2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Multi-Robot Foraging based on Darwin's Survival of the Fittest

Micael S. Couceiro, *IEEE Student Member*, Rui P. Rocha,
Carlos M. Figueiredo, J. Miguel A. Luz, N. M. Fonseca Ferreira

***Abstract* — This paper presents a collective foraging algorithm designed to simulate natural selection in a group of swarm robots. The Robotic Darwinian Particle Swarm Optimization (*RDPSO*) previously proposed is improved using fractional calculus theory and evaluated on real low-cost mobile robots performing a distributed foraging task. This work aims at evaluating this novel exploration strategy, by studying the performance of the algorithm within a population of up to 12 robots, under communication constraints. In order to simulate the maximum allowed communication distance, robots were provided with a list of their teammates' addresses. Experimental results show that only 4 robots are needed to accomplish the proposed mission and, independently on the number of robots, maximum communication distance and fractional coefficient, the optimal solution is achieved in approximately 90% of the experiments.**

I. Introduction

The cooperative foraging observed in nature has been studied for the last decades and used as inspiration in optimization problems, applied computation, robotics and many other applications [1]. This collective behavior emerges not only in hunting practices of lions, hyenas or wolves but also in small colonies of insects such as ants, bees, termites or wasps [2]. In spite of the potential advantages of Multi-Robot Systems (*MRS*) in cooperative foraging tasks, related with space and time distribution, it is necessary that each robot maintains a sufficient and consistent level of awareness about the mission assigned to the team and about its teammates in order to attain effective cooperation while avoiding collisions. Furthermore, in most real situations, robots have to move to complete their tasks while maintaining communication among them using ad-hoc communication, since communication infrastructure may be damaged or missing (*e.g.*, hostile environments, search & rescue, disaster recovery). Just like in *MRS* wherein groups of robots interact to accomplish their goals [3], particle optimization algorithms, such as the well-known Particle Swarm Optimization (*PSO*) [4] and the Darwinian Particle Swarm Optimization (*DPSO*) [5], use groups of interacting virtual agents in order to achieve their optimization. In a previous work [6], a multi-robot exploration algorithm was proposed and denoted as Robotic Darwinian Particle Swarm Optimization (*RDPSO*). This evolutionary algorithm was evaluated in simulation environment using Darwin's natural selection to enhance the ability to escape from sub-optimal solution. Also, an extension of the algorithm was proposed in [7] to guarantee Mobile Ad-Hoc Network (*MANET*) connectivity and to establish the initial deployment of robots based on the *Spiral of Theodorus*.

A. Statement of Contributions and Paper Organization

Bearing this idea in mind, a brief review of *PSO* techniques applied to *MRS* is presented in section II. This is followed by two key contributions demonstrated through an extension of the Robotic Darwinian *PSO* (*RDPSO*) previously presented in [6] and [7]. First, the concept of fractional calculus (*FC*) is used to control the convergence rate of the *RDPSO*, while considering the robot dynamical characteristics (section III). Secondly, an hierarchical control architecture of the *RDPSO* is presented where the Low-Level Control (*LLC*) handles all the kinematics and dynamics inherent to the robots (section IV). To assess these new features of the *RDPSO*, this novel distributed algorithm is evaluated in a cooperative foraging task performed by real mobile platforms (section V). The main conclusions are outlined in Section VI.

II. Related Work

In nature, Darwin's natural selection selects foraging behavior that maximizes lifetime calorie content or minimizes the probability of starvation. In robotics, engineering design favors decision-making algorithms that maximize accumulated value or minimize the probability of not reaching performance targets [8]. A behavior-based strategy to maintain *MRS* connectivity has been studied in [9]. The authors presented the extension of the Null-Space-based Behavioral approach to control a group of marine vehicles to execute missions such as formation control or cooperative target visiting with communication constraints. This is a promising approach since it would be possible to merge the behaviors of the *RDPSO* with different priorities in order to define the motion directives of robots. However, design choices concerning how to organize the behaviors by priority represent a higher complexity since these choices derive from practical considerations related to both the mission objective and the physical characteristics of the robotic system.

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

M. S. Couceiro and R. P. Rocha are with the Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal, e-mail: {micaelcouceiro,rprocha}@isr.uc.pt

C. M. Figueiredo J. Miguel A. Luz and N. M. F. Ferreira are with RoboCorp at the Electrical Engineering Department, Engineering Institute of Coimbra, Rua Pedro Nunes - Quinta da Nora, 3030-199 Coimbra, Portugal, e-mail: {cfigueiredo,miguel.luz,nunomig}@isec.pt

Similarly to the presented work, a multi-search algorithm inspired by the *PSO* algorithm is proposed in [10]. However, the algorithm was only evaluated based on *Webots* simulator and the search scenario was very limited, since only one target was used thus avoiding the effect of sub-optimal solution in the *PSO* algorithm. The work of Prasanna and Saikishan [11] involves path-planning and coordination between multiple mobile agents in a static-obstacle environment based on the *PSO* and Bacteria Foraging Algorithm (*BFA*). As this work uses Darwinian principles in the *PSO* to avoid getting trapped in sub-optimal solution, 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. Also, only one target and two robots were used thus limiting the evaluation of the proposed algorithm. Hereford and Siebold [12] presented an embedded version of the *PSO* in swarm platforms. As in *RDPSO*, there is no central agent to coordinate the 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. Also, collision avoidance and fulfillment of *MANET* connectivity was not considered.

Our previous works [6] [7] presented successive improvements of the proposed *RDPSO* to evaluate the convergence of the algorithm under communication constraints while increasing the robot population. Based on this previously proposed algorithm, next section extends the convergence rate of the *RDPSO* using fractional calculus.

III. Fractional Order RDPSO

In this section, a new method to control the convergence rate of the *RDPSO* algorithm [6] based on Pires *et al.* approach to the traditional *PSO* [13] is introduced. A previous work presented a similar approach conducted to the traditional *DPSO* in [14].

To model the swarm, each particle denoted by n moves in a multidimensional space. A particle is represented by the desired position vector x_{t+1}^n and velocity vector v_{t+1}^n which are highly dependent on the current position vector x_t^n, velocity vector (v_t^n), local best vector ($\check{c}_t^n$), global best vector ($\check{s}_t^n$), obstacle susceptibility vector ($\check{o}_t^n$), and *MANET* connectivity vector ($\check{m}_t^n$) information. The vectors size depends on the dimension of the multidimensional space.

$$v_{t+1}^n = wv_t^n + \rho_1 r_1(\check{c}_t^n - x_t^n) + \rho_2 r_2(\check{s}_t^n - x_t^n) + \rho_3 r_3(\check{o}_t^n - x_t^n) + \rho_4 r_4(\check{m}_t^n - x_t^n), \quad (1)$$

$$x_{t+1}^n = x_t^n + v_{t+1}^n, \quad (2)$$

where the coefficients w, ρ_1, ρ_2, ρ_3 and ρ_4 assign weights to the inertial influence, the local best (cognitive component), the global best (social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity, respectively. ters r_1, r_2, r_3 and r_4 are random vectors wherein each component is generally a uniform random number between 0 and 1. Considering equations (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 the other requirements (*i.e.*, avoid obstacles and maintain a certain distance between neighbors).

Fractional calculus (*FC*) has attracted the attention of several researchers [15], being applied in various scientific fields, such as engineering, computational mathematics, fluid mechanics, among others.

The *Grünwald–Letnikov* definition based on the concept of fractional differential with $\alpha \in C$ of the signal $x(t)$, is given by

$$D^{\alpha}[x(t)] = \lim_{h\to 0}\left[\frac{1}{h^{\alpha}}\sum_{k=0}^{+\infty}\frac{(-1)^k\Gamma(\alpha+1)x(t-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}\right], \quad (3)$$

where Γ is the gamma function. An important property revealed by (3) is that while an integer-order derivative just implies a finite series, the fractional-order derivative requires an infinite number of terms. Therefore, integer derivatives are "local" operators while fractional derivatives have, implicitly, a "memory" of all past events. However, the influence of past events decreases over time.

Based on (3), a discrete time implementation expression can be defined as:

$$D^{\alpha}[x(t)] = \frac{1}{T^{\alpha}}\sum_{k=0}^{r}\frac{(-1)^k\Gamma(\alpha+1)x(t-kT)}{\Gamma(k+1)\Gamma(\alpha-k+1)}, \quad (4)$$

where T is the sampling period and r is the truncation order.

The features inherent to fractional calculus make this mathematical tool well suited to describe many phenomena, such as irreversibility and chaos, because of its inherent memory property. In this line of thought, the dynamic phenomena of a robot's trajectory configure a case where fractional calculus tools fit adequately.

Considering the inertial influence in (1), $w = 1$, assuming $T = 1$ and based on [13] work, the following expression can be defined:

$$D^{\alpha}v_{t+1}^n = \rho_1 r_1(\check{c}_t^n - x_t^n) + \rho_2 r_2(\check{s}_t^n - x_t^n) + \rho_3 r_3(\check{o}_t^n - x_t^n) + \rho_4 r_4(\check{m}_t^n - x_t^n). \quad (5)$$

Preliminary experimental tests on the algorithm presented similar results for $r \geq 4$. Furthermore, the computational requirements increase linearly with r, *i.e.*, the fractional order *RDPSO* present a $\mathcal{O}(r)$ memory complexity. Hence, considering only the first four terms, $r = 4$, of differential derivative given by (4), equation (5) can be rewritten as:

$$\begin{aligned} v_{t+1}^n = {} & \alpha v_t^n + \frac{1}{2}\alpha v_{t-1}^n + \frac{1}{6}\alpha(1-\alpha)v_{t-2}^n + \\ & \frac{1}{24}\alpha(1-\alpha)(2-\alpha)v_{t-3}^n + \rho_1 r_1(\check{c}_t^n - x_t^n) + \\ & \rho_2 r_2(\check{s}_t^n - x_t^n) + \rho_3 r_3(\check{o}_t^n - x_t^n) + \rho_4 r_4(\check{m}_t^n - x_t^n). \end{aligned} \quad (6)$$

The *RDPSO* is therefore a particular case of the fractional order *RDPSO* for $\alpha = 1.0$ (without "memory"). Nevertheless, the value of α greatly affects the inertial of a robot. With a small value of α the robot will ignore its previous activities, thus ignoring the system dynamics (*i.e.*, exploita-

tion behavior). On the other hand, with a large value of α the robot may be unable to prevent collisions and maintain the connectivity with its teammates (*i.e.*, exploration behavior).

One way to analyze the convergence of the algorithm consists on adjusting the fractional coefficient based on physical mobile robots constraints inherent to their dynamical characteristics. Usually, this is not addressed in the literature concerning the classical *PSO* methods, since virtual agents (*i.e.*, particles) are not constrained by such behaviors. Hence, let us then suppose that a robot is traveling at a constant velocity such that $v_{t-k}^n = v$ with $k \in \mathbb{N}_0$ and it is able to find its equilibrium point in such a way that $x_t^n = \check{c}_t^n = \check{s}_t^n = \check{o}_t^n = \check{m}_t^n$. In other words, the robot converges to the best position such that the local, global, obstacle and *MANET* vectors are coincident. As a result, the robot needs to decelerate until it stops, *i.e.*, $v > v_{t-1}^n \geq \cdots \geq v_{t+k}^n \geq \cdots \geq 0$. Consequently, equation (6) can be rewritten as:

$$0 \leq v\left(\alpha + \frac{1}{2}\alpha + \frac{1}{6}\alpha(1-\alpha) + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)\right) < v, \tag{7}$$

thus resulting in

$$0 < \alpha \leq 0.632. \tag{8}$$

Therefore, one can conclude that $\alpha = 0.632$ is the boundary of the attraction domain: The *RDPSO* is stable for $0 < \alpha \leq 0.632$ and unstable for $0.632 < \alpha \leq 1$. As a result of the above analysis, the fractional coefficient can be parameterized 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.

Next section presents the control architecture of the *RDPSO* considering real heterogeneous mobile robots.

IV. Control Architecture of Swarm Robots

This section presents how the *RDPSO* algorithm can be used on real robotic platforms. The *RDPSO* performs all the computation by considering the robots as particles, thus ignoring any kinematic or dynamic inherent to the real platform. In other words, the output is a position vector for each robot (*cf.* equations 2 and 6). Depending on the robot kinematical and dynamical characteristics, this new position may be achieved from different ways. For instance, a holonomic robot (*e.g.*, omnidirectional drive system), which has the kinematic advantage of allowing continuous translation and rotation in any direction, can move to a desired position regardless of its orientation. However, a non-holonomic robot (*e.g.*, differential drive system) needs to first change its orientation to be aligned with the target and then move forward to the desired position.

In order to achieve a higher level of hardware abstraction, a Low-Level Control (*LLC*) was introduced in the control architecture (Fig. 1). In this work, and since all robots are non-holonomic, the *LLC* was designed for the specific structure of the differential-drive robot to make it turn and then follow the desired position vector received by the *RDPSO*.

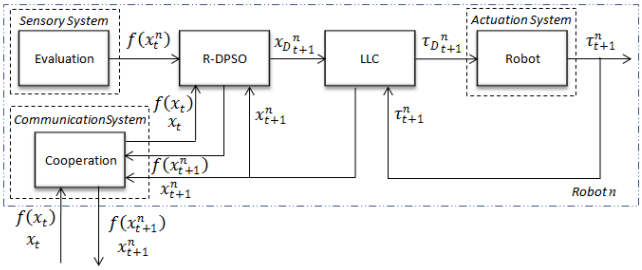

Fig. 1. Control architecture of the *RDPSO* with LLC of a robot *n*.

As shown in Fig. 1, the output of the *RDPSO* is given as a reference value to the *LLC* that considers the kinematical and dynamical characteristics of the robot thus defining the commands to the actuators, *e.g.*, number of pulses for *DC* motors equipped with encoders.

This two-level control loop organization allows using the *RDPSO* with different types of robotic systems, neglecting their kinematical and dynamical characteristics. These aspects are considered by the *LLC* that needs to be properly designed for a specific robotic system.

To our specific situation, *i.e.*, differential-drive robots, the *LLC* receives the desired position $x_{D_{t+1}}^n$, which corresponds to a cartesian position $[x_{D1}{}_{t+1}^n \quad x_{D2}{}_{t+1}^n]^T$, and computes the inverse kinematic model based on the following equations:

$$h_{t+1}^n = \sqrt{\left(v_{1}{}_{t+1}^n\right)^2 + \left(v_{2}{}_{t+1}^n\right)^2}, \tag{9}$$

$$\sigma_{t+1}^n = atan2\left(v_{2}{}_{t+1}^n, v_{1}{}_{t+1}^n\right), \tag{10}$$

$$\theta_{t+1}^n = \sigma_{t+1}^n - \sigma_t^n, \tag{11}$$

wherein $v_{1}{}_{t+1}^n$ and $v_{2}{}_{t+1}^n$ are obtained using (6). The function *atan2* in (10) is a variant of the trigonometric arctangent function, but accounts for the quadrant in which σ_{t+1}^n lies. Note that, σ_t^n corresponds to the previous computation of (9) and may be considered zero in the first iteration (*i.e.*, initial orientation of zero degrees). The output of the inverse kinematic model is represented by the rotation θ_{t+1}^n that the robot needs to perform to be aligned with the desired position and the distance h_{t+1}^n it needs to travel to reach it.

The rotation $\tau_{D1}{}_{t+1}^n$ and the forward movement $\tau_{D2}{}_{t+1}^n$ of the differential-drive robot are defined by:

$$\tau_{D1}{}_{t+1}^n = \tau_{rev} . \frac{\theta_{t+1}^n}{2\pi} . \frac{r_{robot}}{r_{wheel}}, \tag{12}$$

$$\tau_{D2}{}_{t+1}^n = \tau_{rev} . \frac{h_{t+1}^n}{2\pi} . \frac{1}{r_{wheel}}, \tag{13}$$

wherein τ_{rev} is the total number of steps or pulses per revolution. The radius of the robot and the wheels are defined by r_{robot} and r_{wheel}, respectively. In order to improve the time response of the robot and the smoothness of its movement, a rotational threshold θ_T was introduced. Rotations θ_{t+1}^n inferior to θ_T are then ignored and only the forward distance h_{t+1}^n is considered. Bearing in mind this assumption, and since a possible loss of steps or pulses may occur while executing the commands, *i.e.*, $\tau_{1}{}_{t+1}^n \neq \tau_{D1}{}_{t+1}^n$ or $\tau_{2}{}_{t+1}^n \neq \tau_{D2}{}_{t+1}^n$, a new real position is then recalculated and consid-

ered as the current position x_{t+1}^n of the robot. This new position and the corresponding value of the objective function $f(x_{t+1}^n)$ defined in this position (*i.e.*, sensed by the sensory system) needs to be shared between robots (*cf.*, *Communication System* in Fig. 1) so that cooperation can emerge. To that end, this information is sent directly to the robots in the neighborhood defined by d_{max} and relayed to other robots based on ad-hoc networking paradigm.

V. Experimental Results

In this section, it is explored the effectiveness of using the *RDPSO* on swarms of real robots, while performing a collective foraging task with local and global information under communication constraints. Multiple test groups of 20 trials of 3 minutes each were considered. A minimum, initial and maximum number of 1, 2 and 3 swarms were used independently of the population of robots.

The *eSwarBot* (Educational Swarm Robot) was the platform used to evaluate the algorithm (Fig. 2). It consists on a differential ground platform recently developed and presented in [16] for applications in swarm robotics. Although the platform presents a limited kinematic resolution of 3.6 degrees while rotating and 2.76 mm when moving forward, its low cost and high autonomy allowed to perform experiments with up to 12 robots, with $N = \{4, 8, 12\}$. *RGB-LED*s on top of the *eSwarBots* are used to identify its swarm. *RGB-LED*s allow the representation of a wide range of different colors matching different swarms. All of the experiments were carried out in a 2.55 meters to 2.45 meters scenario.

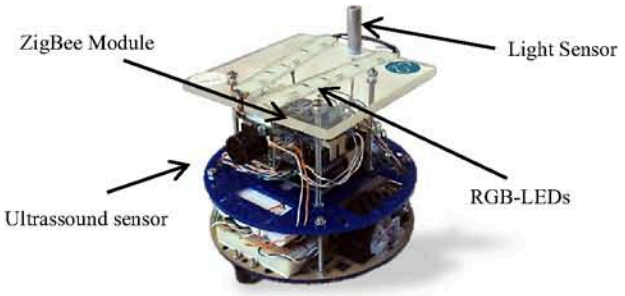

Fig. 2. The *eSwarBot* differential ground platform.

The experimental environment (Fig. 3a) was an enclosed arena that contained two sites represented by illuminated spots uniquely identifiable by controlling the brightness of the light. Despite being an obstacle free scenario, the robots themselves act as dynamic obstacles –a maximum number of 12 robots correspond to a population density of approximately 2 robots.m^{-2}. Each robot possessed overhead light sensors (*LDR*) that allowed it to find candidate sites and measure their quality. The brighter site (optimal solution) was considered better than the dimmer one (sub-optimal solution), and so the goal of the robots was to collectively choose the brighter site. The intensity values $F(x, y)$ represented in Fig. 3b were obtained sweeping the whole scenario with a single robot in which the light sensor was connected to a 10-bit analog input. To improve the interpretation of the algorithm performance, results were normalized in a way that the objective of robotic teams is to find the optimal solution of $f(x, y) = 1$.

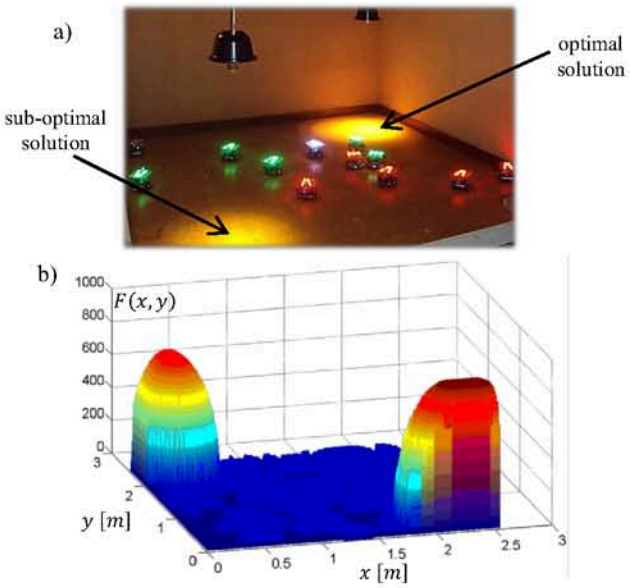

Fig. 3. Experimental setup. a) Enclosed arena with 2 swarms (different colors); b) Virtual representation of the target distribution.

Inter-robot communication to share positions and local solutions were carried out using *ZigBee* 802.15.4 wireless protocol. Since robots were equipped with *XBee* modules that allow a maximum communication range larger than the whole scenario, robots were provided with a list of their teammates' address in order to simulate the ad-hoc multi-hop network communication with limited range. The maximum communication distance between robots d_{max} was varied between 0.50 meters and 1.50 meters. At each trial, robots were manually deployed on the scenario in a spiral manner while preserving the maximum communication distance d_{max} (as previously presented in [7]).

In order to evaluate the impact of fractional calculus in the convergence of the algorithm, the original *RDPSO* ($\alpha = 1.0$) was compared to the fractional order *RDPSO* with $\alpha = 0.632$ (*cf.*, Section III).

The previously described conditions give a total of 240 experiments, thus leading to a runtime of 12 hours. The next sequence of frames (Fig. 4) presents a trial of the team's performance using $N = 12$, $d_{max} = 1.50$ meters and $\alpha = 0.632$. Since these experiments represent a foraging task, it is necessary to evaluate both the completeness of the mission and the time needed to complete it. Therefore, Fig. 5 depicts the convergence of the *RDPSO* for the several proposed conditions. The median of the best solution in the 20 experiments was taken as the final output in the set $N = \{4, 8, 12\}$ for each d_{max} and α.

Analyzing Fig. 5, it is clear that the proposed mission can be accomplished by any number of robots between 4 and 12. In fact, independently on the number of robots, teams converge to the solution in approximately 90% of the experiments. The charts also show that increasing the number of robots slightly increases the convergence rate. A population of 4, 8 and 12 robots takes, in average, 77, 106 and 112 seconds to converge to the optimal solution, respectively. This is a consequence of having more robots inside the same arena – the number of dynamic obstacles is higher.

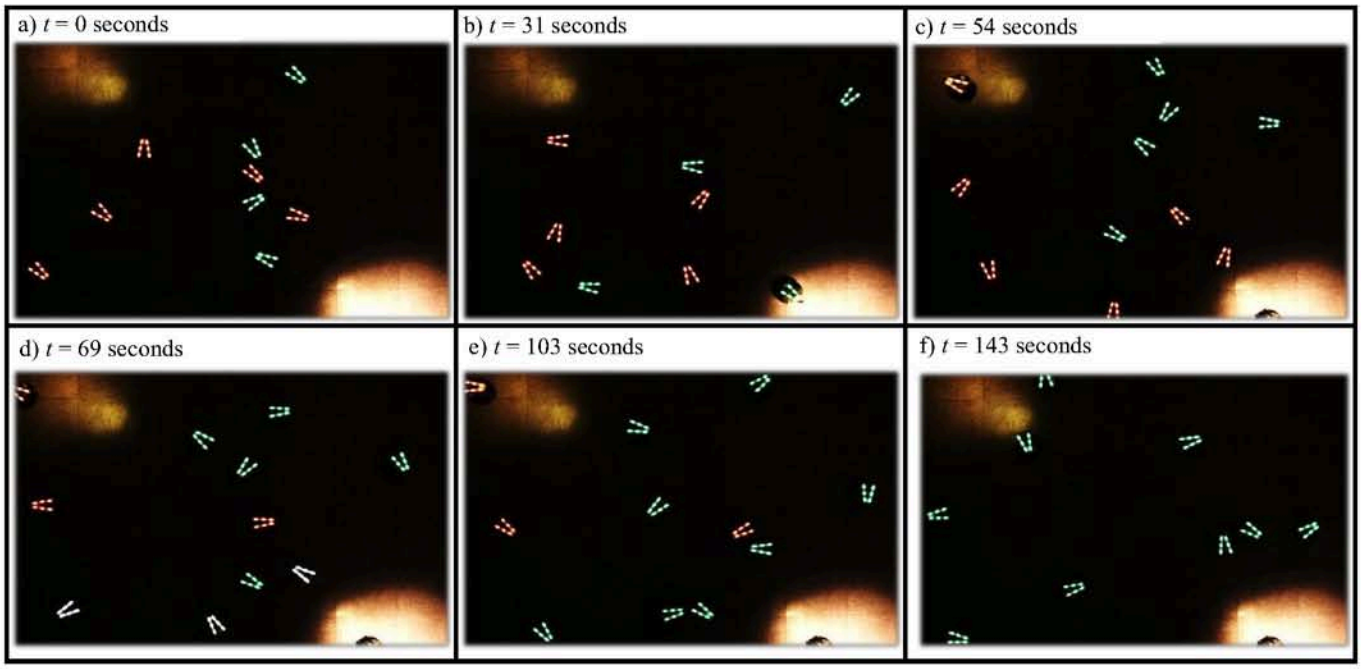

Fig. 4. Frame sequence showing the *RDPSO* performance on a population of 12 robots (some robots may be outside camera's range). a) The population is initially divided into two swarms – green and red – deployed in a spiral manner; b) The swarms independently search for the brighter site taking into account a maximum communication distance of 1.5 meters between robots of the same swarm; c) One robot from the red and green swarm finds the sub-optimal and optimal solution, respectively; d) As the red swarm does not improve, some robots are excluded, thus being added to the socially excluded group (white swarm); e) Since the green swarm has improved, it is able to call new members from the socially excluded group; f) Finally, the green swarm proliferates calling all the previously excluded robots that were unable to improve their solution. Note that robots do not all converge the optimal solution as they try to maintain a distance of d_{max} between them.

As expected, increasing the maximum communication distance generally results in a faster convergence to the optimal solution. On the other hand, the decrease of α from 1 to 0.632 improves the convergence rate of the algorithm in some situations also marginally improving the median value of the solution. However, this relationship is not linear and varies depending on the maximum communication distance and the number of robots in the population. For instance, for a communication distance of 0.5 meters and regardless on the number of robots, the fractional order *RDPSO* depicts an improved convergence when compared to the integer order. 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 used in these robots, is a classical method, being of low-cost and simple use. However, it is verified the existence of errors inherent to their use are cumulative, which makes it difficult for the robots to complete the proposed odometry objectives accurately.

VI. Conclusion

This paper presents the evaluation of a foraging sociobiological-inspired algorithm, denoted as *RDPSO* - Robotic Darwinian Particle Swarm Optimization, on real platforms. Furthermore, fractional calculus is introduced to control the convergence rate of the algorithm taking into account robot dynamics. Experimental results show that the performance of the robotic population in the proposed scenario, with a biologically inspired behaviour based on natural selection and social exclusion, is not significantly affected by the number of robots in the population, the maximum communication distance between robots and the fractional coefficient α. Yet, the fractional extension of the *RDPSO* allows taking into account the dynamic phenomena of robots' trajectory. Nevertheless, one of the main limitations of the algorithm resides in having several parameters influencing the performance of the robotic team. Therefore, one of the future approaches will be the analytical analysis of the *RDPSO* in order to find a relationship between parameters, thus optimizing the algorithm with regard to the main objective, robot dynamics, obstacles susceptibility and *MANET* connectivity. Also, *eSwarBot* platforms should be enhanced in order to avoid mislocalization errors. The robots odometry could be improved by replacing the current encoders with more precise and with better resolution ones, keeping the low-cost goal. Nevertheless, this exchange may not be enough by itself. There are other methods that can be used with the current odometry in order to overcome these problems by correcting the errors in robot location based on reference points in the scenario. For instance, analysing the strength of the received signal (*RSSI*) [17] (taking advantage of this *XBee* modules' capability) would allow determining the location of mobile module through triangulation methods.

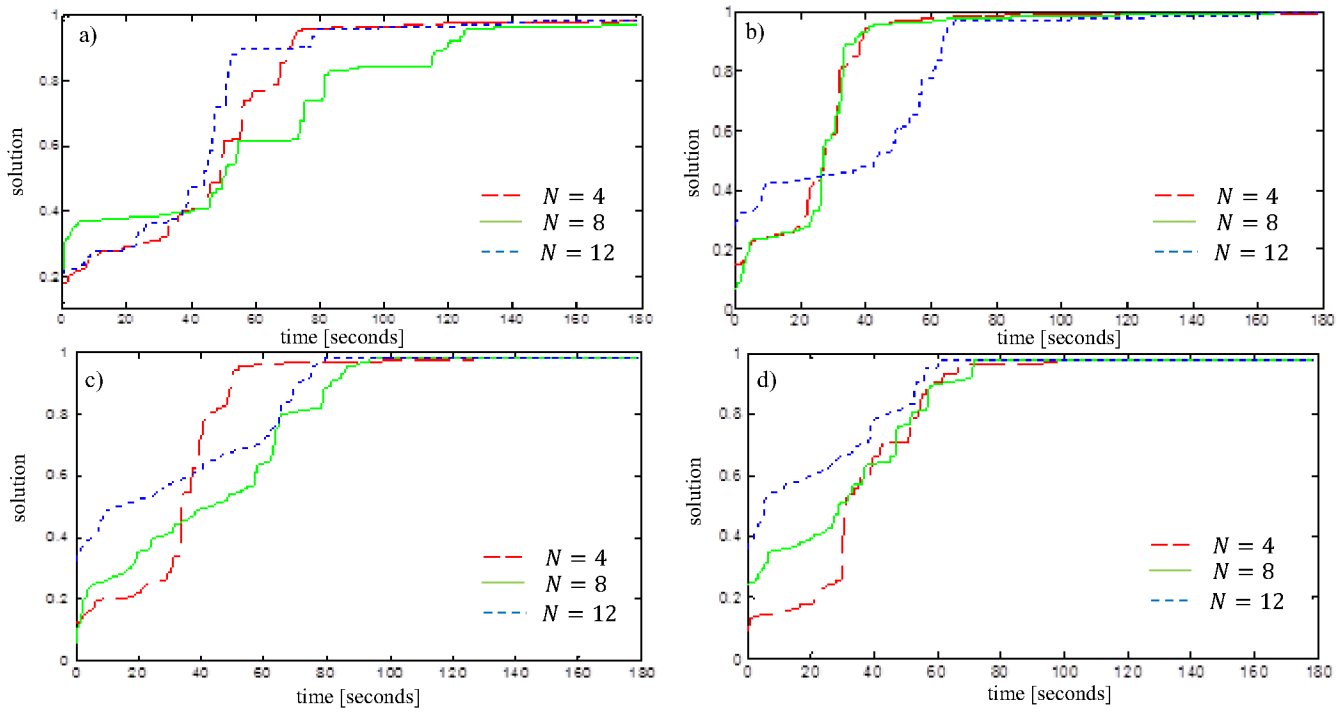

Fig. 5. Performance of the algorithms changing the number of robots *N* in the population: a) $d_{max} = 0.5\ meters;\ \alpha = 1.0$; b) $d_{max} = 1.5\ meter;\ \alpha = 1.0$; c) $d_{max} = 0.5\ meters;\ \alpha = 0.632$; d) $d_{max} = 1.5\ meter;\ \alpha = 0.632$.

REFERENCES

[1] K. M. Passino, *Biomimicry for Optimization, Control, and Automation*. London: Springer-Verlag, 2005.

[2] J. A. Newman, *Foraging: Behaviour and Ecology*, D. W. Stephens, J. C. Brown, and R. C. Ydenberg, Eds. Chicabo: The University of Chicago Press, 2006.

[3] R. Rocha, J. Dias, and A. Carvalho, "Cooperative Multi-Robot Systems: a study of Visionbased 3-D Mapping using Information Theory," *Robotics and Autonomous Systems*, vol. 53(3–4), pp. 282-311, 2005.

[4] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," in *Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.

[5] 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*, pp. 1474-1487, 2005.

[6] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms," in *in proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011*, Kyoto, Japan, 2011, pp. 327-332.

[7] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms," in *in proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011*, Kyoto, Japan, 2011, pp. 284-289.

[8] T. P. Pavlic and K. M. Passino, "Generalizing foraging theory for analysis and design," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 505-523, 2011.

[9] F. Arrichiello, S. Chiaverini, and T. I. Fossen, "Formation control of marine surface vessels using the Null-Space-based Behavioral Control," *Group Coordination and Cooperative Control*, pp. 1-19, 2006.

[10] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.

[11] D. Saikishan and K. Prasanna, "Multiple robot co-ordination using particle swarm optimisation and bacteria foraging algorithm," Department of Mechanical Engineering, National Institute of Technology BTech thesis, 2010.

[12] J. Hereford and M. Siebold, "Multi-robot search using a physically-embedded Particle Swarm Optimization," *International Journal of Computational Intelligence Research*, vol. 4, no. 2, p. 197–209, 2008.

[13] E. J. S. Pires, J. A. T. Machado, P. B. M. Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Journal on Nonlinear Dynamics*, vol. 61, p. 295–301, 2010.

[14] M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the Fractional Order Darwinian PSO," *Signal, Image and Video Processing, Springer*, no. Fractional Signals and Systems Special Issue, 2012.

[15] J. Sabatier, O. P. Agrawal, and J. A. T. Machado, *Advances in Fractional Calculus - Theoretical Developments and Applications in Physics and Engineering*. Berlin: Springer, 2007.

[16] 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," *International Journal of Robots, Education and Art*, 2011.

[17] Y.-T. Chen, C.-L. Yang, Y.-K. Chang, and C.-P. Chu, "A RSSI-based Algorithm for Indoor Localization Using ZigBee in Wireless Sensor Network," in *Proceedings of the 15th International Conference on Distributed Multimedia Systems (DMS 2009)*, San Francisco, USA, 2009, pp. 70-75.