# Neuro-Dominating Set Scheme for a Fast and Efficient Robot Deployment in Internet of Robotic Things

Cristanel Razafimandimby, Valeria Loscrì, Anna Vegni, Abderrahim Benslimane

# Neuro-Dominating Set Scheme for a Fast and Efficient Robot Deployment in Internet of Robotic Things

Cristanel Razafimandimby[a,1], Valeria Loscrí[a], Anna Maria Vegni[b], Abderrahim Benslimane[c]

[a] *Inria Lille - Nord Europe, France*
[b] *Department of Engineering, Roma Tre University*
*COMLAB Telecommunications Laboratory, Rome, Italy*
[c] *CERI/LIA University of Avignon, Avignon, France*

---

**Abstract**

Internet of Robotic Things (IoRT) is a new concept introduced for the first time by ABI Research. Unlike the Internet of Things (IoT), IoRT provides an active sensorization and is considered as the new evolution of IoT. In this context, we propose a Neuro-Dominating Set algorithm (NDS) to efficiently deploy a team of mobile wireless robots in an IoRT scenario, in order to reach a desired inter-robot distance, while maintaining global connectivity in the whole network. We use the term Neuro-Dominating Set to describe our approach, since it is inspired by both neural network and dominating set principles. With NDS algorithm, a robot adopts different behaviors according whether it is a dominating or a dominated robot. Our main goal is to show and demonstrate the beneficial effect of using different behaviors in the IoRT concept. The obtained results show that the proposed method outperforms an existing related technique (*i.e.*, the Virtual Angular Force approach) and the neural network based approach presented in our previous work. As an objective, we aim to decrease the overall traveled distance and keep a low energy consumption level, while maintaining network connectivity and an acceptable convergence time.

*Keywords:* Neural network, Dominating Set, Heterogeneity, Multi-robot systems

---

## 1. Introduction

Nowadays, Internet of Things (IoT) technology begins to take an important place in economic systems and in society daily life [1]. It has got a large success in several application areas [2, 3, 4], ranging from smart city [5, 6], Industry 4.0 [7], to smart grid applications [8]. However, most of actuated devices constituting IoT paradigm are only passive so far. Adding an active role for these actuated devices will be needed, in order to optimize the systems where they are present. Willow Garages PR2 robot [9] and Wifibot robot [10] are a good example of dynamic actuated devices.

Robotic systems match very well to this new need, since robots can sense and interact with the environment. Therefore, ABI Research introduced a new concept called Internet of Robotic Things

*Email addresses:* jean.razafimandimby_anjalalaina@inria.fr (Cristanel Razafimandimby), valeria.loscri@inria.fr (Valeria Loscrí), annamaria.vegni@uniroma3.it (Anna Maria Vegni), abderrahim.benslimane@univ-avignon.fr (Abderrahim Benslimane)

[1] Corresponding author.

*Preprint submitted to Elsevier* *August 29, 2018*

(IoRT) [11], defined as an intelligent set of devices that can monitor events, fuse sensor data from a variety of sources, use local and distributed intelligence to determine a best course of action, and then act to control or manipulate objects in the physical world. This new concept is expected to be the evolution of Internet of Things (IoT) and robotics [12].

In various IoRT applications (*e.g.*, smart agriculture, smart environment monitoring, smart exploration, smart disaster rescue, etc.), the use of mobile robots' teams brings many advantages over one powerful IoRT device. As a matter of fact, a team of robots can accomplish tasks more efficiently, faster and more reliable than a single robot [13, 14, 15]. To carry out cooperative tasks, IoRT team members need to communicate with each other, often via a wireless link *i.e.*, Wifi, and Bluetooth. Maintaining communication and connectivity among multiple mobile IoRT robots is therefore a crucial issue.

In our previous work [16], we proposed a distributed artificial neural network based approach (namely, ANN-based), which allows the robots to reach the desired inter-robot distance and desired communication quality, while ensuring global connectivity. Our ANN-based approach has been trained from a set of data obtained by using virtual force based approach. We compared performance of the ANN-based approach with a virtual force approach, and showed as the ANN-based is more efficient in achieving final targets. However, robots were endowed with the same algorithm and behavior to achieve these objectives.

In addition, one of the most recent approaches using VFA, that we can compare to, is in [17]. In [17], Casteigts *et al.* propose the Virtual Angular Force (VAF) technique to biconnect [2] a fleet of mobile robots. VAF is a mix of virtual force and angular force algorithms, so that the virtual force is used to regulate the distance between two nodes, while the angular force is used to regulate the angle between a node and its two consecutive neighbors in order to stabilize as an equilateral triangle.

In this paper, we go a step further than our previous work [16] by studying the beneficial effect of using different behaviors in the IoRT concept. We propose a Neuro-Dominating Set (NDS) approach for global connectivity maintenance and robot motion control. We use the term Neuro-Dominating Set to describe our approach, since it is inspired by both neural network and dominating set strategies. Each IoRT robot [3] adopts a different strategy according whether it is a dominating robot or a dominated robot. This heterogeneity of strategies may improve global efficiency (expressed in terms of minimization of traveled distance by each robot), while keeping more or less similar convergence time with respect to the ANN-based approach [16]. Finally, with the aim of providing a more complete assessment analysis, we compare our technique with VAF algorithm, since it is one of the most representative approaches of virtual and angular force based techniques.

The rest of this paper is organized as follows. Section 2 provides some backgrounds and explains our motivations behind this paper. In Section 3 we detail the theoretical analysis and proof of our proposed Neuro-Dominating Set algorithm. In Section 4 we provide the simulation results expressed in terms of robot traveled distance, the average distance between any pair of robots, and the QoS level achieved. A comparison with a related technique based on virtual forces [17] and with our previous approach [16] is also carried out. Finally, conclusions are drawn at the end of the paper.

[2] A network is said to be biconnected if it is still connected after any of the nodes is removed or failed [17].
[3] In this paper, the terms *IoRT robot* and *robot* will be used interchangeably.

## 2. Background and Motivations

Biological societies show various example of diversity allowing participants to self organize and solve global problems in a more efficient way [15, 18, 19]. The use of this diversity –or heterogeneity– in IoRT context may therefore open possibilities to solve more complex tasks, since different skills and behaviors can be combined together. In the literature, the definition of the *heterogeneity* often varies according to used applications [20, 21, 22]. However, heterogeneity can be defined in terms of variety in robot capacity, hardware, size, cognition, behavior, etc.

In this paper, heterogeneity refers to difference in behavior since we adopt different roles among robots to enable them better achieve the global system performance. It follows that our main goal in this work is to reduce the overall traveled distance (and hence the energy consumption), while maintaining network connectivity and an acceptable convergence time. The robots deployment should be therefore efficient in terms of energy consumption since robots only work under battery.

Many approaches have been designed to maintain the connectivity in multi-robot systems [16, 17, 23, 24, 25, 26, 27, 28, 29, 30]. All these approaches can be classified into two groups according to the degree of *connectivity maintenance i.e.* (*i*) local, and (*ii*) global. With the local connectivity maintenance, the initial set of edges which define the connectivity graph must be always preserved over time. Unlike local connectivity maintenance, global connectivity maintenance allows suppression and creation of some edges, as long as the overall connectivity of the graph is conserved.

In Multi-Robot Systems (MRS), global connectivity maintenance is often used since the preservation of each local communication link in the network is a very restrictive requirement which significantly limits the capability of the MRS itself [24]. This is the main reason why we focused on global connectivity maintenance approach in our previous work [16].

Besides the connectivity maintenance, ensuring the *collective coverage* is also important to meet the application requirements. In the literature, coverage and connectivity issues are treated separately in general, mainly due to their antagonistic property. However, there are some approaches that could capture a trade-off between these two properties based on a certain relationship between the communication range (*i.e.*, $R$ [m]) and sensing range (*i.e.*, $r$ [m]) [31, 32, 33, 34]. Other approaches like ours do not need that kind of relationship to work.

To provide a full coverage, while maintaining the network connectivity (based on the relationship between $R$ and $r$ or not), many deployment algorithms can be used. These algorithms can be classified into three different categories *i.e.*, (*i*) virtual force approaches [16, 17, 35, 36, 37] (*ii*) geometrical approaches [38, 39, 40] and (*iii*) grid-based approaches [41, 42, 43].

Virtual force approaches use repulsive and attractive forces to move the nodes toward or away from each other in order to meet the full coverage. Nodes will continue to move until convergence (*i.e.* steady state) is reached. They are often used to enhance the coverage after a random deployment in a given area. Geometrical approaches provide a geometric computation in order to detect coverage holes in the region of interest (ROI). In these approaches, the geometric computation can only be done when the global location information of all the nodes in the network is known. Examples of geometry-based approaches include Voronoi diagram and Delaunay triangulation. Finally, grid-based approaches are focused on determining the location of nodes using a special grid pattern. The commonly used patterns are triangular lattice, square grid, hexagonal grid and Honeycomb gird. In these approaches, the ROI is divided into cells and according to the used strategy, the nodes are positioned either in the cell vertices or at the center of the cell. Grid-based approaches provide therefore also a way to compute the effectiveness of network coverage during deployment time.

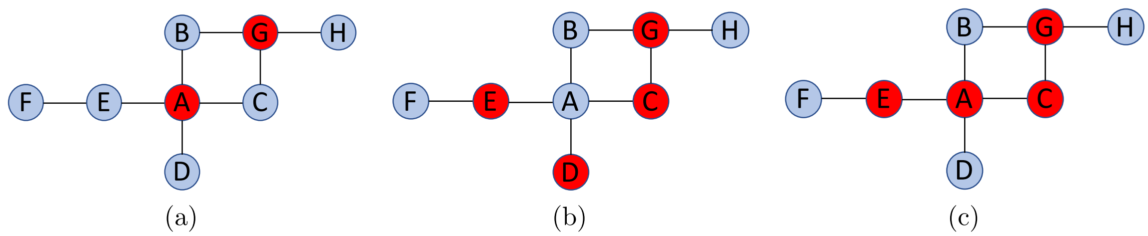



Figure 1: Dominating (*red vertices*) and dominated (*blue vertices*) nodes in different representations of dominating sets *i.e.*, (a) Undominating Set (US), (b) Dominating Set (DS), and (c) Connected Dominating Set (CDS).

In this work, we focus only on Virtual Force Approaches (VFA). Specifically, we consider a Multiple Mobile Robots (MMR) network composed of robots that are able to perform exploration of unknown environments. This type of activity is fundamental for several real-world applications such as search and rescue, map building, access to a toxic zone, etc. [44].

The deployment of MMRs for exploring purpose presents numerous benefits, like for instance the whole system is more robust, and the exploration of the unknown area is more efficient. In this paper we will consider a MMR network for *exploratory* and *coordination* activities in order to answer to a main question: "*who goes where?*". In order to properly answer to this question, we will rely on two important tools, *i.e.*, (*i*) the concept of Dominating Sets in graph context with the aim of differentiating the robots and minimizing the traveled distance in order to enlarge as much as possible the coverage area, and (*ii*) Artificial Neural Network, in order to let the nodes learning information about the new position towards those nodes that have to move to keep global connectivity in the network.

Our MMR network can be modeled as a graph $G(V, E)$, where $V$ is the set of vertices representing each robot and $E \subseteq V^2$ is the set of edges. $E$ can be defined as:

$$E = \left\{(i,\, j)\, \in V^2 \mid\, i \neq j \,\wedge\, d(i,j) \leq R\right\}, \tag{1}$$

where $d(i,j)$ is the euclidean distance between the $i$-th and $j$-th robot, and $R$ [m] is the communication range. Following the above definition, let $N_i$ be the one-hop neighborhood of the $i$-th robot. Thus, $N_i$ is the set of robots that can exchange information with the $i$-th robot via a direct link, *i.e.*,

$$N_i = \{j \in V \mid d(i,j) \leq R\}\,. \tag{2}$$

The graph $G(V, E)$ may evolve over time due to the robots' motion but has to be always connected.[4]

### *2.1. Dominating Sets in Graphs*

By definition, a set $D \subseteq V$ of vertices in a graph $G(V, E)$ is called a Dominating Set (DS) if every vertex $v \in V$ is either an element of $D$ or has a neighbor in it [45]. For instance, Figure 1 (a) represents an undominating set, since not all the vertexes $v \in V$ have a neighbor in $D$ (*e.g.*, the node $F$ does not have a neighbor in $D$). The nodes in a DS are called *dominating* nodes (*i.e.*,

[4]An undirected graph $G$ is connected if there exists a path between each pair of vertices.

the red nodes in Figure 1), the others are called *dominated* (*i.e.*, the blue nodes in Figure 1). This kind of set is not necessarily connected (see Figure 1 (b)). We refer to a Connected Dominating Set (CDS) when the subgraph induced by $D$ is connected, as represented in Figure 1 (c). On every connected graph $G(V, E)$ it can be found at least one DS since the set of all vertices is dominating according to the definition.

In this work, we compute locally DS, while trying to minimize its size. The computation of DS is also periodically executed in order to take into account the variation of the graph $G(V, E)$ over time. Notice that the computation of minimum dominating set (MDS) is an NP-hard problem. In the literature several methods proposed an approximation version of MDS [46, 47]. In this work, we use the distributed greedy algorithm to approximate MDS; this is not the most efficient in terms of complexity but it represents the simplest approach to implement. The aim of this paper is to show that, with the help of the dominant set approach, we can reduce the distance traveled by a single robot, and then the energy consumption.

### 2.2. Artificial Neural Networks

Artificial neural network (ANN) was inspired by the human brain and was designed as a computational model to solve specific problems. Its architecture is defined by ($i$) basic processing elements called artificial neurons, and ($ii$) the way in which they are interconnected. The output value of a single neuron is given by:

$$output = f\left(\sum_i w_i x_i + b\right) = f\left(\mathbf{W}^T\mathbf{X} + \mathbf{b}\right), \tag{3}$$

where $x_i$ are the inputs, $w_i$ are the connections' weights between $x_i$ and the neuron, $\mathbf{W}$ the weights' vector, $\mathbf{X}$ is the inputs' vector, $\mathbf{b}$ is the vector of bias, and $f$ is the activation function.

The basic architecture of ANN contains three neuron layers *i.e.*, input layer, hidden layer, and output layer. In this case, the outputs of one layer become the inputs of next layer [48]. A key element of an artificial neural network is its ability to learn. This means that ANN has to learn from a data set in order to match the inputs to the desired output. During the learning process, weights and biases are adjusted till the desired output will be reached. There are several learning algorithm but in this paper back-propagation algorithm [49] will be used.

In our previous work [16], we proposed a distributed trained neural network in order to ($i$) maintain the global connectivity, and ($ii$) control the robots' motion. ANN-based approach was trained from a set of data obtained through the centralized IoT-based approach proposed in [16]. The trained ANN is constituted by 2 input units, and 1 output unit. The two input units are $d(i, j)$ and $\theta_{ij}$, while the output is $\overrightarrow{P_{ij}}$ (see Table 1).

Algorithm 1 summarizes the ANN-based approach. First, the $i$-th robot needs to know its one-hop neighbor (*i.e.*, Phase I). Then, it computes the force $\overrightarrow{P_{ij}}$ that it has to exert with its neighbor $j$ by using the trained ANN (*i.e.*, Phase II). If the robot $i$ has many neighbors (*i.e.*, $> 2$), its new position $\overrightarrow{P_i}$ is calculated as the summation of the forces with respect to all the neighbors (*i.e.*, Phase III), *i.e.*,

$$\overrightarrow{P_i} = \sum_{j \in N_i} \overrightarrow{P_{ij}}. \tag{4}$$

Finally, each robot $i$ moves to its computed new position. Further details on this approach can be found in our previous work [16].

| | |
|---|---|
| Inputs | $\theta_{ij}$: the orientation of the line segment from the $i$-th and the $j$-th robots |
| | $d(i,j)$: the euclidean distance between the $i$-th and $j$-th robot |
| Output | $\overrightarrow{P_{ij}}$: vector position that dictates the force between the $i$-th and the $j$-th robot |

Table 1: Inputs and output of the ANN-based approach.

**Algorithm 1** ANN-Based approach (runs every $t$ units of time)

**Phase I: Neighbor Discovery**

MyNeighbor = FindNeighbor(RobotId)

**Phase II: Estimate the position $\overrightarrow{P_{ij}}$ between two robots**

$\overrightarrow{P_{ij}} = trained\_ann(d(i,j), \theta_{ij})$

**Phase III: Compute the new position $\overrightarrow{P_i}$**

$\overrightarrow{P_i} = \sum_{j \in N_i} \overrightarrow{P_{ij}}$

**Phase IV: Deployment**

move to $\overrightarrow{P_i}$

## 3. Neuro-Dominating Set schemes

It is easy to observe that if a subset of robots in the network moves less than the other robots, the global traveled distance will be decreased, and hence the energy consumption as well. In order to benefit from this effect, we need to exploit the good properties of Dominating Set.

Table 2 collects the main variables used in this section. Let us consider a robot network that can be decomposed into two subsets *i.e.*, ($i$) a dominating set and ($ii$) a dominated set. Let $A$ be the dominating set and $m = \|A\|$ its cardinality (respectively, $B$ the dominated set and $l = \|B\|$ its cardinality). If we have $n$ robots in the network and the computed dominating set has a minimum size, then the following formulas are always true:

$$n = m + l, \tag{5}$$

and

$$m \leq l. \tag{6}$$

Now, let $Dist_{Tot}(t)$ be the total distance traveled by all robots in the network at time $t$.

Table 2: Main variables used in the Neuro-Dominating Set theory.

| Variable | Definition |
|---|---|
| $n$ | Number of robots |
| $m$ | Number of nodes in the dominating set |
| $l$ | Number of nodes in the dominated set |
| $d_i$ | Distance traveled by the $i$-th node |
| $d_a$ | Distance traveled by a dominating robot |
| $d_b$ | Distance traveled by a dominated robot |
| $\alpha$ | Reduction factor |
| $\overrightarrow{P}_i$ | Position vector of the $i$-th node w.r.t. its neighbors |

$Dist_{Tot}(t)$ can be defined as:

$$\begin{aligned} Dist_{Tot}(t) &= d_1(t) + d_2(t) + ... + d_n(t) \\ &= \|\overrightarrow{P_1}(t)\| + \|\overrightarrow{P_2}(t)\| + ... + \|\overrightarrow{P_n}(t)\| \\ &= \sum_{i=1}^{n} \|\overrightarrow{P_i}(t)\| = \sum_{i=1}^{n} d_i(t), \end{aligned} \tag{7}$$

where $d_i(t)$ is the traveled distance of the $i$-th robot at time $t$, and $\overrightarrow{P_i}(t)$ is the position vector of the $i$-th robot with respect to all its neighbors. It follows that $d_i(t) = \|\overrightarrow{P_i}(t)\|$.

Considering that some robots are dominating and some others are dominated, we have:

$$Dist_{Tot}(t) = \sum_{a\,in\,A} d_a(t) + \sum_{b\,in\,B} d_b(t), \tag{8}$$

where $d_a(t)$ is the traveled distance of a dominating robot $a$ at time $t$, and $d_b(t)$ is the traveled distance of a dominated robot $b$ at time $t$. It follows that two possible approaches can be deduced from Eq. (8) *i.e.*, ($i$) dominating robots move less distance than dominated robots so that

$$\sum_{a\,in\,A} d_a(t) < \sum_{b\,in\,B} d_b(t), \tag{9}$$

and ($ii$) the dominated robots move less distance than dominating robots *i.e.*,

$$\sum_{b\,in\,B} d_b(t) < \sum_{a\,in\,A} d_a(t). \tag{10}$$

Let us call NDS-A the first approach (*i.e.* dominating robots move less distance than dominated robots) and NDS-B the second approach.

The total traveled distance using NDS-A approach can be defined as:

$$Dist_{Tot_A}^{NDS-A}(t) = \alpha \sum_{a\,in\,A} d_a(t) + \sum_{b\,in\,B} d_b(t), \tag{11}$$

while, the total traveled distance using NDS-B approach is defined as:

$$Dist_{Tot_B}^{NDS-B}(t) = \sum_{a\,in\,A} d_a(t) + \alpha \sum_{b\,in\,B} d_b(t), \tag{12}$$

where $\alpha \in [0, 1]$ is a reduction factor. For $\alpha < 1$, the dominated robots (*i.e.*, the set of nodes in NDS-A) travel less distance in respect of nodes belonging to the set NDS-B. On the other hand, for $\alpha = 1$, the NDS algorithm is equivalent to the ANN approach.

To know which approach is the most efficient in term of minimization of traveled distance, let us subtract Eq. (12) and (11), *i.e.*:

$$\begin{aligned} Dist_{Tot_B}^{NDS-B}(t) - Dist_{Tot_A}^{NDS-A}(t) = \\ = (1-\alpha)\cdot\left[\sum_{a\,in\,A} d_a(t) - \sum_{b\,in\,B} d_b(t)\right]. \end{aligned} \tag{13}$$

As $(1-\alpha) \geq 0$, we will focus our study on $\sum_{a\,in\,A} d_a(t) - \sum_{b\,in\,B} d_b(t)$. We assume that all the robots have the same velocity either they are dominating or dominated. If $a \in A$ is a dominating robot, it is connected at least to one robot in $B$. Therefore, $d_a$ is a magnitude of a resultant force generated by one or more robot in $B$. This means that:

$$\sum_{a\,in\,A} d_a(t) \leq \sum_{b\,in\,B} d_b(t). \tag{14}$$

Therefore, we get:

$$Dist_{Tot_B}^{NDS-B}(t) \leq Dist_{Tot_A}^{NDS-A}(t). \tag{15}$$

The above formula proves that better efficiency, in term of minimization of traveled distance, is obtained when NDS-B approach is used. In other word, the global traveled distance of our robot network will improve if the dominated robots move less distance than the dominating robots.

### *3.1. NDS algorithm*

In this work, we consider that each robot aims to reach the desired inter-robot distance and the desired communication quality, while ensuring global connectivity in the whole system. As mentioned before, our goal is to decrease the overall traveled distance while maintaining network connectivity and an acceptable convergence time. In this paper, we state that an algorithm converges if after some iterations the euclidean distance between any pair of robots is equal to a desired distance $D_{th}$. NDS algorithm is a mix of ANN-based approach and dominating set strategy. In this algorithm, NDS-B approach will be used since its effectiveness with respect to NDS-A has already been proved in the previous section. In this case, the dominated robots have to travel less distance than the dominating robots.

The major steps of the chosen NDS-B algorithm that run in each robot are enlisted as follows *i.e.*, (*i*) neighbor discovery, (*ii*) computation of the dominating set, (*iii*) computation of the vector position between two neighboring robots, (*iv*) computation of the robot new position, and (*v*) movement towards the computed position.

Algorithm 2 summarizes our proposed NDS-B approach. First, the $i$-th robot needs to know its one-hop neighbors (*i.e.*, Phase I). Then, in Phase II it computes the MDS through a distributed greedy algorithm that approximates the MDS. The distributed greedy algorithm was chosen due

**Algorithm 2** Neuro-Dominating Set approach (runs every $t$ units of time)

**Phase I: Neighbor Discovery**

MyNeighbor = FindNeighbor(RobotId)

**Phase II: Compute the Dominating Set**

DS = GetDS()

**Phase II: Compute the position $\overrightarrow{P_{ij}}$ between two robots**

**if** (RobotId $\in$ DS) **then**
  $\overrightarrow{P_{ij}} = trained\_ann(d(i,j), \theta_{ij})$
**else**
  $\overrightarrow{P_{ij}} = \alpha * [trained\_ann(d(i,j), \theta_{ij})]$
**end if**

**Phase III: Compute the new position $\overrightarrow{P_i}$**

$\overrightarrow{P_i} = \sum_{j \in N_i} \overrightarrow{P_{ij}}$

**Phase IV: Deployment**

move to $\overrightarrow{P_i}$

to its simplicity. However, faster algorithms such as in [46, 47, 50] can be used for the MDS approximation. The slowest of these algorithms runs in $O(\log n \log \Delta)$ rounds with high probability, where $n$ is the number of nodes and $\Delta$ is the maximal degree of the graph $G$.

The computation of the MDS is NP-hard, therefore a lot of works in the literature proposed an approximation version of MDS. The aim here is to show that, with the help of the dominant set approach, we can reduce the distance traveled by a robot. After identifying the MDS, the $i$-th robot computes the force $\overrightarrow{P_{ij}}$ that it has to exert with its $j$-th neighbor by using a trained ANN (*i.e.*, Phase III). The computation of $\overrightarrow{P_{ij}}$ is different according whether the $i$-th robot is dominating or dominated. If the $i$-th robot has many neighbors, its new position $\overrightarrow{P_i}$ is calculated according to Eq. (4). Finally, the $i$-th robot moves to the computed position (*i.e.*, Phase IV). It is worth to mention that each robot knows its own position by using GPS (in case of outdoor environments) or other localisation systems (for indoor environments).

Finally, we point out that for what concerns the ANN algorithm, the complexity is $O(|n|)$, where $n$ is the number of neighbors. For the dominating set part, the distributing greedy algorithm computes a $\ln \Delta$-approximation for the minimum dominating set problem in $O(n)$ rounds, where $\ln \Delta$ is the performance ratio of the approximation. We recall that a dominating set $D$ is a $f$-approximation (with $f \in \mathbb{R}^+$) to a minimum dominating set $S$ if $|D|/|S| \leqslant f$.

## 4. Performance Evaluation and Simulations

In this section, the proposed approach will be assessed in outdoor environment, although a validation in indoor scenarios can be also provided by means of collaborative localisation tech-

| | | |
|---|---|---|
| Physical | Propagation | Two ray ground |
| | Error model | Real |
| | Antennas gain | $G_{Tx} = G_{Rx} = 1$ |
| | Antennas height | $h_{Tx} = h_{Rx} = 1$ m |
| | Communication range | 250 m |
| Topology | Width $\times$ height | $3 \times 3$ km$^2$ |
| | Number of robots | $[5, 70]$ |
| Statistics | Number of samples | 100 |
| | Simulation time | 3000 s |
| | Confidence Interval | 95% |

Table 3: Simulation parameters, [51].

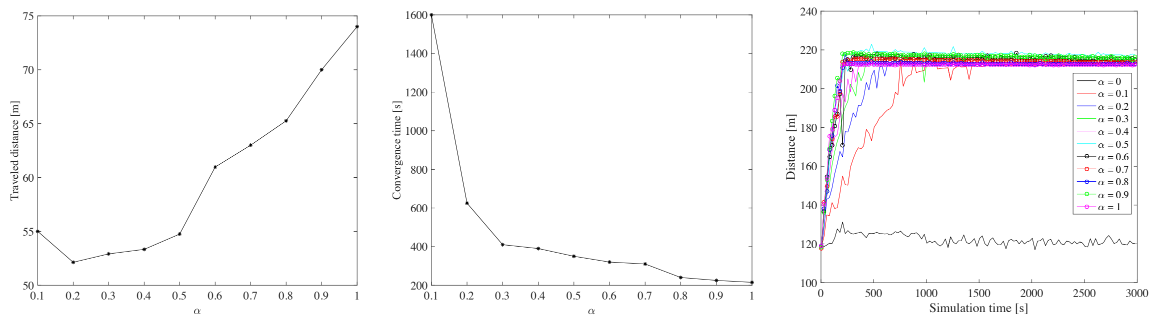
| | | |
|---|---|---|
| Mobility | Computation of the new position | see Algorithm 2 |
| | Reduction factor, $\alpha$ | $[0, 1]$ |
| | Desired distance, $D_{th}$ | 212 m |
| ANN | Layer number | 4 |
| | Input number | 2 |
| | Output number | 1 |
| | Neuron number in hidden layers | 15 |
| | Desired Error | 0.00001 |
| | Max epochs | 10000 |
| | Activation function | Sigmoid symmetric |
| | Learning rate | 0.2 |
| | Training algorithm | Backpropagation |

Table 4: Algorithm parameters.

niques [52, 53, 54, 55]. Beacon messages are used to allow robots exchange their positions with their one-hop neighbors.

We provide simulation results of our algorithm, and we are interested in studying how the proposed technique converges to the desired distance $D_{th}$ between any pair of robots. The influence of the parameter $\alpha$ in our proposed approach will be also highlighted. A comparative analysis of our technique w.r.t the ANN-based [16] and VAF [17] approaches will be described. Specifically, we will assess our algorithm with respect to ($i$) the minimized robot traveled distance, ($ii$) the average distance between any pair of robots, and ($iii$) the QoS level expressed in terms of RSSI (Received Signal Strength Indicator). Simulations have been carried out in NS2 for a variable number of robots (*i.e.*, ranging from 5 to 70 robots) in an outdoor area of $3 \times 3$ km$^2$. Notice that the presence of obstacles along the traveled path of the robots is not an issue since our robots (*e.g.*, WiFibots and turtlebots) are equipped with ultrasound sensors allowing them to overcome the obstacle and turning around.

All the algorithms in this paper have been implemented in version 2.29 of Network Simulator with patch from [51] that reflects a realistic channel propagation and error model. This patch is used in order to provide the effect of interference and different thermal noises to compute the signal-to-noise plus interference ratio (SINR) and accounting for different bit error rate (BER) to




(a) (b) (c)

Figure 2: Performance assessment in terms of (a) traveled distance, (b) convergence time, and (c) variation of robot position, versus $\alpha$, in a network scenario comprised of 15 robots.

SINR curves for the various codings employed [56]. Table 3 summarizes the parameters used in the simulations. All of the obtained results are the average of 100 times simulations and we assume that the topology is totally connected at the beginning of each simulation. According to [17], in order to meet the full coverage deployment, the only constraint is that $D_{th} \leq\sim 0.851 \cdot C_R$. In this paper, we set $D_{th}$ to the upper bound *i.e.*, $D_{th} = 0.851 \cdot C_R$, where $C_R$ [m] is the communication range of a single robot.

The main simulation parameters are reported in Table 3, while, the main configuration parameters of the algorithm are summarized in the table 4.

The initial position of each robot is known through GPS or other localisation systems, and it is set randomly. A synchronous model for communications is used, that is, in every communication round, each robot is allowed to send a probe packet to each of its direct neighbors in the network. In this work, we assume also that each robot decides to move independently of the other robots. Each robot moves only on the basis of its local neighborhood knowledge.

The discount factor $\alpha$ is one of the most important parameters of our proposed algorithm, since its value affects the efficiency improvement. If $\alpha$ is small, the traveled distance decreases but the convergence time increases accordingly since the system needs more time to converge in this case. According to Figure 2, we can say that for $\alpha \to 1$, the traveled distance increases, while the convergence time reduces to minimum value (*i.e.*, 200 s). Furthermore, in Figure 2 (c), for $\alpha$ equal to zero, the algorithm does not even allow the robots to converge to the desired distance $D_{th}$ (*i.e.*, 212 m). Notice that the choice of evaluating $\alpha$ on a network of 15 robots was arbitrary. We can actually evaluate it on different number of robots but the conclusion will remain the same, *i.e.*, a small value of $\alpha$ decreases the traveled distance but increases the convergence time. As a result, the value of $\alpha$ has to capture the trade-off between the traveled distance and the convergence time.
Thus, for the purposes of our evaluation, $\alpha$ is set to 0.7 since after the sensitivity analysis we found that this value captures well the trade-off between the traveled distance and the convergence time.

Figure 3 depicts the performance comparison –expressed in terms of traveled distance– of the proposed approaches with respect to ANN [16] and VAF [17]. We observe that whatever the used approach and when the number of robots is less than 60, the distance traveled by the robots is proportional to the number of robots in the networks. Therefore, the average traveled distance of a robot is sensitive to the number of robots in all these algorithms. This can be explained by

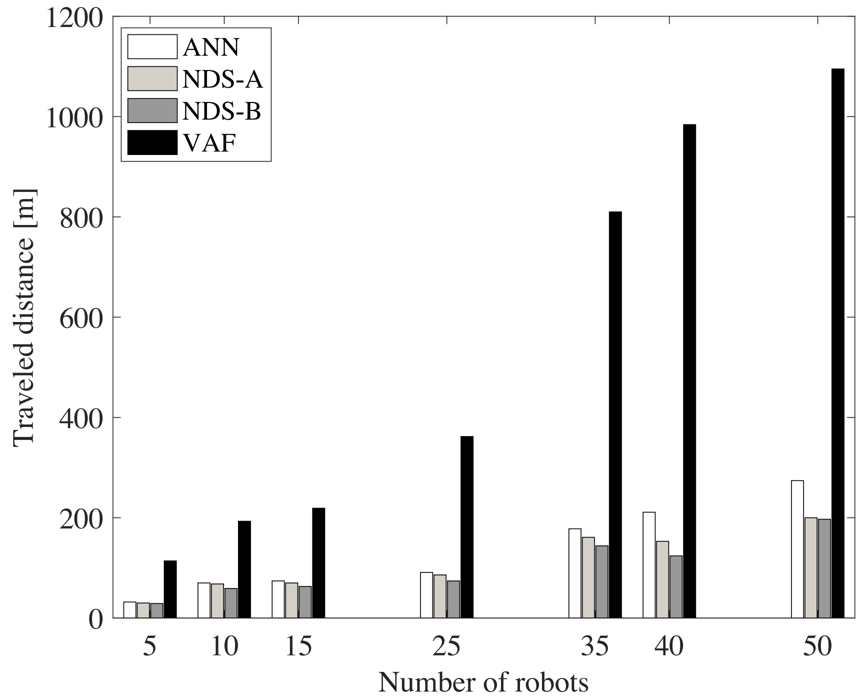



Figure 3: Traveled distance according to the number of the robots, for different approaches.

the fact that a robot is expected to have more neighbors when the number of robots increases. This property is no longer valid when the density of the robots is quite high (*i.e.*, specifically for a number of robots higher than 50). In this case, there will be a strong collision and the knowledge of the neighborhood will not be exact. From Figure 4 we can notice a false information about the neighborhood when the number of the robots is more than 50. The average number of neighbors for 60 and 70 robots is less than 2, which is completely false.

In the following, we present simulation results for different robot networks *i.e.*, ranging from 10 (Figure 5) to 70 robots (Figure 8). We aim to assess the effectiveness of our proposed NDS approaches with respect to ANN and VAF. As stated before, results are expressed in terms of ($i$) average traveled distance by a single robot, ($ii$) inter-robot reciprocal distance for a desired threshold $D_{Th}$, and ($iii$) power level, meaning the maintained QoS level, versus the simulation time.

### 4.1. Average traveled distance

In Figure 5 (a) we notice that our algorithms decrease considerably the average distance traveled by a robot, reaching a maximum value of $\approx$ 60 m. On the other hand, high values of traveled distance are obtained by VAF approach, which shows an increasing slope, and then it is stable at $\approx$ 190 m. Similar performances are observed in Figure 6 (a) and Figure 7 (a), where NDS-A and NDS-B present lower values of traveled distance than ANN and VAF. Obviously, due to the

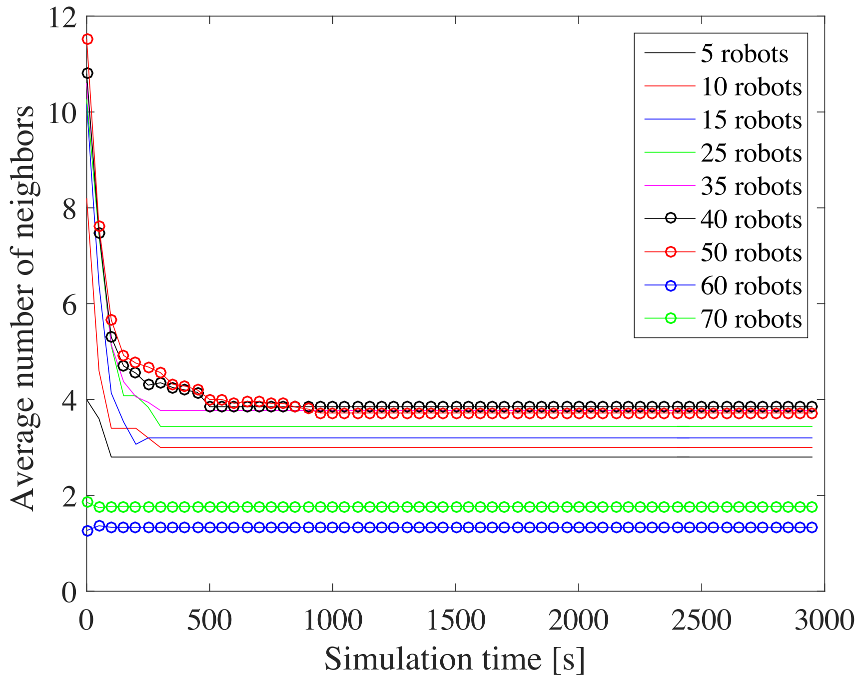



Figure 4: Average number of neighbors for different number of robots.

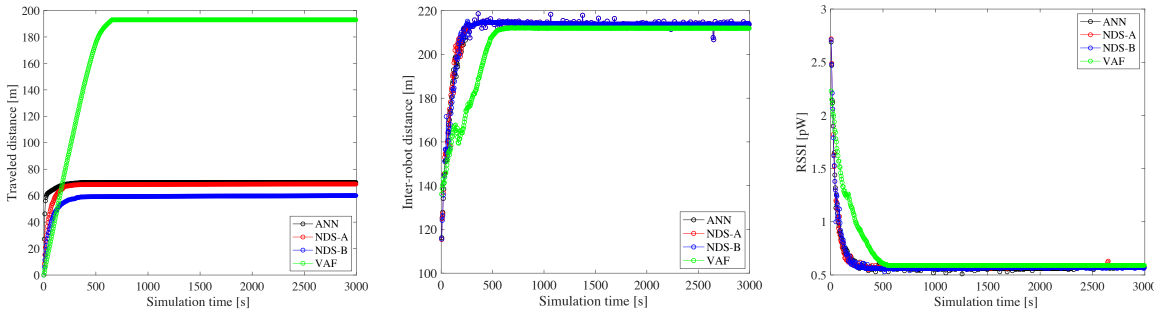
increasing robot density in these scenarios, the traveled distance is higher than the value obtained in case of 10 robots only (*i.e.*, $\approx 200$ m).

In case of increasing network size (see Figure 7 (a)), we observe the loss of the linearity property noticed when the robot number is less than 50 (*i.e.*, the average traveled distance increases with the robot number). However, NDS-B algorithm remains the most effective in terms of traveled distance, which validates our theoretical model described in Section 3. In contrast, VAF shows a very low convergence time, as well as a slow decrease toward the desired RSSI level, as depicted in Figure 7 (b) and Figure 7 (c), respectively. Finally, by considering the relationship between energy and traveled distance, we can say that our NDS approaches are energy efficient since they decrease the average distance traveled by a robot (hence, the energy consumed by a robot decreases as well).

### *4.2. Convergence time*

Figure 5 (b) illustrates the convergence of the algorithms to the desired distance set to 212 m. We can see that the convergence time of our NDS approaches is more or less equal to the convergence time of ANN, while VAF presents a slow slope towards the desired distance.

Neighbor discovery is a key component in our approaches since each robot uses the knowledge of the neighborhood to compute its new position. Therefore, a poor knowledge of the neighborhood can affect the effectiveness of our algorithms; this can happen in low density scenarios, due to lower number of robots in the network. In case of the scenario with 50 robots, we observe a slow increase of VAF (*i.e.*, it needs more time to converge to the desired distance $D_{Th}$). As stated before, this




(a) (b) (c)

Figure 5: Simulation results in terms of (a) traveled distance, (b) inter-robot distance, and (c) RSSI, obtained with 10 robots.

is due to the fact that VAF does not have only to regulate the distance to the desired distance but it has to regulate the angle between a robot and its two consecutive neighbors. It should be noted that the convergence is acquired if and only if, for any pair of robots $(u, v)$ in the network we have $d(u, v) = D_{th} \pm \epsilon$, where $\epsilon$ is the tolerance value.

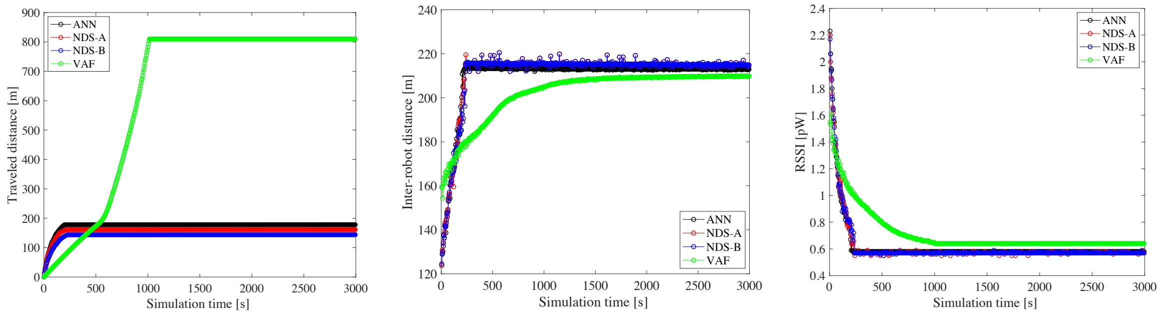
### *4.3. QoS assessment*

Finally, Figure 5 (c) shows that our proposed approaches can maintain the desired QoS level (expressed as RSSI of $\approx 0.5$ pW) among neighboring robots. In this paper, the choice of the desired QoS level was made on the basis of the desired distance $D_{th}$ *i.e.*, the RSSI measured at a distance $D_{th}$) but can be used of course independently. For example, we can setup the desired QoS level based on the results in [57], which show that the packet received rate is at least 85% for links with RSSI above 3.16 pw (*i.e.*, $-87$ dBm), using CC2420 single-chip RF transceiver that is compliant with IEEE 802.15.4 standard. In Figure 5 (c) we observe that VAF reaches the desired RSSI slowly, while in other scenarios with higher robot densities VAF approaches higher RSSI values (see Figure 6 (c), Figure 7 (c), and Figure 8 (c)).

To summarize, we can conclude that our approaches outperform VAF for what concerns the traveled distance, the convergence time and the desired power level. This occurs both in low and high density scenarios. NDS-A and NDS-B present better performances also with respect to ANN. We observe that for increasing network size NDS-A and NDS-B have a faster convergence time, as well as a fast slope toward the desired RSSI level. Finally, the distance traveled by a single robot is reduced with the proposed techniques, as compared to both ANN and VAF.

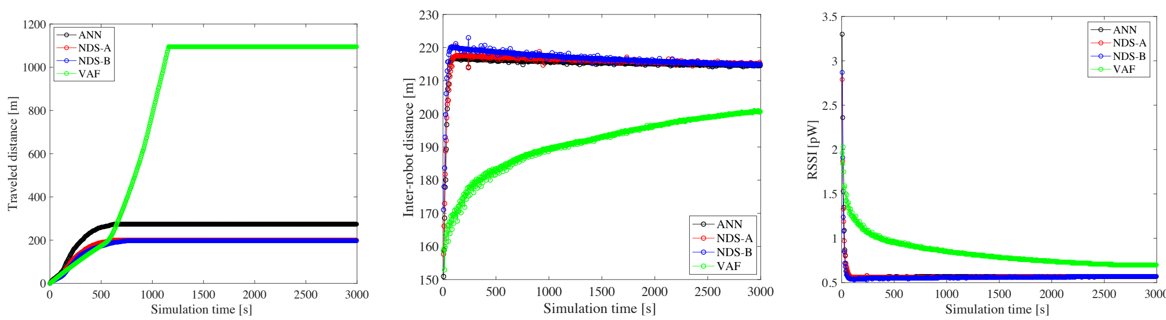
## 5. Conclusions

In this paper, we introduced the NDS algorithm with the aim of efficiently maintain the global connectivity among multiple mobile robots to a desired distance and quality-of- service level. The proposed approach is based on neural network and dominating set strategies. Our contribution was to demonstrate that heterogeneous behavior of robot in MMR networks and IoRT concept can lead to decrease the global traveled distance while achieving an minimum convergence time. Through




(a) (b) (c)

Figure 6: Simulation results in terms of (a) traveled distance, (b) inter-robot distance, and (c) RSSI, obtained with 35 robots.




(a) (b) (c)

Figure 7: Simulation results in terms of (a) traveled distance, (b) inter-robot distance, and (c) RSSI, obtained with 50 robots.

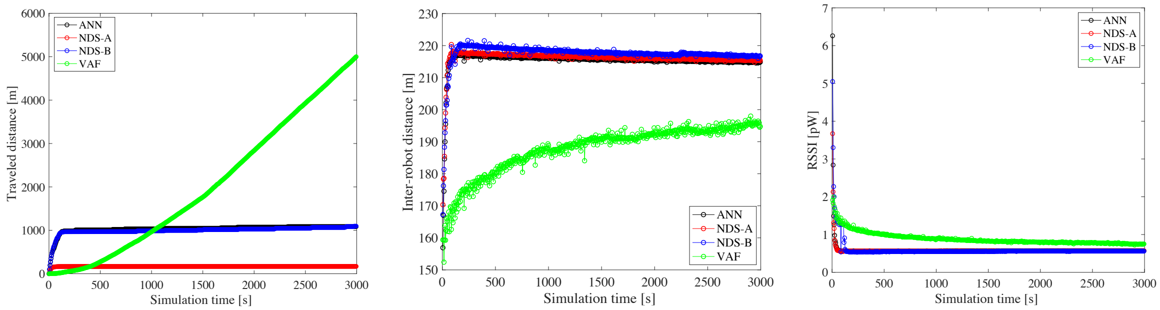
a theoretical analysis and extensive simulations, we showed that our approach outperforms the ANN-Based and VAF approaches proposed in [16] and [17] in terms of traveled distance, while maintaining similar convergence time.

Due to the nature of the problem on the real world *i.e.*, environmental changes, presence of interference, etc., our algorithm must be further evaluated by real experiments. Our future work will focus on this direction; to create a real MMRS scenario by using real mobile robots.

## References


[1] Council, Internet of things council, `http://www.theinternetofthings.eu`, accessed November 18, 2015.

[2] A. Rachedi, M. H. Rehmani, S. Cherkaoui, J. J. P. C. Rodrigues, Ieee access special section editorial: The plethora of research in internet of things (iot), IEEE Access 4 (2016) 9575–9579. doi:10.1109/ACCESS.2016.2647499.




(a) (b) (c)

Figure 8: Simulation results in terms of (a) traveled distance, (b) inter-robot distance, and (c) RSSI, obtained with 70 robots.


[3] A. A. Khan, M. H. Rehmani, A. Rachedi, Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions, IEEE Wireless Communications 24 (3) (2017) 17–25. doi:10.1109/MWC.2017.1600404.

[4] D. Bendouda, A. Rachedi, H. Haffaf, Programmable architecture based on software defined network for internet of things: Connected dominated sets approach, Future Generation Computer Systems 80 (2018) 188 – 197. doi:https://doi.org/10.1016/j.future.2017.09.070.
URL http://www.sciencedirect.com/science/article/pii/S0167739X17314590

[5] G. Aloi, L. Bedogni, M. Di Felice, V. Loscrí, A. Molinaro, E. Natalizio, P. Pace, G. Ruggeri, A. Trotta, N. R. Zema, Stemnet: an evolutionary network architecture for smart and sustainable cities, Transactions on Emerging Telecommunications Technologies 25 (1) (2012) 21–40.

[6] R. Petrolo, V. Loscrí, N. Mitton, Towards a smart city based on cloud of thoings, in: Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities, WiMobCity, ACM, 2014, pp. 61–66.

[7] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, J. Lloret, Context-aware cloud robotics for material handling in cognitive industrial internet of things, IEEE Internet of Things Journal (2017) 1–1doi:10.1109/JIOT.2017.2728722.

[8] O. Qinghai, Z. Yan, L. Xianghen, Z. Yiying, Z. Lingkang, Application of internet of things in smart grid power transmission, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, Vol. 2, IEEE, 2003, pp. 1293–1303.

[9] W. Garage, Pr2 robot, http://www.willowgarage.com/pages/pr2/overview, accessed March 15, 2017.

[10] N. Robotics, Wifibots, http://www.wifibot.com/, accessed March 15, 2017.

[11] A. Research, Internet of robotic things, https://www.abiresearch.com, accessed November 2, 2015.

[12] P. P. Ray, Internet of robotic things: Concept, technologies, and challenges, IEEE Access 4 (2016) 9489–9500. doi:10.1109/ACCESS.2017.2647747.

[13] T. Arai, E. Pagello, L. E. Parker, Editorial: Advances in multi-robot systems, IEEE Transactions on robotics and automation 18 (5) (2002) 655–661.

[14] Y. U. Cao, A. S. Fukunaga, A. Kahng, Cooperative mobile robotics: Antecedents and directions, Autonomous robots 4 (1) (1997) 7–27.

[15] L. E. Parker, Multiple mobile robot systems, in: Springer Handbook of Robotics, Springer, 2008, pp. 921–941.

[16] C. Razafimandimby, V. Loscrí, A. M. Vegni, A neural network and IoT-based scheme for performance assessment in Internet of Robotic Things, in: I4T - 1st International Workshop on Interoperability, Integration, and Interconnection of Internet of Things Systems, Berlin, Germany, 2016.
URL `https://hal.inria.fr/hal-01261842`

[17] A. Casteigts, J. Albert, S. Chaumette, A. Nayak, I. Stojmenovic, Biconnecting a network of mobile robots using virtual angular forces, Computer Communications 35 (9) (2012) 1038–1046.

[18] R. Beckers, O. Holland, J.-L. Deneubourg, From local actions to global tasks: Stigmergy and collective robotics, in: Artificial life IV, Vol. 181, 1994, p. 189.

[19] A. Campbell, C. Riggs, A. S. Wu, On the impact of variation on self-organizing systems, in: Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on, IEEE, 2011, pp. 119–128.

[20] F. Ducatelle, G. A. Di Caro, C. Pinciroli, L. M. Gambardella, Self-organized cooperation between robotic swarms, Swarm Intelligence 5 (2) (2011) 73–96.

[21] A. Prasath, A. Venuturumilli, A. Ranganathan, A. A. Minai, Self-organization of sensor networks with heterogeneous connectivity, in: Sensor Networks, Springer, 2010, pp. 39–59.

[22] A. Prorok, M. A. Hsieh, V. Kumar, Fast redistribution of a swarm of heterogeneous robots, in: International Conference on Bio-inspired Information and Communications Technologies, 2015.

[23] M. C. D. Gennaro, A. Jadbabaie, Decentralized control of connectivity for multi-agent systems, in: Decision and Control, 2006 45th IEEE Conference on, Citeseer, 2006, pp. 3628–3633.

[24] L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, Distributed control of multirobot systems with global connectivity maintenance, Robotics, IEEE Transactions on 29 (5) (2013) 1326–1332.

[25] N. Bouraqadi, S. Stinckwich, V. Moraru, A. Doniec, et al., Making networked robots connectivity-aware, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 3502–3507.

[26] M. Ahmadi, P. Stone, Keeping in touch: Maintaining biconnected structure by homogeneous robots, in: Proceedings of the National Conference on Artificial Intelligence, Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 580.

[27] M. M. Zavlanos, G. J. Pappas, Distributed connectivity control of mobile networks, Robotics, IEEE Transactions on 24 (6) (2008) 1416–1428.

[28] E. Stump, A. Jadbabaie, V. Kumar, Connectivity management in mobile robot teams, in: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 1525–1530.

[29] M. A. Hsieh, A. Cowley, V. Kumar, C. J. Taylor, Maintaining network connectivity and performance in robot teams: Research articles, J. Field Robot. 25 (1-2) (2008) 111–131. doi:10.1002/rob.v25:1/2.
URL `http://dx.doi.org/10.1002/rob.v25:1/2`

[30] M. Schuresko, J. Cortés, Distributed motion constraints for algebraic connectivity of robotic networks, Journal of Intelligent and Robotic Systems 56 (1-2) (2009) 99–126.

[31] H. Zhang, J. C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, Ad Hoc & Sensor Wireless Networks 1 (1-2) (2005) 89–124.

[32] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, 2003, pp. 28–39.

[33] K. Kar, S. Banerjee, Node placement for connected coverage in sensor networks, in: WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003, pp. 2–pages.

[34] F. Guerriero, V. Loscrí, P. Pace, R. Surace, Neural networks and sdr modulation schemes for wireless mobile nodes: A synergic approach, Ad Hoc Networks 54 (2017) 17 – 29. doi:https://doi.org/10.1016/j.adhoc.2016.09.016.
URL `http://www.sciencedirect.com/science/article/pii/S1570870516302311`

[35] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, Vol. 2, IEEE, 2003, pp. 1293–1303.

[36] J. Chen, S. Li, Y. Sun, Novel deployment schemes for mobile sensor networks, Sensors 7 (11) (2007) 2907–2919.

[37] G. Tan, S. A. Jarvis, A.-M. Kermarrec, Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks, IEEE Transactions on Mobile Computing 8 (6) (2009) 836–848.

[38] G. Wang, G. Cao, T. F. La Porta, Movement-assisted sensor deployment, IEEE Transactions on Mobile Computing 5 (6) (2006) 640–652.

[39] A. Breitenmoser, A. Martinoli, On combining multi-robot coverage and reciprocal collision avoidance, in: Distributed Autonomous Robotic Systems, Springer, 2016, pp. 49–64.

[40] M. Abo-Zahhad, N. Sabor, S. Sasaki, S. M. Ahmed, A centralized immune-voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks, Information Fusion 30 (2016) 36–51.

[41] Y.-h. Kim, C.-M. Kim, D.-S. Yang, Y.-j. Oh, Y.-H. Han, Regular sensor deployment patterns for $p$-coverage and $q$-connectivity in wireless sensor networks, in: Information Networking (ICOIN), 2012 International Conference on, IEEE, 2012, pp. 290–295.

[42] Z. Yun, X. Bai, D. Xuan, T. H. Lai, W. Jia, Optimal deployment patterns for full coverage and $k$-connectivity ($k \leq 6$) wireless sensor networks, IEEE/ACM Transactions on Networking (TON) 18 (3) (2010) 934–947.

[43] H. Chen, H. Wu, N.-F. Tzeng, Grid-based approach for working node selection in wireless sensor networks, in: Communications, 2004 IEEE International Conference on, Vol. 6, IEEE, 2004, pp. 3673–3678.

[44] R. Kazala, A. Taneva, M. Petrov, S. Penkov, Wireless network for mobile robot applications, IFAC-PapersOnLine 48 (24) (2015) 231 – 236, 16th IFAC Conference on Technology, Culture and International Stability TECIS 2015. doi:https://doi.org/10.1016/j.ifacol.2015.12.088.
URL `http://www.sciencedirect.com/science/article/pii/S2405896315027123`

[45] G. Mahadevan, On domination theory and related concepts in graphs, ph.D. thesis, Manonmaniam Sundaranar University, India (2005).

[46] C. Lenzen, R. Wattenhofer, Minimum dominating set approximation in graphs of bounded arboricity, in: Proceedings of the 24th International Conference on Distributed Computing, DISC'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 510–524.
URL `http://dl.acm.org/citation.cfm?id=1888781.1888844`

[47] F. Kuhn, R. Wattenhofer, Constant-time distributed dominating set approximation, Distributed Computing 17 (4) (2005) 303–310. doi:10.1007/s00446-004-0112-5.
URL `https://doi.org/10.1007/s00446-004-0112-5`

[48] A. Abraham, Artificial neural networks, handbook of measuring system design.

[49] M. Riedmiller, Advanced supervised learning in multi-layer perceptronsfrom backpropagation to adaptive learning algorithms, Computer Standards & Interfaces 16 (3) (1994) 265–278.

[50] L. Jia, R. Rajaraman, T. Suel, An efficient distributed algorithm for constructing small dominating sets, Distributed Computing 15 (4) (2002) 193–205. doi:10.1007/s00446-002-0078-0.
URL `https://doi.org/10.1007/s00446-002-0078-0`

[51] M. Fiore, Ns-2.29 wireless update patch, `http://perso.citi.insa-lyon.fr/mfiore/research.html`, accessed November 19, 2015.

[52] Q. Sun, Y. Tian, M. Diao, Cooperative localization algorithm based on hybrid topology architecture for multiple mobile robot system, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2812179.

[53] Y. Noh, H. Yamaguchi, U. Lee, Infrastructure-free collaborative indoor positioning scheme for time-critical team operations, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (3) (2018) 418–432. doi:10.1109/TSMC.2016.2615652.

[54] J. W. Qiu, C. P. Lin, Y. C. Tseng, Ble-based collaborative indoor localization with adaptive multi-lateration and mobile encountering, in: 2016 IEEE Wireless Communications and Networking Conference, 2016, pp. 1–7. doi:10.1109/WCNC.2016.7564799.

[55] J. W. Qiu, Y. C. Tseng, M2m encountering: Collaborative localization via instant inter-particle filter data fusion, IEEE Sensors Journal 16 (14) (2016) 5715–5724. doi:10.1109/JSEN.2016.2566679.

[56] J. D. P. Pavon, S. Choi, Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement, in: Communications, 2003. ICC'03. IEEE International Conference on, Vol. 2, IEEE, 2003, pp. 1108–1113.

[57] K. Srinivasan, P. Levis, RSSI Is Under-Appreciated, in: Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), 2006.