_en.html. Accessed July 28, 2013.
45. M. S. Couceiro, D. Portugal and R. P. Rocha, “A Collective Robotic Architecture in Search and Rescue Scenarios,” *Proceedings of the 28th Symposium On Applied Computing (SAC2013)*, Coimbra, Portugal (2013) pp. 64–69.
46. D. D. Luca, F. Mazzenga, C. Monti and M. Vari, “Performance Evaluation of Indoor Localization Techniques Based on RF Power Measurements from Active or Passive Devices,” *EURASIP J. Appl. Signal Process.* **2006**, 1–11 (2006).
47. B. Sklar, “Rayleigh fading channels in mobile digital communication systems. I. Characterization,” *IEEE Commun. Mag.* **35**(7), 90–100 (1997).
48. F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. C. Zufferey, D. Floreano and A. Martinoli, “The E-Puck – A Robot Designed for Education in Engineering,” *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* (2009) pp. 59–65.
49. W. Tsai, “Social structure of “coopetition” within a multiunit organization: Coordination, competition, and intraorganizational knowledge sharing,” *Organ. Sci*. **13**(2), 179–190 (2002).
50. E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems* (Oxford University Press, New York, NY, 1999).