# Distributed Algorithm to Improve Coverage for Mobile Swarms of Sensors

Valeria Loscrì, Enrico Natalizio, Tahiry Razafindralambo, Nathalie Mitton

# Distributed Algorithm to Improve Coverage for Mobile Swarms of Sensors

Valeria Loscrí*,Enrico Natalizio†,Tahiry Razafindralambo‡, Nathalie Mitton‡
* DEIS - Universitá della Calabria, Cosenza, Italy. e-mail: vloscri@deis.unical.it
†Heudiasyc - UMR CNRS 7253 - Université de Technologie de Compiègne, France. e-mail: enrico.natalizio@hds.utc.fr
‡Inria Lille - Nord Europe, Lille, France. e-mail: firstname.lastname@inria.fr

***Abstract*—In this paper we focus on the problem of dynamic event coverage. We assume that no knowledge about either event position or duration is given *a priori*. Nonetheless, the events need to be monitored and covered thanks to mobile wireless sensors. Thus, mobile sensors have to discover the events and move towards a new Zone of Interest (ZoI) when the previous monitored event is over. An efficient, distributed and localized solution of this problem would be immediately exploitable by several applications domains, such as environmental, civil, etc. We propose two novel approaches to deal with dynamic event coverage. The first one is a modified version of the PSO, where particles (mobile sensors, nodes or devices in the following) update their velocity by using only local information coming from their neighbors. In practice, the velocity update is performed by considering neighbors' sensed events. Our distributed version of PSO is integrated with a distributed version of the Virtual Force Algorithm (VFA). Virtual Force approach has the ability to "position" nodes with no overlap, by using attractive and repulsive forces based on the distance between nodes. The other proposed algorithm is a distributed implementation of the VFA by itself. Both techniques are able to reach high levels of coverage and show a satisfying reactivity when the ZoI changes. This output parameter is measured as the capability for the sensors to "follow" a sequence of events happening in different ZoIs. The effectiveness of our techniques is shown through a series of simulations and comparisons with the classical centralized VFA.**

***Keywords*—Wireless Sensors Networks, Particle Swarm Optimization, Virtual Forces, Zone of Interest, Swarm Intelligence**

## I. Proposed Algorithms

Coverage is a very critical issue in Wireless Sensor Networks [7]. This section presents our distributed Virtual Force Algorithm (VFA-D) and our Serial Particle Swarm Optimization (PSO-S), by outlining differences in respect of both the centralized version and the original PSO, respectively.

### A. Distributed Virtual Force Algorithm: VFA-D

The VFA is based on the concept of the virtual forces field, and its main objective is the maximization of the coverage in a Wireless Sensor Network (WSN) [4]. In the same way electromagnetic particles attract or repel each other based on potential fields, sensors attract or repel each other based on their mutual distance. The total force, attractive and repulsive that all the $k$ nodes present in the field exert on node $i$, can be expressed by the following formula:

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

where $d_{ij}$ and $\alpha_{ij}$ 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 [2] from, $F_{xy}$ the magnitude of $\vec{F_i}$ and its x and y components, $F_x$ and $F_y$, as follows:

$$x_{new} = x_{old} + (\frac{F_x}{F_{xy}}) \cdot MaxStep \cdot e^{(\frac{-1}{F_{xy}})} \quad (2)$$

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

where $MaxStep$ is the predefined maximum moving distance. In the classical version of this algorithm, a central entity is required to collect all 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 complexity and the problems caused by a single point of failure, introduced with a central coordinator, we show that the VFA-C fails when ZoI change dynamically.

In this work we modified and implemented a distributed version of the VFA that does not require global information. VFA-D introduces a maximum distance $C$, related to sensing range of nodes, in order for the nodes to retrieve information. 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. Specifically, we assume that $C$ value is $4$ times greater than the sensing range of a node and is consequently related with the communication range, that is assumed to be twice the sensing range. The distributed version of the VFA approach requires a specific setup of the 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 for an heterogeneous scenario where sensor nodes with different sensing range 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 sensor nodes is the following:

$$\vec{F}_{i,j} = \begin{cases} 0 & \text{if } d_{ij} \geq C \\ (w_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } C > d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \left(w_R(\frac{1}{d_{ij}}), \alpha_{ij} + \pi\right) & \text{if } d_{ij} < d_{th} \end{cases} \quad (4)$$

### B. Serial Particle Swarm optimization Algorithm (PSO-S)

PSO is an extremely versatile technique of swarm intelligence based on particles [1]. The particles are localized inside a searching space and evaluate an objective function depending on their own position. The particles can also move around the searching space and combine their own knowledge with the data received from neighbors. By assuming that particles move in a 2D searching space, the velocity of the units will be computed iteration by iteration as:

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + \phi_p \cdot \vec{r}_p \circ (\vec{p}_i - \vec{x}_i(t)) + \phi_g \cdot \vec{r}_g \circ (\vec{p}_g - \vec{x}_i(t)) \quad (5)$$

where $x_i(t)$, $v_i(t)$, $p_i$ $p_g$, $r_p$ and $r_g$ are $R^2$ vectors. Specifically, $x_i(t)$ and $v_i(t)$ are the current position and the velocity of the particle $i$; $p_i$ is the best personal position of $i$, $p_g$ is the best position of the swarm; $r_p$ and $r_g$ are two random vectors in the domain $U(0,1)$; $w$, $\phi_p$ and $\phi_g$ 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*. The new updated position of $i$ at the next step is:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (6)$$

where the new position is given incrementally from the previous position when the new velocity has been applied in the time instant under observation.

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

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + \phi_p \cdot \vec{r}_p \circ (\vec{p}_i - \vec{x}_i(t)) + \phi_g \cdot \vec{r}_g \circ (\vec{l}_i) \quad (7)$$

where:

$$\vec{l}_i = \frac{\vec{x}_k - \vec{x}_i}{\|\vec{x}_k - \vec{x}_i\|} \cdot \frac{\|\vec{x}_k - \vec{x}_i\|}{d_{rep}} \quad (8)$$

In eq. (8), $x_k$ is the node position in the set of neighbors of $i$ that sensed the highest number of events in the previous iteration and $d_{rep}$ is a repulsive coefficient that avoids the overlap of nodes. The inertial weight $w$ varies between $w_{max}$ and $w_{min}$, as in [5]. This variant has been extensively simulated and the usage of local consensus has been introduced to give a different weight to each neighbor [6].

Serial Particle Swarm Optimization, PSO-S algorithm is designed by considering separately the local variant of the PSO and the VFA-D. Specifically, we first apply the local variant of PSO and when a sub-optimum solution is achieved, we apply the VFA to optimize 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 to be used:
1) node does not move significantly during last iteration (traveled distance smaller than 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 a certain number of iterations (number of iterations larger than the maximum number of iterations $it_{max}$).

TABLE I. EVALUATION PARAMETERS

| | |
|---|---|
| Field Area ($LxL$) | 100 $m$ x 100 $m$ |
| Number of Mobile Sensors ($N$) | 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 |
| Threshold Coverage ($c_{th}$) | $0.5 - 0.9$ |
| 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 for forces among nodes ($d_{th}$) | $2r_s$ |
| Threshold Distance of virtual force vanishing ($C$) | $4r_s$ |
| Repulsion Coefficient for PSO ($d_{rep}$) | $2r_s$ |
| Confidence Interval | 95% |
| Number of Runs | 100 |

When any of the previous conditions is verified the local variant of PSO stops running and the VFA-D enters into action. Specifically, the third condition is useful when nodes keep moving without finding a satisfying solution. We have to remark that all these conditions are verified distributedly by each node, therefore the algorithm does not need any centralized unit to run and terminate.

## II. THE SIMULATION ENVIRONMENT

In order to evaluate the effectiveness of our algorithms, a certain number of events occur simultaneously in a square sensor field of $100 \times 100$ where a fixed number of mobile sensors are randomly placed. 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 ZoI coverage with a minimum movement energy consumption. The energy model for the movement used for this work is taken from [3], it takes account of the nodes traveled distance by a constant $k$ that we set equal to $0.1J/m$. Relevant simulation parameters are summed up in Table I. From our first campaign of simulations used for tuning the parameters of the various algorithms, we found out that we obtain good results when repulsive forces are much bigger than attractive ones ($\omega_R >> \omega_A$). Since the communication cost is usually a very small fraction of the movement cost, it is not considered.Figure 1 show the simulated scenarios. The white zones represent the ZoIs, i.e. the areas where events happen and have to be monitored, the black zones are areas where no events occur. The choice of these specific scenarios is related to the capability of the proposed techniques to adequate in a dynamic fashion to many different situations and we will show that our algorithms are able to capture the events in a distributed fashion.

## III. SIMULATION RESULTS

We show the behaviour of VFA-C, the combined version of the local variant of PSO and the VFA-D (PSO-S), by considering 1) the coverage, as the fraction of ZoI covered by sensors in order to see the effectiveness of the tested

algorithms, 2) the energy consumed by nodes movement, which represents the cost of the algorithms. As we assume a probabilistic model for Virtual Forces, 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:

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

where $r_e$ is the measure of detection uncertainty, $\lambda_1 = r_e - r + d(s_i, P)$ and $\lambda_2 = r_e + r - d(s_i, P)$, $\alpha_1$, $\alpha_2$, $\beta_1$ and $\beta_2$ are detection probability parameters. The values of $\alpha_1$, $\alpha_2$, $\beta_1$ and $\beta_2$ depend on the sensor characteristics. In practice, the probabilistic detection model consider a possible areas overlapping to compensate the low detection probability in the area. Let $S_{ov}$ be a set of sensors that "overlap" a certain point with coordinates $(x, y)$. The detection probability that a point can be successfully detected by at least one sensor is:

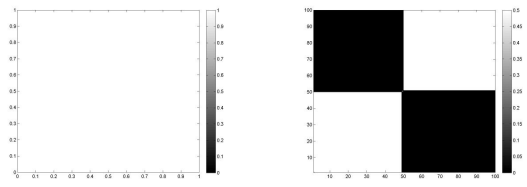
$$c_{x,y}(S_{ov}) = 1 - \prod_{s_i \in S_{ov}} (1 - c_{x,y}(s_i)) \tag{10}$$

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

Performance parameters are evaluated by considering the threshold value $c_{th}$ ranging from $0.5$ and $0.9$. It is worth recalling that the algorithms terminate for any of the three different conditions enumerated in Section I-B. We present results obtained by varying the coverage threshold between 0.5 and 0.9, and we fix the number of sensors to 80.

We assume that events in the sensor field are initially distributed as shown in Figure 1 (a) and, at a random instant during the simulation, the events disappear from the initial scenario and reappear as in the scenario 2 of Figure 1. All the results have been statistically averaged on a confidence interval of 95%. An increase of the coverage threshold for the same number of sensor means, in general, a lower probability to consider the point as covered.

The following figures show the impact of concentrating the ZoIs in two areas (scenario 2) from an initial situation of spread events (scenario 1). The behavior in terms of coverage is simple to read. Figure 2 shows that the coverage decreases for all 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.



(a) (b)

Fig. 1. Simulated scenarios, events occur in white areas. (a) Scenario 1: events are uniformly distributed, (b) Scenario 2: events are concentrated in two squares.

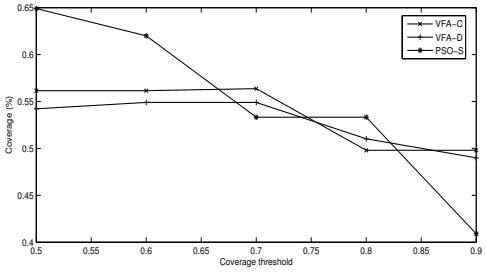



Fig. 2. Coverage achieved when Scenario 2 changes to Scenario 4.

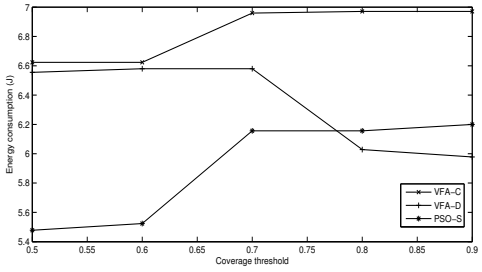



Fig. 3. Energy spent when Scenario 2 changes to Scenario 4.

It is important to notice in Figure 3 that even though PSO-S does not achieve the best coverage performance for all the thresholds, still it consumes the smallest amount of energy to achieve a stable placement. Only the VFA-D consumes less when the coverage threshold is very high, whereas the VFA-C is the worst algorithm in terms of consumed energy, even though it shows the same coverage of the distributed version.

## REFERENCES


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

[2] S. Li, C. Xu, W. Pan and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," Information Fusion, Inter. Conference on, vol.2, pp. 7, 25-28, 2005

[3] P. A. Tipler, "Physics for Scientists and Engineers", Worth Publishers, 1991.

[4] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces, " in Joint Conf. of the IEEE Computer and Communications, INFOCOM, pp. 1293-1303, vol.2, 2003.

[5] Shi, Y., Eberhart, "R. C. Empirical study of particle swarm optimization," In ECTA, vol 3, (1999) 101- 106, 1999.

[6] V. Loscrí, E. Natalizio, F. Guerriero and G. Aloi, "Particle Swarm Optimization Schemes Based on Consensus for Wireless Sensor Networks," in the Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (PM2HW2N '12), pp. 77–84, 2012.

[7] T. Razafindralambo; N. Mitton; A. C. Viana; M. D. De Amorim; K. Obraczka "Adaptive Deployment for Pervasive Data Gathering in Connectivity-Challenged Environments," in the Proceedings of the Eighth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM), March 2010.