International Mathematical Forum, Vol. 7, 2012, no. 32, 1587 - 1601

Analysis and Parameter Adjustment of the *RDPSO* Towards an Understanding of Robotic Network Dynamic Partitioning based on Darwin's Theory

Micael S. Couceiro[123]

Department of Electrical and Computer Engineering,
University of Coimbra, Polo II
3030-290, Coimbra, Portugal
micaelcouceiro@isr.uc.pt

Fernando M. L. Martins[3]

Instituto de Telecomunicações (Covilhã),
Coimbra College of Education,
Rua Dom Joao III - Solum
3030-329 Coimbra, Portugal
fmlmartins@esec.pt

Rui P. Rocha[2]

Department of Electrical and Computer Engineering,
University of Coimbra, Polo II
3030-290, Coimbra, Portugal
rprocha@isr.uc.pt

N. M. Fonseca Ferreira[4]

RoboCorp, Department of Electrical Engineering,
Engineering Institute of Coimbra, Coimbra, Portugal
nunomig@isec.pt

Abstract

[1]Corresponding author

[2]Member of the Institute of Systems and Robotics, University of Coimbra, Portugal

[3]Member of the RoboCorp, DEE, Engineering Institute of Coimbra, Portugal

[4]Member of the Knowledge Engineering and Decision Support Research Center, Porto Portugal.

Although the well-known Particle Swarm Optimization (*PSO*) algorithm has been first introduced more than a decade ago, there is a lack of methods to tune the algorithm parameters in order to improve its performance. An extension of the *PSO* to multi-robot foraging has been recently proposed and denoted as Robotic Darwinian *PSO* (*RDPSO*), wherein sociobiological mechanisms are used to enhance the ability to escape from local optima. This novel swarm algorithm benefits from using multiple smaller networks (one for each swarm), thus decreasing the number of nodes (*i.e.*, robots) and the amount of information exchanged among robots belonging to the same sub-network. This article presents a formal analysis of *RDPSO* in order to better understand the relationship between the algorithm's parameters and its convergence. Therefore, a stability analysis and parameter adjustment based on acceleration and deceleration states of the robots is performed. These parameters are evaluated in a population of physical mobile robots for different values of communication range. Experimental results show that, for the proposed mission and parameter tuning, the algorithm con-verges to the global optimum in approximately 90% of the experiments regardless on the number of robots and the communication range.

Mathematics Subject Classification: 39A30, 70E60, 65L20

Keywords: foraging, parameter adjustment, stability analysis

1 Introduction

The principles of self-organization of biological simple creatures (*i.e.*, bees, ants) are appealing for explaining biological collective phenomena where the resulting structures and functionalities greatly exceed in complexity the perceptual, physical, and cognitive abilities of the participating organisms. The Particle Swarm Optimization (*PSO*) developed by Kennedy and Eberhart [10] is an optimization technique that models a set of potential problem solutions as a swarm of particles moving around in a virtual search space. However, a general problem with the *PSO* and other optimization algorithms is that of becoming trapped in a local optimum, such that it may work in some problems but may fail on others. In search of a better model of natural selection using the *PSO* algorithm, the Darwinian Particle Swarm Optimization (*DPSO*) was formulated by Tillet et al. [12]. In this algorithm, multiple swarms of test solutions performing just like an ordinary *PSO*, may exist at any time with some rules governing the collection of swarms that are designed to simulate natural selection.

However, regardless of *PSO* main variants [4], the difficulties in setting and adjusting the parameters, as well as in maintaining and improving the search capabilities for higher dimensional problems, is still a matter that has been addressed in recent works [2, 9, 13]. One of the most common methods presented in the literature to resolve issues in setting and adjusting *PSO* parameters is based on the stability analysis of the algorithm.

In [2], it is analyzed the individual particle's trajectory leading to a generalized model of the algorithm, which contains a set of coefficients to control the system's convergence tendencies. The resulting system was a second-order linear dynamical system whose stability and parameters depended on the system poles or the eigenvalues of the state matrix. Alternatively, Kadirkamanathan et al. [9] proposes a stability analysis of a stochastic particle dynamics by representing it as a nonlinear feedback controlled system. The Lyapunov stability method was applied to the particle dynamics in determining sufficient and conservative conditions for asymptotic stability. However, the analysis provided by the authors has addressed only the issue of absolute stability, thus ignoring the optimization convergence to the global optimum.

More recently, Yasuda et al. [13] presented an activity-based numerical stability analysis method, which involves the feedback of swarm activity to control diversification and intensification during the search. The authors show that swarm activity can be controlled by employing the stable and unstable regions of *PSO*. However, in a distributed approach, such as the *RDPSO*, calculating the swarm activity would imply that each robot shares its current velocity with all other members.

Contrarily to the herein proposed multi-robot foraging approach, all previously presented works only consider *PSO* and its main variants applied to optimization problems. Contrarily to virtual agents (*i.e.*, particles), robots are designed to act in the real world where their dynamical characteristics and obstacles need to be taken into account. Also, and since that in certain environments or applications, such as hostile environments, search and rescue, disaster recovery, battlefields, space and others, the communication infrastructure may be damaged or missing, the self-spreading of autonomous mobile nodes of a mobile ad-hoc network (*MANET*) over a geographical area needs to be considered.

Bearing this idea in mind, this paper presents a stability analysis of the *RDPSO* which allows obtaining the attraction domain. Moreover, in order to improve the convergence and performance of the algorithm, a method for obtaining a parameter adjustment inside this domain is carried out taking into account the acceleration and deceleration states of the robots.

A brief review of the *RDPSO* algorithm is given in section 2. Section 3 presents the stability analysis and parameter adjustment of the algorithm. A population of real robots is used to evaluate the performance of the algorithm in Section 4. Finally, Section 5 outlines the main conclusions.

2 *RDPSO*

This section briefly presents the *RDPSO* algorithm proposed in [3]. Since the *RDPSO* approach is an adaptation of the *DPSO* to real mobile robots, five general features are proposed: *i)* an improved inertial influence based on fractional calculus concept taking into account convergence dynamics; *ii)* an obstacle avoidance behaviour to avoid collisions; *iii)* an algorithm to ensure that the *MANET* remains connected throughout the mission; *iv)* a novel methodology to establish the initial planar deployment of robots preserving the connectivity of the *MANET* while spreading out the robots as most as possible; and *v)* a novel "punish"-"reward" mechanism to emulate the deletion and creation of robots. The behaviour of robot n can then be described by the following discrete equations at each discrete time, or iteration, $t \in \mathbb{N}_0$:

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

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

where coefficients $\rho_i, i = 1, 2, 3, 4$, assign weights to the inertial influence, the local best (cognitive component), the local best (social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity, with $\rho_i > 0$. Parameters r_i are random matrices where in each component is generally a uniform random number between 0 and 1. $v_n[t]$ and $x_n[t]$ represents the velocity and position vector of robot n, respectively. $\chi_i[t]$ represents the best position of the cognitive, social, obstacle and *MANET* matrix components. The cognitive $\chi_1[t]$ and social components $\chi_2[t]$ are the commonly presented in the classical *PSO* algorithm. $\chi_1[t]$ represents the local best position of robot n while $\chi_2[t]$ represents the global best position of robot n. Since the other features are novel, they are further explored in the following sections.

2.1 Fractional Order Convergence

Fractional calculus (FC) has attracted the attention of several researchers, being applied in various scientific fields such as engineering, computational mathematics, fluid mechanics, among others [3, 11]. One of the most common approaches based on the concept of fractional differential, is the discrete time *Grï¿½nwald-Letnikov* definition given by the equation:

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

for the velocity $v_n[t+1]$ with $0 < \alpha < 1$. In the common *PSO* algorithm, the inertial component $w_n[t]$ is usually proportional to the inertial influence. Based on equation (1) and (3), considering $T = 1$ and $r = 4$, the inertial component of robot n can be defined as:

$$w_n[t] = \alpha v_n[t] + \frac{1}{2}\alpha v_n[t-1] + \frac{1}{6}\alpha(1-\alpha)v_n[t-2] + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)v_n[t-3]. \quad (4)$$

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

2.2 Obstacle Avoidance

A new cost or fitness function is defined in such a way that it would guide the robot to perform the main mission while avoiding obstacles. For this purpose it is assumed that each robot is equipped with sensors capable of sensing the environment for obstacle detection within a finite sensing radius r_s. A monotonic and positive *sensing function* $g(x_n[t])$ that depends on the sensing information (*i.e.*, distance from the robot to obstacle) is defined. In most situations $g(x_n[t])$ can be represented as the relation between the analog output voltage of distance sensors and the distance to the detected object.

$\chi_3[t]$ is then represented by the position of each robot that optimizes the monotonically decreasing or increasing $g(x_n[t])$ and its current position. In a free-obstacle environment, the obstacle susceptibility weight ρ_3 is set to zero. However, in real-world scenarios, obstacles need to be taken into account and the value of ρ_3 depends on several conditions related with the main objective (*i.e.*, minimize a cost function or maximize a fitness function) and the sensing information (*i.e.*, monotonicity of $g(x_n[t])$). Furthermore, the relation between ρ_3 and the other weights depends on the susceptibility of each robot to obstacle avoidance behavior.

2.3 Ensuring *MANET* Connectivity

Robots' position need to be controlled in order to maintain the communication based on constraints such as maximum distance or minimum signal quality. The way network will be forced to preserve connectivity depends on communication characteristics (*e.g.*, multi-hop, biconnectivity). Assuming that the network supports multi-hop connectivity, the communication between two end nodes (*i.e.*, robots) is carried out through a number of intermediate nodes whose function is to relay information from one point to another (note that any robot may be used as a relay node independently of their swarm). Considering that nodes are mobile, it is necessary to guarantee the communication between all nodes. In the case where each robot corresponds to a node, in order to overcome the non-connectivity between them, the desired position, *i.e.*, $x_n[t+1]$, must be controlled since it influences the adjacency matrix A. The adjacency matrix, on the other hand, may depend on the maximum communication range or minimum signal quality. One way to ensure the full connectivity of the *MANET* is to "force" each robot to communicate with its nearest neighbor that has not chosen it as its nearest neighbor. Since the connectivity depends on the distance/signal quality, connectivity between nodes may be ensured by computing the minimum/maximum value of each line of link matrix L, after excluding zeros and (i, j) pairs previously chosen. Therefore, the *MANET* component $\chi_4[t]$ is represented by the position of the nearest neighbor increased by the maximum communication range d_{max} toward robot's current position. A higher ρ_4 may enhance the ability to maintain the network connected ensuring a specific range or signal quality between robots.

2.4 Initial Deployment

This approach tries to get the benefits of a random planar deployment of robots while eliminating the disadvantages inherent to it. Furthermore, the herein proposed approach takes into account the communication constraints using a deployment strategy based on the *Spiral of Theodorus* (aka, square root spiral). This spiral is composed of contiguous right triangles (formerly called rectangled triangles) with each cathetus (aka, leg) having a unit length of 1 [8]. Each of the triangle's hypotenuses gives the square root to a consecutive natural number.

Since this approach uses the spiral of Theodorus to carry out the initial deployment of robots, two general adjustments need to be considered: *i)* the initial position of each robot is set at the further vertex of the centre of the spiral for each right triangle with a random orientation and also a random swarm; and *ii)* the size of the cathetus is set as the maxi-mum communication range (instead of having the unit length 1) consequently changing the triangles' hypotenuses to the product between the maximum communication range and the square root of the consecutive natural number. These assumptions make it possible to have an initial deployment of the robots in an area that depends on both the number of robots and the communication constraints.

2.5 Punish-Reward Mechanism

In the common *DPSO*, "punish" means the deleting of particles and swarms, while "reward" means the spawning of new particles and swarms. In order to adapt *DPSO* to mobile robotics, the deleting and spawning of a robot are modelled by the mechanisms of social exclusion and social inclusion, respectively. The *RDPSO* is then represented by multiple swarms, *i.e.*, multiple groups of robots that altogether form a population. Each swarm individually performs just like a *PSO* adapted to multi-robot applications (explained in the above subsections) in search for the solution and some rules governs the whole population of robots. If there was no improvement in a swarm's objective over a period of time, the swarm is punished by excluding the worst performing robot, which is added to a socially excluded group. The worst performing robot is evaluated by the value of its objective function compared to other members in the same swarm. In other words, if the objective is to maximize the fitness function, the robot to be excluded will be the one with the lower fitness value. Those socially excluded robots, instead of searching for the objective function's global optimum like the other robots in the active swarms, they basically randomly wander in the scenario. This approach improves the algorithm, making it less susceptible of becoming trapped in a local optimum. Note, however, that they are always aware of their individual solution and the global solution of the socially excluded group. Having multiple swarms enables a distributed approach be-cause the network that was previously defined by the whole population of robots is now divided into multiple smaller networks (one for each swarm,) thus decreasing the number of nodes (*i.e.*, robots) and the information exchanged between robots of the same network. In other words, robots interaction with other robots through communication is confined to local interactions inside the same group (swarm), thus making *RDPSO* scalable to large populations of robots.

3 Convergence Analysis

The above presented *RDPSO* is a stochastic procedure in which (1) describes the discrete-time motion of a robot with four external inputs $\chi_i[t]$. The main problem when analyzing this kind of algorithms lies in the fact that external inputs vary in time. However, one can consider that each robot converges to an *equilibrium point* defined by the limit values of the attractor points χ_i. Therefore, assuming that the algorithm converges, this section presents the stability analysis of the *RDPSO* and the parameter adjustment inherent to the dynamical characteristics of robots.

3.1 Problem Formulation

Consider a population of N robots wherein each robot needs to cooperatively find the optimal solution of a given mission within its swarm. The goal is to find the attraction domain A such that, if coefficients $\alpha, \rho_i \in \mathcal{A}, i = 1, 2, 3, 4$, the global asymptotic stability of the system (1) is guaranteed, thus allowing robots to find the optimal solution while avoiding obstacles and ensuring *MANET* connectivity.

3.2 General Approach

Since $v_n[t-k] = x_n[t-k] - x_n[t-(k+1)]$ with $k \in \mathbb{N}_0$, equations (1) and (2), can be rewritten as a nonhomogeneous five-order difference equation:

$$\begin{aligned} x_n[t+1] + \left(-1-\alpha+\textstyle\sum_{i=1}^{4}\rho_i r_i\right) x_n[t] + \left(\tfrac{1}{2}\alpha\right) x_n[t-1] + \\ + \left(\tfrac{1}{3}\alpha+\tfrac{1}{6}\alpha^2\right) x_n[t-2] + \left(-\tfrac{1}{24}\alpha^3-\tfrac{1}{24}\alpha^2+\tfrac{1}{12}\alpha\right) x_n[t-3] \\ + \left(\tfrac{1}{24}\alpha^3-\tfrac{1}{8}\alpha^2+\tfrac{1}{12}\alpha\right) x_n[t-4] = \textstyle\sum_{i=1}^{4}\rho_i r_i \chi_i[t] \end{aligned} \tag{5}$$

The *equilibrium point* x_n^* can be defined as a *constant position solution* of (5), such that, when each robot reaches x_n^*, the velocity $v_n[t+k]$ is zero, *i.e.*, robots will stop at the equilibrium point x_n^*. Supposing that χ_i are constants, *i.e.*, the algorithm does converge, the particular solution of each robot can be defined as [7]:

$$x_n^* = \frac{\sum_{i=1}^{4}\rho_i r_i \chi_i}{\sum_{i=1}^{4}\rho_i r_i}. \tag{6}$$

In other words, each robot will converge to the particular solution x_n^*0, based on the following theorems [7]:

Theorem 3.1 *All solutions of (5) converge to x_n^* as $k \to \infty$, if and only if the homogeneous difference equation of (5) is asymptotically stable.*

Theorem 3.2 *The homogeneous difference equation of (5) is asymptotically stable if and only if all roots of the corresponding characteristics equation have modulus less than one.*

Due to the complexity in obtaining the roots of the characteristics equation of homogeneous difference equation (5), it is established a result that ensures that all roots of the real polynomial $p(\lambda)$ have modulus less than one.

$$\begin{aligned} p(\lambda) \equiv \lambda^5 + \left(-1-\alpha+\textstyle\sum_{i=1}^{4}\rho_i r_i\right)\lambda^4 + \left(\tfrac{1}{2}\alpha\right)\lambda^3 + \left(\tfrac{1}{3}\alpha+\tfrac{1}{6}\alpha^2\right)\lambda^2 + \\ \left(-\tfrac{1}{24}\alpha^3-\tfrac{1}{24}\alpha^2+\tfrac{1}{12}\alpha\right)\lambda + \left(\tfrac{1}{24}\alpha^3-\tfrac{1}{8}\alpha^2+\tfrac{1}{12}\alpha\right) = 0. \end{aligned} \tag{7}$$

Proposition 3.1 *All roots of $p(\lambda)$ have modulus less than one if and only if the following conditions are met.*

$$\begin{cases} 0 < \sum_{i=1}^{4}\rho_i r_i \leq \alpha+2 & ,0 < \alpha \leq 0.6 \\ \frac{15}{4}\alpha - \frac{9}{4} < \sum_{i=1}^{4}\rho_i r_i \leq \alpha+2 & ,0.6 < \alpha \leq 1 \end{cases}. \tag{8}$$

Proof: The real polynomial $p(\lambda)$ described in equation (7) can be rewritten as:

$$a_0\lambda^5 + a_1\lambda^4 + a_2\lambda^3 + a_3\lambda^2 + a_4\lambda + a_5 = 0, \tag{9}$$

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

$$\begin{aligned} c_{11}, c_{12}, \ldots, c_{16} &= a_0, a_1, \ldots, a_5, \\ d_{11}, d_{12}, \ldots, d_{16} &= a_5, a_4, \ldots, a_0, \end{aligned} \tag{10}$$

and subsequent rows defined by:

$$c_{\beta\gamma} = \begin{vmatrix} c_{\beta-1,1} & c_{\beta-1,\gamma+1} \\ d_{\beta-1,1} & d_{\beta-1,\gamma+1} \end{vmatrix}, \tag{11}$$

$$d_{\beta\gamma} = c_{\beta,8-\gamma-\beta}, \tag{12}$$

where $\beta = 2, 3, 4, 5, 6$ and $\gamma = 0, 1, 2, 3$. Jury-Marden's Theorem [1] considers that all roots of polynomial $p(\lambda)$ have modulus less than one if and only if $d_21 > 0, d_\tau < 0$, for $\tau = 3, 4, 5, 6$. Hence, solving this conditions results in (8). ■

Consequently, by Proposition 3.1, Theorem 3.1 and Theorem 3.2, the conditions in (8) are obtained so that all solutions of (5) converge to x_n^*. Although it was possible to define a relatively small attraction domain, it is necessary to further explore particular conditions of the algorithm, by redefining parameters values and their relation.

3.3 Parameter Adjustment

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

$$0 \leq v\left(\alpha + \frac{1}{2}\alpha + \frac{1}{6}\alpha(1-\alpha) + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)\right) < v, \tag{13}$$

thus resulting in

$$0 < \alpha \leq 0.632. \tag{14}$$

Let us now consider the opposite scenario, *i.e.*, a robot that was stopped $v_n[t-k] = 0$ with $k \in \mathbb{N}_0$ starts to move since $x_n[t] \neq \chi_i, i = 1, 2, 3, 4$. The robot needs to accelerate until it reaches the maximum velocity defined by equation (1), taking into account that $w_n[t] = 0$. Similarly to the procedure presented in (5), but considering the previously described conditions, the following nonhomogeneous first-order difference equation results:

$$x_n[t+1] + \left(\sum_{i=1}^{4} \rho_i r_i - 1\right) x_n[t] = \sum_{i=1}^{4} \rho_i r_i \chi_i[t]. \tag{15}$$

Hence, the characteristic equation associated to (15) is

$$\lambda + \left(\sum_{i=1}^{4} \rho_i r_i - 1\right) = 0. \tag{16}$$

Therefore, using Jury-Marden's Theorem [12], the following condition is obtained:

$$0 < \sum_{i=1}^{4} \rho_i r_i < 2. \tag{17}$$

Hence, the global and particular $\mathcal{A}$ is represented by the parameter region, *i.e.*, attraction domain, of the asymptotic stability depicted in Fig. 1.

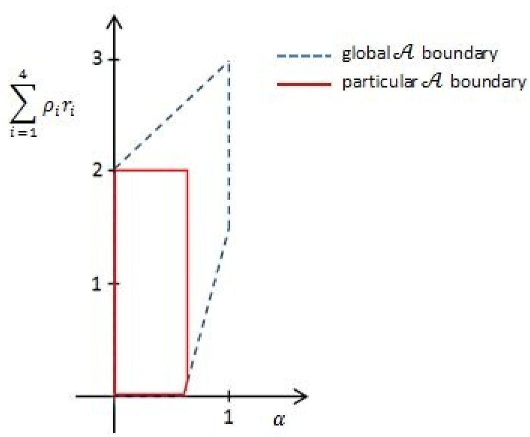

Figure 1: Global and particular attraction domain $\mathcal{A}$ of the asymptotic stability of the *RDPSO*.

As a result of the above analysis, the *RDPSO* can be conceived in such a way that the system's convergence can be controlled taking into account obstacle avoidance and *MANET* connectivity, without resorting to the definition of any arbitrary or problem-specific parameters. Next section presents experimental results obtained using physical robots wherein the *RDPSO* was parameterized, bearing in mind the particular attraction domain $\mathcal{A}$ presented in Fig. 1.

4 Experimental Results

In this section, it is evaluated 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. All of the experiments were carried out in a 2.55 meters to 2.45 meters scenario. The experimental environment (Fig. 3) was an enclosed arena that contained two sites. Each site was represented by an illuminated spot uniquely identifiable by controlling the brightness of the light. Despite being an obstacle free scenario, the robots themselves act as dynamic obstacles - note that a maximum population of 12 robots correspond to a population density of approximately 2 robots per square meter. The *eSwarBot* (Educational Swarm Robot) was the platform used to evaluate the algorithm [5]. Although the platform presents a limited kinematic resolution of 3.6 degrees while rotating and 2.76mm when moving forward, its low cost and high autonomy allowed performing experiments with up to 12 robots. Robots were equipped with *RGB-LEDs* on top of them to identify their swarm and overhead light sensors (*LDR*) to find candidate sites and measure their quality. The brighter site (global optimum) was considered better than the dimmer one (local optimum), and so the robot's goal was to collectively choose the brighter site.

Figure 2: Experimental Setup.

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 (near 30 meters in indoor 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} varied between 0.5 meters and 1.5 meters. At each trial, robots were manually deployed on the scenario in a spiral manner (as previously presented) while preserving the communication distance d_{max}. Since the *RDPSO* is a stochastic algorithm, it may lead to a different trajectory convergence whenever it is executed. Therefore, a test group of 20 trials of 3 minutes each was considered for $N = 4, 8, 12$ robots and $d_{max} = 0.5, 1.5$ meters. Also, a minimum, initial and maximum number of $1, 2$ and 3 swarms were used. The algorithm parameters where chosen in order to satisfy conditions (14) and (17), with $\alpha = 0.5, \rho_1 = \rho_2 = 0.3, \rho_3 = \rho_4 = 0.6$. The previously described conditions give a total of 120 experiments, thus leading to a run-time of 6 hours. Fig. 3 depicts the normalized performance of the algorithm, by changing the maximum communication distance d_{max} and the number of robots N. Boxplot charts are used because they are a quick way of examining graphically the final result of each trial. The ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values, respectively.

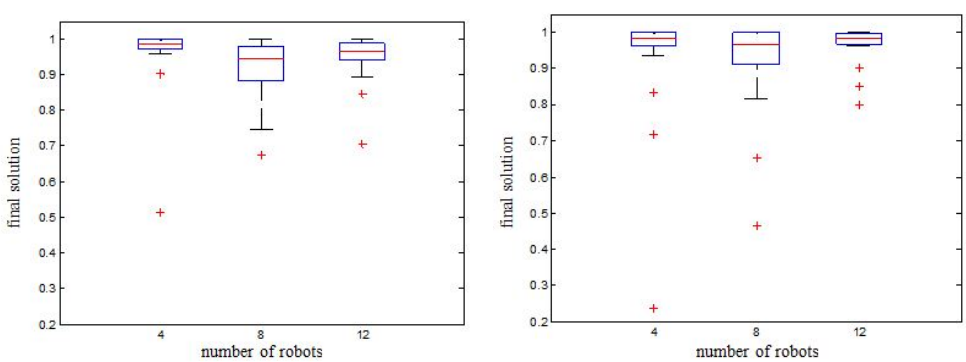

Figure 3: RD*PSO* evaluation changing the maximum communication distance d_max and the number of robots N, a) $d_{max} = 0.5$; b) $d_{max} = 1.5$.

As expected, with the previously specified parameters, the algorithm converges to the solution (*i.e.*, normalized solution of 1) in approximately 90% of the experiments regardless on the number of robots and the communication range. The data distribution, despite the considered trial, turns out to be negatively skewed (*i.e.*, the median is higher than the mean value). This means that, in this case, as the goal is to maximize the fitness function (*i.e.*, find the brighter site), most of the trials are near the desired objective value. Also, it can be seen that increasing the population size from 4 to 8 robots and, consequently, the dynamic obstacles within the scenario, leads to a slightly decrease of performance of the algorithm. However, the performance of the algorithm using a larger population of 12 robots improves when the maximum communication distance increases. It is noteworthy that robots still collide in some situations (a mean of 3 collisions per experiment were verified when using 12 robots) and sometimes are unable to guarantee the maximum communication distance between them (a mean of 3 communication breaches per experiment were verified when using 4 robots). There are two main reasons leading to this behavior. The first reason lies in the adopted kinematic strategy. Robots are programmed with brownian movements where they first orientate, thus moving forward until they reach the target location, which is defined by equations (1) and (2). Although the maximum step between iterations was limited to 150 millimeters, when the robot changes its orientation it does not read the sensor information which may cause some collisions between them. Furthermore, the use of low-cost encoders, such as the ones used in the eSwarBots, present significant cumulative errors. Consequently, it is difficult for robots to complete the proposed odometry objectives accurately, thus being sometimes unable to fulfill communication constraints. Secondly, the methodology proposed herein takes into account that parameters are fixed at constant values throughout the search. However, there are some situations in which parameters should adapt. For instance, if a robot is near collision, the obstacle susceptibility weight ρ_3 should instantaneously increase, hence ignoring the mission and communication constraints.

5 Conclusions

The previously proposed *RDPSO* algorithm is a parameterized foraging algorithm which takes into account real-world multi-robot systems (MRS) characteristics, such as obstacle avoidance and communication constraints. This paper presented the convergence analysis of the algorithm, studying its stability in such a way that parameters may be configured within a small attraction domain. Moreover, an extended analysis based on the dynamical characteristics of robots is conducted, thus decreasing the size of the attraction domain. Experimental results show that the algorithm converges in most situations regardless on the number of robots and the maximum communication range between them. However, it is still possible to exhibit some minor collisions and communication ruptures between robots. Therefore, one of the future approaches will be extending the *RDPSO* with adaptive parameterization to overwhelm these issues. Since the swarm behavior may need to change during the search, it is possible to control the swarm susceptibility to the main mission, obstacle avoidance and communi-cation constraint, by systematically adjusting the parameters within the herein defined attraction domain. Furthermore, an extended stochastic convergence analysis of the *RDPSO* must be undertaken since the analytical results presented in this paper may deviate from the real stochastic algorithm because randomness was not studied.

ACKNOWLEDGEMENTS.
This work was made possible by the support and assistance of Carlos Figueiredo and Miguel Luz and for their cooperation at RoboCorp in the Engineering Institute of Coimbra (ISEC). This work was supported by a PhD scholarship (SFRH/BD /73382/2010) granted to the first author by the Portuguese Foundation for Science and Technology (FCT), the Institute of Systems and Robotics (ISR) and the Institute of Telecommunications (IT-Covilhï¿½) also under regular funding by FCT.

References

[1] S. Barnett, *Polynomials and Linear Control Systems,* Marcel Dekker, Inc.,New York, USA, 1983.

[2] M. Clerc and J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation,* **6(1)** (2002), 58 - 73.

[3] 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,* (2011), 327–332.

[4] M. S. Couceiro, N. M. F. Ferreira and J. A. T. Machado, Fractional Order Darwinian Particle Swarm Optimization, *3th Symposium on Fractional Signals and Systems, FSS'2011*, (2011), 127 - 136.

[5] 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), IN PRESS.

[6] M. S. Couceiro, J. M. A. Luz, C. M. Figueiredo and N. M. F. Ferreira, Modeling and Control of Biologically Inspired Flying Robots, *Journal of Robotica - Press, Cambridge University,* (2011), 107 - 121.

[7] S. Elaydi, *An Introduction to Difference Equations,* Springer, Science and Business Media, Inc., Texas, USA, 2005.

[8] H. K. Hahn and K. Schoenberger, The Ordered Distribution of Natural Numbers on the Square Root Spiral, *The Journal of Business,* (2007), 1 - 5.

[9] V. Kadirkamanathan, H. Selvarajah, and P. J. Fleming, Stability analysis of the particle dynamics in particle swarm optimizer,*IEEE Transactions on Evolutionary Computation,* **10(3)** (2006), 245 - 255.

[10] 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), 39 - 43.

[11] I. Podlubny, *Fractional Differential Equations,* San Diego, California: Academic Press, 1999.

[12] 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), 1474 - 1487.

[13] K. Yasuda, N. Iwasaki, G. Ueno and E. Aiyoshi, Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity,*IEEJ Transactions on Electrical and Electronic Engineering, Wiley InterScience,* **3** 2008, 642 - 659.

Received: January, 2012