Performance Evaluation of Novel Distributed Coverage Techniques for Swarms of Flying Robots

Valeria Loscrì, Enrico Natalizio, Nathalie Mitton

▶ To cite this version:

HAL Id: hal-00920752

https://hal.inria.fr/hal-00920752

Submitted on 30 Apr 2014

Performance Evaluation of Novel Distributed Coverage Techniques for Swarms of Flying Robots

Valeria Loscrí[*], Enrico Natalizio[†], Nathalie Mitton[*]

[*]Inria Lille - Nord Europe, Lille, France. e-mail: *firstname.lastname*@inria.fr
[†]Lab. Heudiasyc, UMR CNRS 7253, UTC, France. e-mail: enrico.natalizio@hds.utc.fr

Abstract—This paper focuses on the coverage of specific Zones of Interest that can change dynamically over time by using a swarm of flying robots. The mobility of the flying devices is achieved by the design of two distributed and local algorithms. The first algorithm is based on Particle Swarm Optimization (PSO) and Virtual Forces Algorithm (VFA). We modify the classical PSO approach to propose a totally distributed algorithm, which only requires the flying robots to receive local information from the neighbors to update their velocity and trajectory (PSO-S). This new distributed version of the PSO is combined with a distributed version of the Virtual Forces Algorithm. The second algorithm is a distributed implementation of the VFA (VFA-D). To the best of our knowledge, these two approaches are novel in their distributed character, scalability and implementability on resource-constrained devices. We show that the proposed algorithms are reactive, i.e. able to capture in an effective fashion the events happening within the field even if the position of the events changes over time. To show the effectiveness of the proposed techniques, we perform extensive simulations to compare both the PSO-S and the VFA-D schemes with a centralized version of the VFA. Simulations show the good performance in terms of coverage and traveled distance as well as the high reactivity of both PSO-S and VFA-D when the ZoI changes.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV), also called *drones* and referred to as *flying robots* in this paper, are currently the fastest growing segment of the global aerospace industry. The Association for Unmanned Vehicle Systems International (AUVSI), an industry group, has estimated the drones industry could contribute more than $80 billion to the U.S. economy over a decade[1]. So far, flying robots' industry has been largely limited to military and law enforcement applications, but the potential applications of fleets of flying robots will include several other domains, such as the environmental, agriculture, civil and entertainment domains [10].

In this work, we focus on the dynamic event coverage performed by robots flying within a limited area. We assume that we have no *a priori* knowledge about events' position and duration. Thus, the robots have to discover the events, monitor them and move towards a new Zone of Interest (ZoI) when the previous monitored event is over. This paper proposes two novel distributed coordination algorithms to deal with dynamic event coverage. We use two approaches of devices coordination that have been strongly consolidated by an extensive usage in Mobile Wireless Sensor Networks: the Particle Swarm

Optimization (PSO) and the Virtual Force Algorithms (VFA). The former is an approach usually included into the Swarm Intelligence literature [13], [6], as it uses the behavior of social bugs as an inspiration source to solve hard problems. In the PSO, a set of *particles* are placed in the search space of an optimization problem and move in this space according to certain rules and global information to find the best solution for the problem. The latter is based on the concept of the virtual forces field and the main objective is the maximization of the coverage in a Wireless Sensor Network (WSN) [16]. In the same way electro-magnetic particles attract or repel each other based on the value of the potential field, sensor nodes attract or repel each other based on their mutual distance. In this paper, we extend and integrate the two mentioned approaches to make them suitable to swarms of flying robots that monitor a dynamic environment.

Our main contributions are the following:

- We propose and simulate a novel implementation of the classical VFA, later referred as VFA-D. The main difference from the original version is that our proposal manages highly dynamical environments distributedly;

- We propose the integration of a localized version of the PSO with the VFA-D, and combines the reactivity of the PSO and the accuracy of the VFA. This scheme will be later referred as PSO-S;

- We investigate, through an extensive performance evaluation, the behavior of the proposed techniques in respect of the centralized version of the classical VFA [16], and in terms of capability to cover time-changing ZoI while minimizing the traveled distance to move to the new ZoI.

The rest of the paper is organized as follows. Section II formulates the problem and introduces the Centralized Virtual Force algorithm (VFA-C), the Distributed Virtual Force Algorithm (VFA-D), the local variant of PSO and its combined version with the Distributed Virtual Force technique (PSO-S). Section III presents the simulation model and the events' pattern and Section IV shows the results. Section V surveys related work, and finally, Section VI concludes the work.

II. PROBLEM FORMULATION AND CONTRIBUTIONS

This section introduces the problem, our distributed version of Virtual Force Algorithm (VFA-D) and Serial Particle Swarm

---

[1]www.reuters.com/article/2013/08/08/us-usa-drones-commercial-idUSBRE97715U20130808

Optimization (PSO-S), outlining their differences with the centralized version and the original PSO, respectively.

A. Problem Formulation

Let consider a network of N energy-constrained flying robots $S = S_1, S_2, \ldots, S_N$, flying over a two-dimensional flat field F and A ZoIs $Z = Z_1^T, Z_2^T, \ldots, Z_A^T$ with $A \leq N$ in order to sense the events happening in the field. Each of the ZoIs is represented by a two-dimensional surface in F, and the set Z changes over time period T. For the sake of simplicity, we assume that the devices move at a fixed distance (altitude) above the events field and that the distance is smaller than the sensing range of the devices. Future studies will focus on determining the most suited altitude to satisfy the desired sensing resolution. The problem is to move the devices to positions $P = P_1, P_2, \ldots, P_N$, so that the surface of all the ZoIs in Z is covered while minimizing the time spent and the distance traveled to reach these positions.

B. Distributed Virtual Force Algorithm: VFA-D

The VFA is based on the concept of the virtual forces field and the main objective is the maximization of the coverage in a WSN [16]. In the model in [16], the total force $\vec{F_i}$ exerted on node i is given as the sum of three forces:

$$\vec{F_i} = \sum_{n=1, n \neq i}^{N} \vec{F}_{i,j} + \sum_{o=1}^{O} \vec{F}_{iR_o} + \sum_{a=1}^{A} \vec{F}_{iA_a} \quad (1)$$

1) $\vec{F}_{i,j}$ is the total force (attractive and repulsive) that all the n nodes present in the field exert on node i;
2) $\vec{F}_{iR}$ is the total repulsive force that all the O obstacles exert on node i. In this work, we consider the perimeter of the field as an obstacle;
3) $\vec{F}_{iA}$ is the total attractive force that all the A zones to be monitored exert on node i.

The first force can be expressed by the following formula:

$$\vec{F}_{i,j} = \begin{cases} (w_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \left(w_R(\frac{1}{d_{ij}}), \alpha_{ij} + \pi\right) & \text{if otherwise} \end{cases} \quad (2)$$

where d_{ij} and α_{ij} are the Euclidean distance and the angle between nodes i and j, d_{th} is the threshold distance for nodes to attract or repel each other, w_A and w_R are the weights of the attractive and repulsive forces, respectively. The novel position is calculated in [7] from, F_{xy} the magnitude of $\vec{F_i}$ and its x and y components, F_x and F_y, as follows:

$$x_{new} = x_{old} + (\frac{F_x}{F_{xy}}) \cdot MaxStep \cdot e^{(\frac{-1}{F_{xy}})} \quad (3)$$

$$y_{new} = y_{old} + (\frac{F_y}{F_{xy}}) \cdot MaxStep \cdot e^{(\frac{-1}{F_{xy}})} \quad (4)$$

where $MaxStep$ is the predefined single maximum moving distance. In the classical version of this algorithm, a central entity is requested to collect all the information from the nodes in order to compute the total force exerted on each of them, for this reason we will refer to this technique as VFA-C (VFA -

Centralized). Besides the existence of a single point of failure, we show that the VFA-C fails when ZoIs change dynamically.

In this work, we modified and implemented a new version of the VFA that avoids the usage of a central entity to calculate the forces. VFA-D introduces a maximum distance, called C, related to the sensing range of the nodes. When another node, an obstacle or a ZoI is farther than C from the current node, then its effects on the node are considered negligible. C is assumed 4 times greater than the sensing range of a node and is consequently related with the communication range, twice greater than the sensing range. This is related with the capability of a node to acquire information in a distributed fashion. The distributed version of VFA approach requires a specific setup of characteristic parameters such as the weights associated to the attractive force and the repulsive force, w_A and w_R respectively. It is worth noting that a similar approach is suitable with a heterogeneous scenario where nodes with different sensing ranges are considered. In fact, every node will compute its own force based only on its sensing range and its neighbors. The mathematical model considered to compute forces among our flying devices is as follows:

$$\vec{F}_{i,j} = \begin{cases} 0 & \text{if } d_{ij} \geq C \text{ or if } d_{ij} = d_{th} \\ (w_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } C > d_{ij} > d_{th} \\ \left(w_R(\frac{1}{d_{ij}}), \alpha_{ij} + \pi\right) & \text{if } d_{ij} < d_{th} \end{cases} \quad (5)$$

C. Serial Particle Swarm optimization Algorithm (PSO-S)

PSO is a swarm intelligence technique based on particles positions [1], localized inside a searching space that evaluate an objective function depending on their own position. They can also move around the searching space and combine their own knowledge with the information received from one or more neighbors. By assuming a 2D searching space, the velocity of the units will be computed iteratively as:

$$\vec{v_i}(t+1) = \omega \cdot \vec{v_i}(t) + \phi_p \cdot \vec{r_p} \circ (\vec{p_i} - \vec{x_i}(t)) + \phi_g \cdot \vec{r_g} \circ (\vec{p_g} - \vec{x_i}(t)) \quad (6)$$

where $x_i(t)$ and $v_i(t)$ are the current position and the velocity of the particle i; p_i and p_g are resp. the best positions of i and of the swarm; r_p and r_g are two random vectors in the domain $U(0,1)$; w, ϕ_p and ϕ_g are selected parameters to control the efficiency of the PSO technique and $\circ$ is the Hadamard multiplicative operator. The three components are also referred as *inertia*, *cognitive component* and *social component*, respectively. The new updated position of the particle at the next step is given incrementally from the previous position when the new velocity has been applied in the time instant under observation as:

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t+1) \quad (7)$$

In this work we use the variant of PSO that makes a node consider the local best achieved in its neighborhood. The velocity update equation results as:

$$\vec{v_i}(t+1) = \omega \cdot \vec{v_i}(t) + \phi_p \cdot \vec{r_p} \circ (\vec{p_i} - \vec{x_i}(t)) + \phi_g \cdot \vec{r_g} \circ (\vec{l_i}) \quad (8)$$

where $\vec{l_i} = \frac{\vec{x_k} - \vec{x_i}}{\|\vec{x_k} - \vec{x_i}\|} \cdot \frac{\|\vec{x_k} - \vec{x_i}\|}{d_{rep}}$ with x_k the position of the node in the set of neighbors of i that sensed the most events in the previous iteration and d_{rep} a repulsive coefficient that avoids

the overlap of nodes. The inertial weight w varies between w_{max} and w_{min}, as in [13].

PSO-S algorithm considers separately the local variant of the PSO and the VFA-D. Specifically, it first applies the local variant of PSO and when a sub-optimum solution is achieved, VFA-D optimizes the final coverage solution. Since the resulting algorithm applies the two presented schemes in a serialized way, we named it PSO-S (PSO - Serialized). Of course, in this case we need to specify the exact times of stop (for the local variant of PSO) and start (for the VFA-D). We formulate three termination conditions:

1) node position does not change significantly during the last iteration (traveled distance smaller than the termination distance d_c),
2) node coverage does not change significantly during a certain number of iterations (coverage improvement smaller than the coverage threshold c_{th}),
3) node has already run the algorithm for several iterations (more than the maximum number of iterations it_{max}).

When any of the previous conditions is verified, the local variant of PSO stops and the VFA-D enters into action. Specifically, the third condition is useful when nodes keep moving without finding a satisfying solution. Note that all these conditions are checked distributedly by each node. The main goal of combining the two algorithms is the correct distribution of nodes. In practice, the variant of the PSO allows nodes to be attracted from a zone of interest in a distributed fashion by reaching a "local" consensus, but even if we included in the formulation of the PSO variant a repulsive effect, nodes will tend to cover the perimeter of ZoI.

III. The Simulation Environment

In order to evaluate the effectiveness of our algorithms, we consider a $100\ m\ \times\ 100\ m$ field with events occurring simultaneously, and N flying robots, whose initial positions are random. We assume that the ZoIs change dynamically during the simulation time, in order to simulate a sequence of events that appear and disappear in the field. Our first objective is to achieve a high level of coverage for the ZoI where events occur, but we also take into consideration the traveled distance. The traveled distance can show the impact of movement for the different techniques. The parameters used in the simulation are summarized in Table I. Our first campaign of simulations, show that we obtain good results when repulsive forces are much bigger than attractive ones ($\omega_R >> \omega_A$). Fig.1 shows the scenarios simulated to evaluate the effectiveness of our techniques and to show how they capture the events in a distributed fashion.

IV. Simulation Results

In order to evaluate the performance of our techniques, we consider two output parameters: 1) the coverage, as the fraction of ZoI covered by sensors in order to see the effectiveness of the tested algorithms, 2) the distance traveled by nodes movement, which represents the cost of the algorithms. Usually, the energy consumed for moving is much higher than that consumed for transmitting data, thus the traveled

TABLE I.   Evaluation Parameters

| Field Area (LxL)	100 m x 100 m
Number of Flying Robots (N)	30-80
Sensing Radius (R_s)	7 m
Communication Radius (R_c)	$2R_s$
Termination Distance (d_c)	0.5 m
Maximum Number of Iterations (it_{max})	500
Inertia Weights ($w_{min} - w_{max}$)	$0.1 - 0.7$
Attractive Force (w_A)	0.01
Repulsive Force (w_R)	1000
Coverage Threshold (c_{th})	$0.5 - 0.9$
Threshold Distance (d_{th})	$2R_s$
Repulsion Coefficient for PSO (d_{rep})	$2R_s$
Confidence Interval	95%
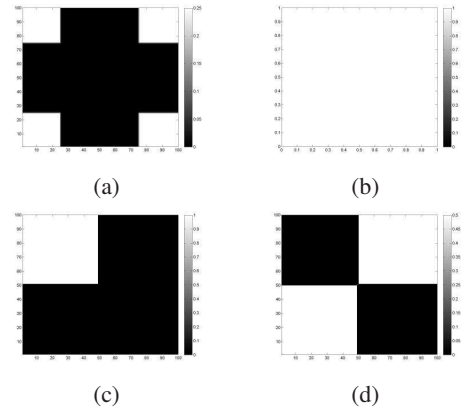
Fig. 1.   Events occur in white areas. (a) Scenario 1: events are concentrated in the corners, (b) Scenario 2: events are uniformly distributed, (c) Scenario 3: events are concentrated in one square, (d) Scenario 4: events are concentrated in 2 squares.

distance can represent also the energy consumption. We have also taken into consideration the number of iterations, which helps understanding the reactivity of the algorithms, but we omit to show the plots related with this parameter because of page limitation. As we assume a probabilistic model for Virtual Forces [7], [16], we consider that a generic point in the field is covered when its coverage is larger than a certain coverage threshold, c_{th}. The probabilistic model that we are considering in this work can be summarized as follows:

$$\vec{c}_{xy}(s_i) = \begin{cases} 0 & \text{if } r + r_e \leq d(s_i, P) \\ 1 & \text{if } d(s_i, P) \leq r - r_e \\ \left(e^{(-\alpha_1 \lambda_1^{\beta_1}/\lambda_2^{\beta_2} + \alpha_2)} \right) & \text{otherwise} \end{cases} \quad (9)$$

where r_e is the measure of detection uncertainty, $\lambda_1 = r_e - r + d(s_i, P)$ and $\lambda_2 = r_e + r - d(s_i, P)$, α_1, α_2, β_1 and β_2 are detection probability parameters. The values of α_1, α_2, β_1 and β_2 depend on the characteristics of the sensors mounted on the robots. In practice, the probabilistic detection model takes account of the fact that it could be necessary to overlap some areas with more robots in order to compensate the low detection probability in the area. Be S_{ov} a set of robots that "overlap" a certain point with coordinates (x, y), we can compute the detection probability that a point can be successfully detected by at least one robot:

$$c_{x,y}(S_{ov}) = 1 - \prod_{s_i \in S_{ov}} (1 - c_{x,y}(s_i)) \quad (10)$$

where $c_{x,y}(s_i)$ is the detection probability of s_i at (x,y). (x,y) is effectively covered if $\min_{x,y} c_{x,y}(s_i, s_j) \geq c_{th}$.

Performance parameters are evaluated by considering the threshold value c_{th} ranging from 0.5 and 0.9.

We present the results of two simulation campaigns:

- when the number of flying robots varies between 30 and 70 with a coverage threshold of 0.7.

- when the coverage threshold varies between 0.5 and 0.9 with 80 robots.

For both simulation campaigns, we assume that events in the field are initially placed as in one of the four scenarios presented in Fig.1 and that, at a random instant during the simulation, the events disappear from the initial scenario and reappear as in one of the other three scenarios. Among the 12 possible combinations of switching, in this paper, we only present 4 of them: the switch between scenarios 2-3 and 4-2 for the first simulation campaign, and the switch 2-4 and 3-2 for the second simulation campaign. Only 3 of the 4 scenarios are used to investigate the effects of concentrating the events, spread all over the sensor field (scenario 2), in a single area (scenario 3) or in two areas (scenario 4), and vice versa. All the results have been averaged over 100 runs in order to respect a confidence interval of 95%.

A. Number of devices

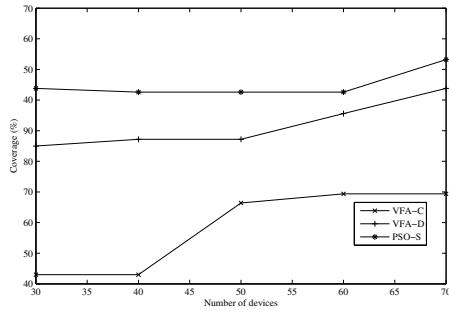Figure 2 displays the difference of coverage achieved by our two proposed schemes (VFA-D, PSO-S) and the centralized version of the virtual force algorithm (VFA-C), when the number of devices increases and the scenario switches from a diffused placement of events (scenario 2) to a concentrated one (scenario 3). PSO-S is the scheme that achieves the highest coverage, which is, on average, 12% higher than the VFA-D, and 25% higher than the PSO-S. Of course, a larger number of devices allows all the schemes to obtain a higher coverage.

The slowest growth rate of coverage for VFA-C comes from the centralized nature of this scheme, which requires a higher number of devices to cover the whole area. In terms of traveled distance, we can see from Fig. 3 that the VF algorithms and the PSO behaves in a completely different way. In fact, VFA-C and VFA-D make node travel more when the number of nodes increases, whereas the traveled distance for the PSO-S drops drastically. This is the effect of better communications among the nodes, and shows that, for this parameter, the PSO-S can sum up the positive effects of a centralized approach (like the VFA-C) in a distributed way, by using the PSO on a large scale and the VFA as a refinement of the final displacement. In fact, as a proof of this reasoning we can see that the VFA in its distributed version causes the highest traveled distance.

While in the previous diagrams we have seen the effects of concentrating the ZoIs, in the following figures, we will see the effect of spreading the events from two specific areas (scenario 4) to the whole field (scenario 2).

In this case, the PSO-S, which still achieves the best results, is not able to keep the organization shown in the previous case. In fact, the coverage decreases when the number of devices increases from 30 to 50 and then increases again, whereas the other schemes improve the coverage when they can use
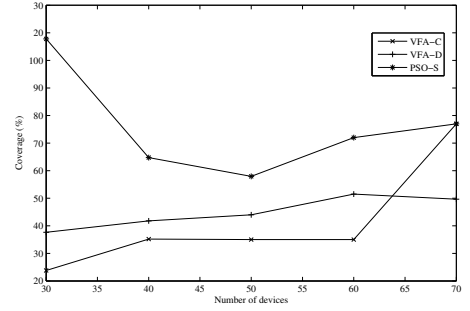

Fig. 2. Coverage when Scenario 2 changes to Scenario 3.

Fig. 4. Coverage when Scenario 4 changes to Scenario 2.

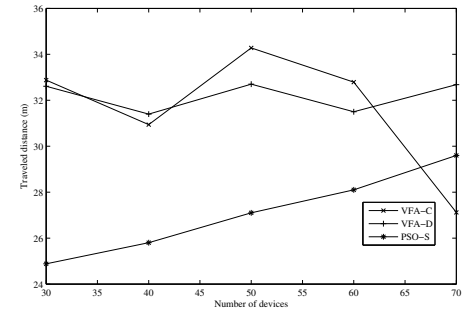Fig. 3. Traveled distance when Scenario 2 changes to Scenario 3.

Fig. 5. Traveled distance when Scenario 4 changes to Scenario 2.

more flying robots. The consideration highlighted for the PSO-S leads to the conclusion that a specific number of robots should be determined to optimize the behavior of this scheme in case of spreading of the ZoIs. The PSO-S is still the best algorithm in terms of traveled distance, as shown in Fig. 5. In fact, it allows the nodes to travel from 4 to $8m$ less in respect of the other two algorithms, which have basically a stable behavior for all the number of nodes (1 m of variation for the VFA-D and 3 m for the VFA-C) but the last one (70 devices), where the VFA-C behaves even better than the PSO-S. If we recall that VFA-C achieves its highest coverage for the last considered number of devices (70), we can conclude that a centralized solution becomes necessary when the number of nodes grows over a certain threshold for this change of scenario.

This simulation campaign showed that the passage between concentrated and spread event placements is less taxing for robots than the opposite (from the spread event placement to the concentrated one). Also, it showed that the PSO-S performs better in terms of coverage and traveled distance.

B. Coverage threshold

In this campaign, we fix the number of devices and let the coverage threshold varies between 0.5 and 0.9. The coverage threshold is a parameter that comes from the probabilistic approach used to characterize the coverage of a point in the field. An increase of the coverage threshold for a same number of devices means, in general, a lower probability to consider the point as covered. Thus, the following diagrams present some peculiarities which are not easy to analyze and comment.
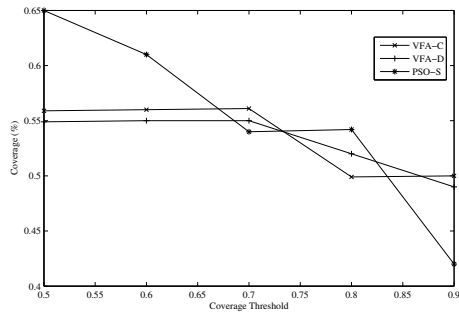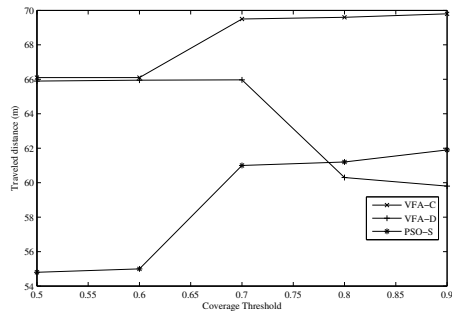
The following figures show the impact of concentrating the ZoIs in two areas (scenario 4) from an initial situation of spread events (scenario 2). Fig. 6 shows that the coverage decreases for all the algorithms when the threshold on the required value to consider a point covered increases. VFA-C and VFA-D behave almost in the same way, even though they use global and local information respectively, whereas the PSO-S shows the best performance when the coverage threshold is very low (0.5) and worsens when it increases. Note that even though PSO-S does not achieve the best coverage performance for all the simulated thresholds (Fig. 7), still it shows the lowest traveled distance to achieve a stable placement. Only the VFA-D consumes less when the coverage threshold is very high, whereas the VFA-C consumes the most energy, even though it shows the same coverage achieved by the distributed version.

The following figures represent again the case of a spread situation (scenario 2) starting from a concentrated situation (3). We will see again that this change of scenario is less intuitive and easy to understand. In terms of coverage (Fig. 8), the VFA-D is the best algorithm and achieves its maximum for a coverage threshold equal to 0.7, which is the middle point used in the simulations. As we could expect, the PSO-S does not present a very high performance, because as we know from the previous campaign, it is not able to manage a too high number of devices (80 for this simulations), which is instead the best work condition for the centralized approach.

The distance traveled to reach a stable placement, in Fig. 9, varies for all the algorithms in a very small range ($3m$) which makes difficult a precise evaluation of the performance. However, we can still appreciate that 0.7 as coverage threshold represents some kind of "switching" behavior for both the
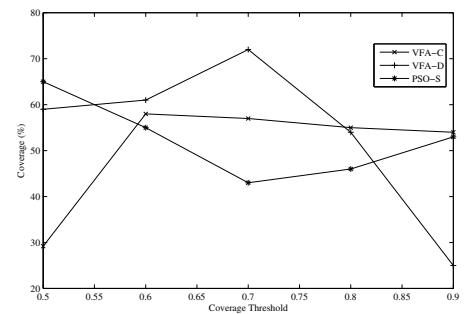

Fig. 6. Coverage achieved when Scenario 2 changes to Scenario 4.

Fig. 8. Coverage achieved when Scenario 3 changes to Scenario 2.

Fig. 7. Traveled distance when Scenario 2 changes to Scenario 4.
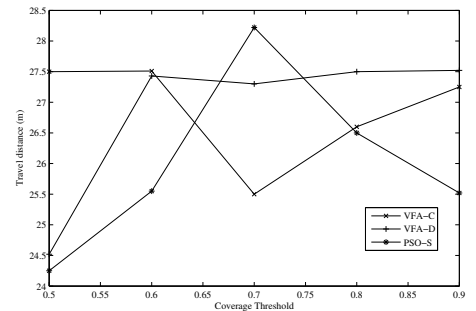

Fig. 9. Traveled distance when Scenario 3 changes to Scenario 2.

VFA-C and PSO-S. In this case, the most stable behavior is presented by the VFA-D, which simply requires robots to fly more to cover the same areas with the same number of devices, when the coverage threshold increases. Fig. 9 shows that this algorithm makes robots fly more than the other two, but not by a large value. We can conclude that the VFA-D makes nodes do very small movements that affect the achieved coverage.

The lessons learnt from this campaign are similar than for the first regarding the passage from a spread to a concentrated scenario and vice versa. Instead, we have seen that, for a varying coverage threshold, the algorithms show a threshold effect around 0.7, for which they change behavior (as the number of iterations for the PSO-S in the switch 2-4 or the coverage for VFA-D in the switch 3-2). Again, the best performance is always achieved by PSO-S or VFA-D, except in some specific cases when the system was highly stressed.

V. RELATED WORK

Events coverage by wireless mobile devices has been widely investigated. However, most literature works pursue goals that are different from ours. Many improvements have been proposed for path planning rather than events coverage [11], [9]. Gusrialdi *et al.* propose a distributed coverage based on the combination of the standard gradient-based coverage algorithm and a leader following algorithm but is designed to maximize the joint detection probabilities of the events in the region of interest [4]. Kong *et al.* assign a portion of the area to monitor to each robot and then run individual algorithms on the robots [5]. Although the authors claim that the coverage is distributed, the area sharing is performed in a centralized way. Dong *et al.* propose a distributed coverage approach based on duty cycle and tight synchronization, which is not suited for flying robots [3]. Pourshoghi and Talebi propose to first organize the flying robots into a hexagonal cluster and then to move them as a single robots to perform sweep coverage of the area to monitor [12]. However, this approach does not allow the detection of dynamic events. Other works try to address the coverage issue jointly with data collection through the use of Bayes estimation approach. While distributed, such an approach requires a large amount of information to process at every node [15]. The PSO technique has been evaluated by considering different objectives and scenarios [6]. However, this scheme is centralized and does not consider the possibility for the nodes to move as a reaction to a change in the events position. The concept of Virtual Forces has been introduced in Mobile Wireless Sensor Networks to improve coverage of a specific geographic area [16]. Initially, nodes are randomly deployed and then are subject to attractive and repulsive forces from other nodes that make them move to maximize the area coverage. Even though VFA is very efficient for coverage, some improvements have been proposed in recent years to enhance connectivity, fault tolerance, energy consumption [8] or image capture [14]. Nevertheless, all these works require global information on nodes positions while in our work, we use only local information. Also evolutionary and learning mechanisms have been used to let nodes find their best placement [2]. However, these schemes do not handles dynamic ZoIs, which would guarantee a continuous monitoring of the field and an autonomous behavior of the mobile nodes.

VI. CONCLUSION

In this paper, we considered particular distributions of events in a field. We assumed flying robots able to communicate with each other. The specific task of the robots is to cover portions of a square area where events occur. Specifically, we considered the centralized Virtual Force Approach (VFA-C), introduced a distributed version of the VFA-C, the VFA-D and a technique that combines a local and distributed version of the PSO and the VFA-D, the PSO-S. We showed the effectiveness of the new techniques to cover the ZoI, that is, the zones where the events occur. Even when these zones are at the extreme of the field and the distributions of the events change in a dynamic fashion, our techniques showed remarkable performance. We showed the reactivity as an inherent property of these techniques. In future works, we will determine an automatic association between the choice of the weights for both the PSO and the VFA-D.

REFERENCES

[1] T. Blackwell, J. Branke, and X. Li, "Particle Swarms for Dynamic Optimization Problems", in Swarm Intelligence Natural Computing Series, 193-217, 2008

[2] C. Costanzo, V. Loscrí, E. Natalizio, T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network", Computer Communications, vol.35(9), 2012.

[3] D. Dong, X. Liaon, K. Liu, Y. Liu, W. Xu, "Distributed Coverage in Wireless Ad Hoc and Sensor Networks by Topological Graph Approaches", IEEE Transactions on computers, vol.61(10), 2012.

[4] A. Gusrialdi, R. Dirza , S. Hirche, "Information-Driven Distributed Coverage Algorithms for Mobile Sensor Networks", Proc. Int. Conf. on Networking, Sensing and Control (ICNSC), Delft, the Netherlands, 2011.

[5] C.S. Kong, N.A. Peng, I. Reikleitis, "Distributed Coverage with Multi-Robot System", Proc. Int. Conf. on Robotics and Automation (ICRA), Orlando, USA, 2006.

[6] R.V. Kulkarni, G.K. Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," IEEE Transactions on Systems, Man, and Cybernetics, vol.41(2), 2011.

[7] S. Li, C. Xu, W. Pan and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets', Information Fusion, vol.2(7), 2005.

[8] T. Li, Y. ChongChong, Y. Minghua, "A multi-center self-deployment algorithm of mobile sensor network," Proc. Int. Conf. on Computer Science and Education (ICCSE), 2010.

[9] W. Li, GY Wang, "Application of Improved PSO in Mobile Robotic Path Planning", Proc. Int. Conf. Intelligent Computing and Integrated Systems (ICISS), Gandhinagar Gujarat, India, 2010.

[10] R. R. McCune, G. R. Madey, "Agent-Based Simulation of Cooperative Hunting with UAVs", Proc. Agent-Directed Simulation Symposium (ADSS), San Diego, USA, 2013.

[11] Y. V. Pehlivanoglu, "A new particle swarm optimization method for the path planning of UAV in 3D environment", Journal of Aeronautics and space technologies , vol.5(4), 2012.

[12] A. Pourshoghi, H. Talebi, "A New Distributed Coverage Algorithm Based on Hexagonal Formation", Proc. Int. Conf. on Systems, Man, and Cybernetics (SMCS), San Antonio, USA, 2009.

[13] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization," Evolutionary Computation, vol.3,1999.

[14] C. Stern, C. Rasche, L. Kleinjohann and B. Kleinjohann, "Efficient Alignment of Aerial Images Based on Virtual Forces", Proc. Int. Conf. on Autonomic and Autonomous Systems (ICAS), Brisbane, Australia, 2012.

[15] M. Zhong, C. Cassandras, "Distributed Coverage Control and Data Collection With Mobile Sensor Networks", IEEE Transactions on Automatic Control, vol.56(10), 2011.

[16] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," ACM Transactions on Embedded Computing System, vol. 3, 2004.