JAIST Repository
https://dspace.jaist.ac.jp/

Title	A geometric approach to deploying robot swarms
Author(s)	Lee, Geunho; Chong, Nak Young
Citation	Annals of Mathematics and Artificial Intelligence, 52(2-4): 257-280
Issue Date	2009-03-07
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/9512
Rights	This is the author-created version of Springer, Geunho Lee and Nak Young Chong, Annals of Mathematics and Artificial Intelligence, 52(2-4), 2009, 257-280. The original publication is available at www.springerlink.com, http://dx.doi.org/10.1007/s10472-009-9125-x
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Noname manuscript No.
(will be inserted by the editor)

A Geometric Approach to Deploying Robot Swarms

Geunho Lee · Nak Young Chong

Received: date / Accepted: date

Abstract We discuss the fundamental problems and practical issues underlying the deployment of a swarm of autonomous mobile robots that can potentially be used to build mobile robotic sensor networks. For the purpose, a geometric approach is proposed that allows robots to configure themselves into a two-dimensional plane with uniform spatial density. Particular emphasis is paid to the hole repair capability for dynamic network reconfiguration. Specifically, each robot interacts selectively with two neighboring robots so that three robots can converge onto each vertex of the equilateral triangle configuration. Based on the local interaction, the self-configuration algorithm is presented to enable a swarm of robots to form a communication network arranged in equilateral triangular lattices by shuffling the neighbors. Convergence of the algorithms is mathematically proved using Lyapunov theory. Moreover, it is verified that the self-reparation algorithm enables robot swarms to reconfigure themselves when holes exist in the network or new robots are added to the network. Through extensive simulations, we validate the feasibility of applying the proposed algorithms to self-configuring a network of mobile robotic sensors. We describe in detail the features of the algorithm, including self-organization, self-stabilization, and robustness, with the results of the simulation.

Keywords robot swarms · local interactions · equilateral triangular lattice · self-configuration · self-reparation

1 Introduction

With the advance of mobile networking technology, much attention has been paid to the use of a swarm of simple robots having mobility and wireless communication

Geunho Lee and Nak Young Chong
School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, JAPAN
Tel.: +81-761-51-1248
Fax: +81-761-51-1149
E-mail: {geun-lee, nakyoung}@jaist.ac.jp

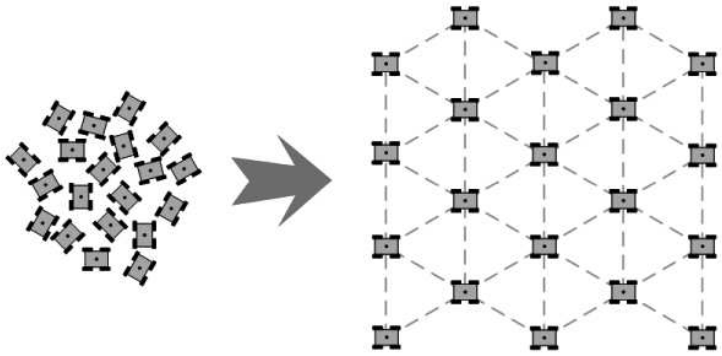


Fig. 1 Deploying networked robots with uniform spatial density

capabilities [1][2]. Mobile robotic sensors can be expected to be used in real applications such as environmental or habitat monitoring, exploration, search-and-rescue, and so on. These applications are performed within an area of interest that should be properly covered by robots [3][4]. It is therefore necessary to coordinate the positions of robots, since their sensing and communication ranges are usually limited to a small area.

Recently, a decentralized control for robot deployment has been reported in the field of swarm robotics. These works can be broadly classified into global and local strategies according to whether global state information is available to all robots. Global strategies [5]-[9] may provide accurate deployment, but lack scalability. On the other hand, local strategies are based on interactions between individual robots observed from social organisms such as colonies of ants or schools of fish, or physical phenomena such as crystallization. Local strategies can be further divided into biological emergence [10]-[14], behavior-based [15][16], and virtual physics-based [17]-[28] approaches. Many of the behavior-based and virtual physics-based approaches used such physical phenomena as electric charges [17], gravitational forces [18][19], springs [20]-[22], potential field [23][24], and other virtual force models [25]-[28]. Those works mostly use a force balance between inter-individual interactions exerting an attractive or repulsive force onto all the robots within a certain range, which might over-constrain individual robots and frequently lead to deadlocks.

To overcome the above-mentioned problems, we propose a self-configuration control that constructs a communication network composed of equilateral triangle lattices as shown in Fig 1. Based on partially connected mesh topology [29][30], the proposed method can take advantage of the redundancy provided by a fully connected network topology without the expense and complexity of networking processes. In [21][22], partial graph pairs of robots were proposed that exert virtual spring forces onto each other when their connection is part of the graph. Our approach can reduce the number of deployed robots in a given location compared with their approaches, and configure more energy-efficient routing protocols than those of randomly deployed robots [31].

The main contribution of this work lies in establishing the selective local interaction among neighboring robots that forms a network of equilateral triangles over a two-dimensional plane. It moreover provides the hole repair capability and improves the network connectivity by increasing the number of robots positioned at the uniform distance. Regarding the convergence of the proposed algorithms, energy-like scalar

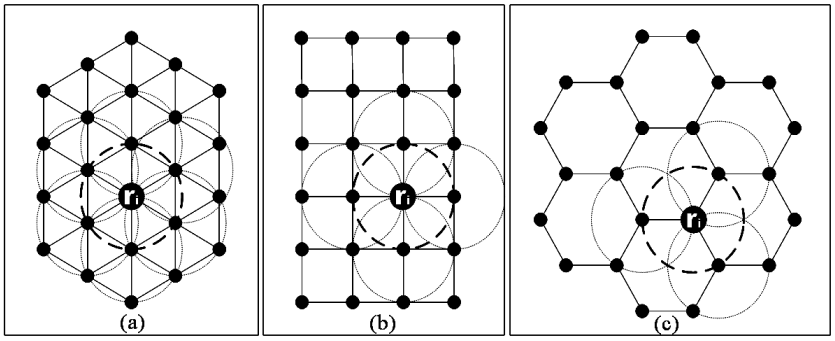


Fig. 2 Three network topologies: (a) triangle, (b) square, (c) hexagon

functions based on Lyapunov theory are utilized [32]-[35], leading to asymptotic stability of the desired configuration from an arbitrary distribution. Both individual behavior of robots and overall shape of the swarm can be coordinated with scalability.

The rest of this paper is organized as follows. Section 2 presents the assumptions about the robot and the definitions of the self-configuration problem. Section 3 describes the fundamental motion planning concept through the selective local interaction with neighboring robots and the convergence property of the proposed interaction model. Section 4 presents the convergence of the algorithm for a swarm of n (that is greater than 3) robots utilizing the collective scalar function which is a diminished energy function with a scalar potential. Section 5 provides the results of simulations and discussion. Section 6 draws conclusions.

2 Background and Problem Statement

2.1 Network Topology

We discuss three basic networking patterns of different geometries. By comparing these patterns, we explain the features of the equilateral triangular lattice characterized by our proposed model. Each of the patterns consists of 24 robots with communication capabilities as shown in Fig. 2. It is assumed that robots are deployed with the same uniform density d_u so that direct communication is available between neighboring robots. Each robot r_i has six, four, and three neighboring robots uniformly located around itself, yielding the geometry of triangle, square, and hexagon, respectively. Given the same number of robots, we compare the coverage area, coverage density [39], and k connectivity [40][41] of the network. The coverage area of each triangular lattice is $\frac{\sqrt{3}}{4}{d_u}^2$. n robots can cover approximately an area of less than $\frac{\sqrt{3}}{2}n{d_u}^2$. Similarly, the same number of robots arranged in square lattices and hexagonal lattices can cover approximately an area of $n{d_u}^2$ and $\frac{3\sqrt{3}}{4}n{d_u}^2$, respectively. Secondly, the coverage density can accordingly be obtained by dividing the number of robots by the coverage area. Each density of triangle, square, and hexagon are approximated to be $\frac{1.2}{{d_u}^2}$, $\frac{1}{{d_u}^2}$, and $\frac{0.77}{{d_u}^2}$, respectively. Finally, k connectivity represented by the number of neighbor robots indicate six in the triangular lattice, four in the square lattice, and three in the hexagonal lattice. It is evident that the hexagonal lattice provides the best spatial

coverage with the same number of robots. In contrast, the triangular lattice provides enhanced coverage density and k connectivity while trading off the coverage range. Thus, it is our goal to construct an equilateral triangular lattice toward an efficient and robust network covering an assigned area.

2.2 Robotic Model

We consider a swarm of autonomous mobile robots $r_1, \cdots, r_n$. It is assumed that an initial distribution of all robots is arbitrary and distinct. Each robot is modeled as a point with orientation, that freely moves on a two-dimensional plane with limited ranges of sensing. They have no identifiers and do not share any common coordinate system. They do not retain any memory of past states or actions, which gives inherently self-stabilizing property[1] [5], provided that no two robots are exactly identical (*i.e.*, having both the same initial position and the same local coordinates). They can detect the position of other robots in close proximity, but are not allowed to communicate explicitly with them. Each of the robots executes the same algorithm, but acts independently and asynchronously from other robots.

2.3 Notations and Formal Definitions

This part gives notations and formal definitions used throughout the paper. The distance between the robot r_i's position p_i and the robot r_j's position p_j is denoted as $dist(p_i, p_j)$. Denote a constant distance as d_u that is finite and greater than zero. Each robot has a limited sensing boundary SB. Then r_i detects the position of other robots $\{p_1, p_2, \cdots\}$ located within its SB, and makes the set of the observed positions O_i with respect to its local coordinate system. Now r_i can select two robots $s1$ and $s2$ from O_i and denote their positions as p_{s1} and p_{s2}, respectively. We call $s1$ and $s2$ the neighbors of r_i and denote the set of their positions $\{p_{s1}, p_{s2}\}$ as N_i.

Definition 1. (Triangular Configuration) *Given p_i and N_i, a triangular configuration is defined as a set of three distinct positions $\{p_i, p_{s1}, p_{s2}\}$ denoted by*

$$\mathbb{T}_i = \{p_i, p_{s1}, p_{s2}\},$$

where we define the internal angle of r_i as α_i, which is $\angle p_{s1}p_ip_{s2}$ (or $\angle p_{s2}p_ip_{s1}$).

When robots form $\mathbb{T}_i$, the following definition can be addressed:

Definition 2. (Equilateral Configuration) *Among $\mathbb{T}_i$, we define an equilateral configuration $\mathbb{E}_i$, provided that all the possible distance permutations $dist(p_{\pi(i)}, p_{\pi(j)})$ are equal to d_u. (It can also be expressed in the following way. Given $\mathbb{T}_i$ whose internal angles are all the same, we call such $\mathbb{T}_i$ an equilateral configuration.)*

Now we need a measure indicating to which degree $\mathbb{T}_i$ is configured into $\mathbb{E}_i$ as follows:

[1] Self-stabilizing system, when started from an arbitrary initial state, always converges toward a desired behavior [37][38].

Definition 3. (Distance Matrix) *Given $\mathbb{T}_i$, we can express all the possible distance permutations among robots as the following matrix termed the distance matrix with respect to r_i.*

$$\mathbf{D}_i = \begin{cases} \Big(dist(p_m, p_n) - d_u\Big)^2 & \text{if } m \neq n \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\{\{p_m, p_n\} |\ p_m, p_n \in \mathbb{T}_i = \{p_i, p_{s1}, p_{s2}\}\}$.

We will denote $(dist(p_m, p_n) - d_u)^2)$ for simplicity as $(d_k - d_u)^2$.

Using $\mathbb{T}_i$ and $\mathbb{E}_i$, we can define the local interaction as follows:

Problem 4. (Local Interaction) *Given $\mathbb{T}_i$, the local interaction is to have r_i maintain d_u with N_i at each instant in time toward forming $\mathbb{E}_i$.*

Based on the local interaction that allows three robots to converge into $\mathbb{E}_i$, Self-Configuration Problem by a swarm of mobile robots can be stated as follows:

Problem 5. (Self-configuration) *Given a swarm of robots $r_1, \cdots, r_n$ with arbitrarily distinct positions in a two-dimensional plane, all robots can configure themselves into $\mathbb{E}_i$ without irregularities such as holes through the local interaction after a finite number of activation steps.*

Each robot can either be idle or execute the interaction, repeating recursive activation at each cycle. They compute their movement position (computation), based on the positions of other robots (observation), and move toward the computed position (motion). Here, the motion of each robot is controlled by the first-order linear differential equation for the distance from the centroid of $\mathbb{T}_i$, and for the internal angle of $\mathbb{T}_i$, respectively (see Section 3.3).

3 Local Interactions

This section describes the local interaction algorithm that enables three neighboring robots to generate $\mathbb{E}_i$ from an arbitrary $\mathbb{T}_i$ (see Algorithm-1), and proves the convergence of the algorithm.

3.1 Algorithm Description

The algorithm consists of a function $\varphi_{interaction}$ whose arguments are p_i and N_i at each activation. Consider the robot r_i and its two neighbors $s1$ and $s2$ located within r_i's *SB*. As shown in Fig. 3-(a), three robots are configured into $\mathbb{T}_i$ whose vertices are p_i, p_{s1}, and p_{s2}, respectively. As illustrated in Fig. 3-(b), r_i finds the centroid p_{ct} of the triangle $\triangle p_i p_{s1} p_{s2}$ with respect to its local coordinates, and measures the angle ϕ between the line connecting two neighbors and r_i's horizontal axis. Using p_{ct} and ϕ, r_i

Algorithm-1 Local Interaction
constant d_u := uniform distance
FUNCTION $\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$
1 $(p_{ct,x}, p_{ct,y}) := centroid(p_{s1}, p_{s2}, p_i)$
2 ϕ := angle between $\overline{p_{s1}p_{s2}}$ and r_i's local horizontal axis
3 $p_{ti,x} := p_{ct,x} + d_u \cos(\phi + \pi/2)/\sqrt{3}$
4 $p_{ti,y} := p_{ct,y} + d_u \sin(\phi + \pi/2)/\sqrt{3}$
5 $p_{ti} := (p_{ti,x}, p_{ti,y})$

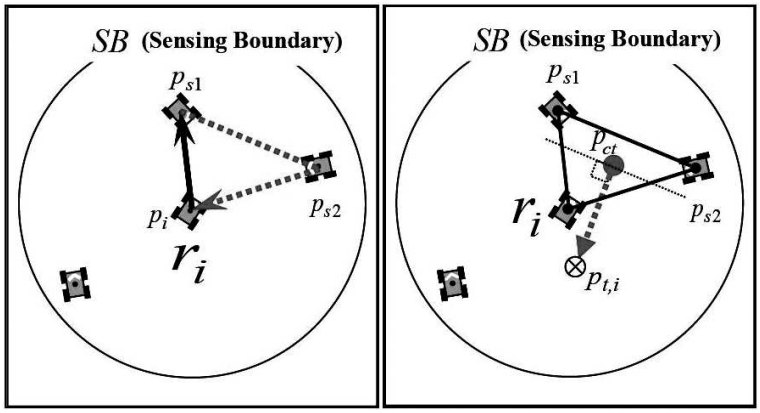


Fig. 3 Illustration of Algorithm-1 (a) triangular configuration ($\mathbb{T}_i$), (b) computation of the target point

calculates the target point p_{ti} as described in Line 3 and Line 4 of Algorithm-1. Each robot computes the target point at each instant in time by their current observation of neighboring robots.

To assist the explicit understanding of the proposed local interaction, we explain how the positions of three robots converge into $\mathbb{E}_i$. Consider a triangle with three vertices p_a, p_b, and p_c that represent the position of three robots A, B, and C as shown in Fig. 4. Let α, β, and γ denote the internal angles of the triangle, respectively. Each robot located at the vertex of $\triangle p_a p_b p_c$ moves to the new target position p_{ta}, p_{tb}, and p_{tc} computed by Algorithm-1. The internal angles of $\triangle p_{ta} p_{tb} p_{tc}$ are α', β', and γ', respectively. Let p_{ct} denote the centroid of $\triangle p_a p_b p_c$. Also, let p denote the point projected from p_{ct} onto $\overline{p_a p_b}$. Similarly, let q indicate the point projected from p_{ct} onto $\overline{p_a p_c}$. If we consider a quadrangle $\square p_a p p_{ct} q$, the angles of p and q are right angles. Therefore, $\angle p p_{ct} q$ becomes $180^\circ - \alpha$. From Fig. 4, $\angle p_{tb} p_{ct} p_{tc}$ is equal to $\angle p p_{ct} q$. $\triangle p_{tb} p_{ct} p_{tc}$ is an isosceles triangle since $\overline{p_{ct} p_{tb}}$ and $\overline{p_{ct} p_{tc}}$ is identical ($d_u/\sqrt{3} = \sqrt{3}/2 \times d_u \times 2/3$). Hence, α of $\triangle p_a p_b p_c$ is equal to $2a$ in the figure. With the same manner, β and γ become $2b$ and $2c$, respectively. Moreover, we see that α' of $\triangle p_{ta} p_{tb} p_{tc}$ is $(\beta+\gamma)/2$ (or equal to $(b+c)$). Likewise, β' indicates $(\alpha+\gamma)/2$ and γ' does $(\alpha+\beta)/2$. Accordingly, α' is given by $(\beta+\gamma)/2$. Now the relation between internal angles can be rewritten as a function of time to give the following equation:

$$\begin{aligned} \alpha(t+1) &= (\beta(t) + \gamma(t))/2 \\ \beta(t+1) &= (\gamma(t) + \alpha(t))/2 \\ \gamma(t+1) &= (\alpha(t) + \beta(t))/2 \end{aligned} \quad . \tag{2}$$

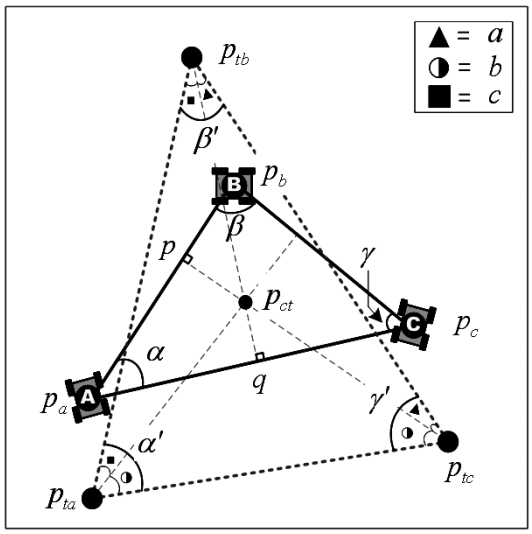


Fig. 4 Robots attempt to form an isosceles triangle at each time

where t and $t+1$ represent the current time instant and the next time instant.

Similarly, the following equations hold:

$$\begin{array}{l} \alpha(t+2) = \frac{\beta(t+1)+\gamma(t+1)}{2} = \frac{\alpha(t)}{2} + \frac{\beta(t)+\gamma(t)}{4} \\ \beta(t+2) = \frac{\gamma(t+1)+\alpha(t+1)}{2} = \frac{\beta(t)}{2} + \frac{\gamma(t)+\alpha(t)}{4} \\ \gamma(t+2) = \frac{\alpha(t+1)+\beta(t+1)}{2} = \frac{\gamma(t)}{2} + \frac{\alpha(t)+\beta(t)}{4} \end{array} . \tag{3}$$

Now, using (2), (3) can be rewritten as follows:

$$\begin{array}{l} \alpha(t+2) = \alpha(t)/2 + \alpha(t+1)/2 \\ \beta(t+2) = \beta(t)/2 + \beta(t+1)/2 \\ \gamma(t+2) = \gamma(t)/2 + \gamma(t+1)/2 \end{array} . \tag{4}$$

It is straightforward to transform (2) into the following matrix:

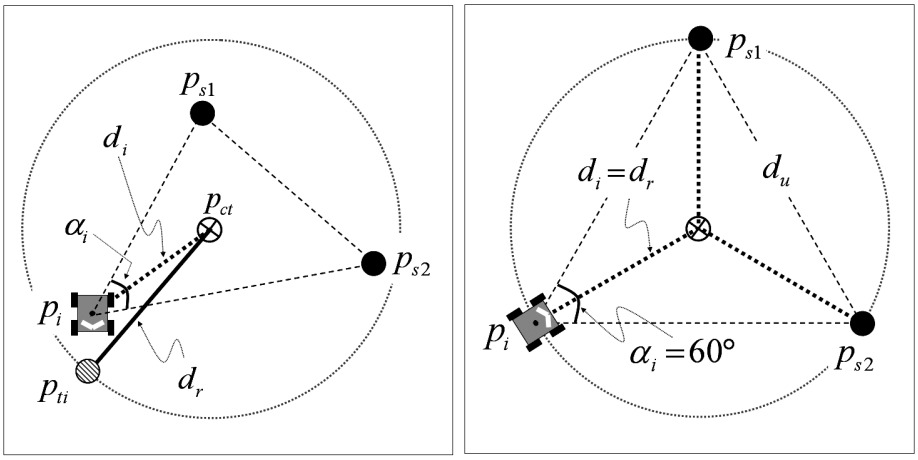
$$\begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \\ \gamma(t+1) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} . \tag{5}$$

(4) can also be rewritten as follows:

$$\begin{bmatrix} \alpha(t+2) \\ \beta(t+2) \\ \gamma(t+2) \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \\ \gamma(t+1) \end{bmatrix} \right) . \tag{6}$$

Substituting (5) into (6) gives

$$\begin{aligned} \begin{bmatrix} \alpha(t+2) \\ \beta(t+2) \\ \gamma(t+2) \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \frac{1}{2^2} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} \\ &= \frac{1}{2^2} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \frac{3}{2^2} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{\alpha(t)+\beta(t)+\gamma(t)}{3} . \end{aligned} \tag{7}$$



(a) control parameters: range and bearing (b) equilateral triangular configuration

Fig. 5 Two control parameters in local interaction

The relation for $(t+n)$ becomes the following generalized equation:

$$\begin{bmatrix}\alpha(t+n)\\ \beta(t+n)\\ \gamma(t+n)\end{bmatrix} = \frac{1}{2^n}\begin{bmatrix}\alpha(t)\\ \beta(t)\\ \gamma(t)\end{bmatrix} + \frac{\sum_{k=1}^{n} 3\cdot 2^{k-2}}{2^n}\begin{bmatrix}1\\1\\1\end{bmatrix}\frac{\alpha(t)+\beta(t)+\gamma(t)}{3}. \tag{8}$$

Taking an infinite value, the final value is given by,

$$\lim_{n\to\infty}\begin{bmatrix}\alpha(t+n)\\ \beta(t+n)\\ \gamma(t+n)\end{bmatrix} = 0\times\begin{bmatrix}\alpha(t)\\ \beta(t)\\ \gamma(t)\end{bmatrix} + 1\times\begin{bmatrix}1\\1\\1\end{bmatrix}\frac{\alpha(t)+\beta(t)+\gamma(t)}{3} = \begin{bmatrix}60^\circ\\ 60^\circ\\ 60^\circ\end{bmatrix}. \tag{9}$$

From (9), we see that each internal angle converges into 60° after infinite activation steps. r_i attempts to maintain d_u with its two neighbors, forming an isosceles triangle at each time instant. By repeatedly doing this, three robots will configure themselves into an equilateral triangle with side length d_u.

3.2 Convergence of Local Interactions

Let's consider the circumscribed circle of an equilateral triangle whose center is p_{ct} of $\triangle p_i p_{s1} p_{s2}$ configured from the positions of three robots and radius is $d_u/\sqrt{3}$. The motion planning of the robots is performed by controlling the distance from p_{ct} and the internal angle (See Fig. 5-(a)).

First, the distance is controlled by the following equation

$$\dot{d}_i(t) = -a\Big(d_i(t) - d_r\Big) \tag{10}$$

where a is a positive constant and d_r represents the length $d_u/\sqrt{3}$. Indeed, the solution of (10) is $d_i(t) = |d_i(0)|e^{-at} + d_r$ that converges exponentially to d_r as t approaches infinity.

Secondly, the internal angle is controlled by the following equation

$$\dot{\alpha}_i(t) = k\Big(\beta_i(t) + \gamma_i(t) - 2\alpha_i(t)\Big) \tag{11}$$

where k is a positive number. Because the total internal angle of a triangle is 180°, (11) can be rewritten as

$$\dot{\alpha}_i(t) = k'\Big(60^\circ - \alpha_i(t)\Big), \tag{12}$$

where k' is $3k$. Likewise, the solution of (12) is $\alpha_i(t) = |\alpha_i(0)|e^{-k't} + 60^\circ$ that converges exponentially to 60° as t approaches infinity.

Note that (10) and (12) imply that the trajectory of r_i converges to d_r and 60°, an equilibrium state shown in Fig. 5-(b). This also implies that three robots eventually form an equilateral triangle with d_u. In order to prove the correctness, we will take advantage of stability based on Lyapunov's theory [35]. The stability theorem states if there exists a scalar function $f_{l,i}$ of the state $\mathbf{x} = [d_i(t)\ \alpha_i(t)]^T$ with continuous first order derivatives such that $f_{l,i}$ is positive definite, $\dot{f}_{l,i}$ is negative definite, and $f_{l,i} \to \infty$ as $\| \mathbf{x} \| \to \infty$, then the equilibrium at the desired state $[d_r\ 60^\circ]^T$ is asymptotically stable. Thus, at the desired configuration $\mathbb{E}_i$, the energy level of the scalar function is minimized.

THEOREM 1. *Given that three robots are positioned arbitrarily and distinctively in a two-dimensional plane, under the local interaction algorithm, each robot can converge onto any vertex of $\mathbb{E}_i$ with d_u.*

proof: Consider the following scalar function:

$$f_{l,i} = \frac{1}{2}(d_i - d_r)^2 + \frac{1}{2}(60^\circ - \alpha_i)^2 \tag{13}$$

This scalar function is always positive definite except $d_i \neq d_r$ and $\alpha_i \neq 60$. The derivative of the above function is given by

$$\dot{f}_{l,i} = -(d_i - d_r)^2 - (60^\circ - \alpha_i)^2, \tag{14}$$

which is obtained using (10) and (11). Eq. (14) is negative definite. The scalar function $f_{l,i}$ is radially unbounded since it tends to infinity as $\| \mathbf{x} \| \to \infty$. Therefore, the equilibrium state is asymptotically stable, implying that r_i reaches a vertex of $\mathbb{E}_i$. □

4 Self-configuration

This section describes the self-configuration algorithm, termed Algorithm-2, to deploy a swarm of robots into equilateral triangle lattices based on the local interaction and proves the convergence of the algorithm.

Algorithm-2 Self-configuration
Function $\varphi_{configuration}(O_i, p_i)$
$p_{s1} := \min_{p \in O_i - \{p_i\}} [dist(p_i, p)]$
$p_{s2} := \min_{p \in O_i - \{p_i, p_{s1}\}} [dist(p_{s1}, p) + dist(p, p_i)]$
$\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

4.1 Algorithm Description

A multitude of equilateral triangular lattices is denoted by $\sum_{i=1}^{n} \mathbb{E}_i$. At each time, to form $\mathbb{T}_i$, r_i selects the first neighbor $s1$ positioned at the shortest distance away from itself within O_i. When there exists more than one candidate for $s1$, r_i uniquely determines its $s1$ by sorting the positions of the candidates in increasing order as follows;

$$\forall p_{s1_m}, p_{s1_n}, \exists p_{s1_m} = (x_{s1_m}, y_{s1_m}), p_{s1_n} = (x_{s1_n}, y_{s1_n})$$
$$if(p_{s1_m} < p_{s1_n}) \Longleftrightarrow [(y_{s1_m} < y_{s1_n}) \vee \{(y_{s1_m} = y_{s1_n}) \wedge (x_{s1_m} < x_{s1_n})\}]$$

where p_{s1_m} and p_{s1_n} are the positions of the candidates r_{s1_m} and r_{s1_n} with respect to the r_i's local coordinates. Next, the second neighbor $s2$ is selected such that the total distance from $s1$ (p_{s1}) to r_i (p_i) passing through $s2$ (p_{s2}) is minimized. If there exist multiple candidates for $s2$, r_i determines its $s2$ by applying the same sorting rule stated above. This condition is called the minimum perimeter condition of $\mathbb{T}_i$. Then, r_i forms $\mathbb{T}_i$ with p_{s1} and p_{s2}, and computes the target point p_{ti} toward forming $\mathbb{E}_i$ by $\varphi_{interaction}$ of Algorithm-1. When three neighboring robots are all aligned, the centroid p_{ct} is set to the center point of the line segment between p_{s1} and p_{s2}. If r_i is located on the line segment, p_{ti} is set to the point $\frac{\sqrt{3}d_u}{2}$ away from p_{ct}. Otherwise, p_{ti} is set to the point $\frac{d_u}{\sqrt{3}}$ away from p_{ct}. Now, Algorithm-2 is proposed to enable a swarm of robots to be arranged in $\sum_{i=1}^{n} \mathbb{E}_i$.

4.2 Convergence of Self-Configuration

Let's assume that when r_i arrives at p_{ti}, a new robot D is found in close vicinity as illustrated in Fig. 6. Then, under Algorithm-2, r_i selects D as $s1$, since it is located the shortest distance, and C as $s2$, since the distance from p_{s1} to p_i via p_{s2} can be minimized. Thus, r_i shuffles its neighbors within SB at each time instant, enabling the robots to configure themselves without having adjacent triangles partly overlapping each other.

We now introduce a modified scalar function of (13) in order to consider the effect of changing neighbors.

LEMMA 2. *Under* Algorithm-2*,* $||p_i - p_{s1}||$ *and* $||p_i - p_{s2}||$ *converge into* d_u *for* r_i *after a finite number of activation steps.*

proof: We use Lyapunov's theory with a scalar function given by

$$f_{sc,i} = \sum_{\mathbb{T}_i} (d_k - d_u)^2 + f_{l,i} \tag{15}$$

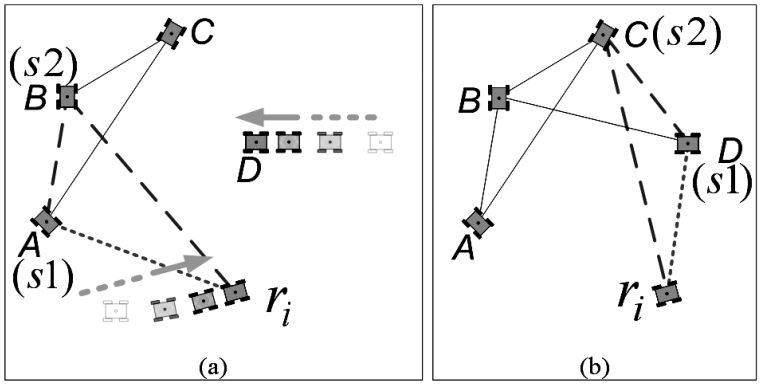


Fig. 6 Dynamically changing neighbors by the minimum perimeter test ((a) r_i moves toward p_{ti}, (b) r_i selects new neighbors according to the minimum perimeter)

where $f_{l,i}$ is given in (13) and $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ is defined as the constant value associated with $\mathbb{T}_i$ at each time (see Definition 3).

A symmetric matrix $\mathbf{D}_i$ can be said to be positive definite, denoted by $\mathbf{D}_i > 0$, if $\mathbf{x}^T\mathbf{D}_i\mathbf{x} > 0$ for every nonzero $\mathbf{x}$ [36]. Thus, from Theorem 1 and $\mathbf{D}_i > 0$, the scalar function (15) is always positive definite except $d_i \neq d_r$ and $\alpha_i \neq 60$. (If $\mathbb{T}_i$ is equal to $\mathbb{E}_i$, it is easily seen that $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ reaches 0, resulted from $d_r = d_u/\sqrt{3}$. Note that even though $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ is 0, $f_{l,i}$ is nonzero by Theorem 1.) The derivative of the scalar function is given by

$$\dot{f}_{sc,i} = \dot{f}_{l,i} = -(d_i - d_r)^2 - (60^\circ - \alpha_i)^2. \tag{16}$$

Eq. (16) is negative definite. Finally, $f_{sc,i}$ is radially unbounded since it tends to infinity as $\| \mathbf{x} \| \rightarrow \infty$. Therefore, the equilibrium state is asymptotically stable, implying that r_i reaches a vertex of $\mathbb{E}_i$ from an arbitrary $\mathbb{T}_i$. □

Next, the collective scalar function $\mathbf{F}_{sc}$ of a swarm of robots is a nonzero function with the property that any solution of the set of algebraic constraints on range and bearing (see Fig. 5-(b)) is closely related to a set of equilibria for $\{r_i | 1 \leq i \leq n\}$ and vice versa. Without loss of generality, $\mathbf{F}_{sc}$ is a diminished energy function with a scalar potential. Now we prove the convergence of the algorithm for a swarm of n robots.

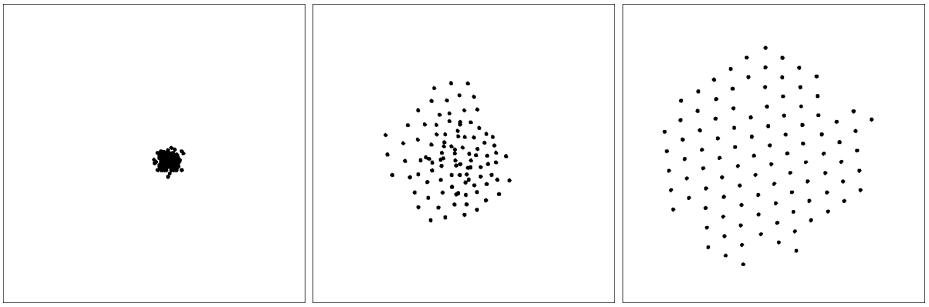
LEMMA 3. *Under* Algorithm-2, $||p_i - p_{s1}||$ *and* $||p_i - p_{s2}||$ *converge into* d_u *for all robots.*

proof: The n-order scalar function $\mathbf{F}_{sc}$ is defined as

$$\mathbf{F}_{sc} = \sum_{i=1}^{n} f_{sc,i} = \sum_{i=1}^{n} \sum_{\mathbb{T}_i} (d_k - d_u)^2 + \sum_{i=1}^{n} f_{l,i}. \tag{17}$$

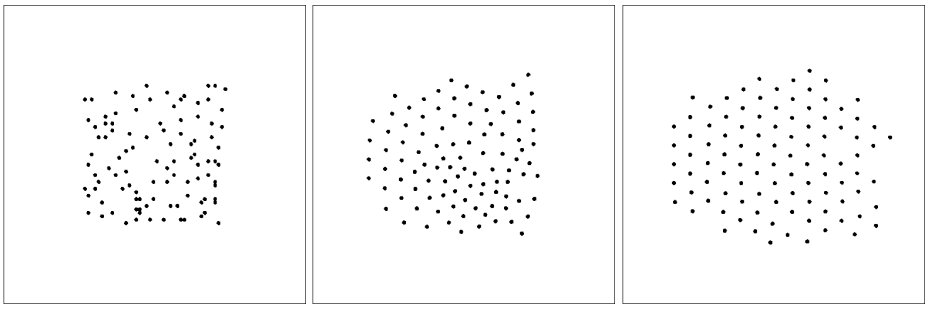
From Lemma 2, it is straightforward to verify that $\mathbf{F}_{sc}$ is positive definite and $\dot{\mathbf{F}}_{sc}$ is negative definite. $\mathbf{F}_{sc}$ is radially unbounded since it tends to infinity as t approaches infinity. Consequently, a swarm of n robots converges into $\sum_{i=1}^{n} \mathbb{E}_i$. □

The proposed self-configuration algorithm terminates when the distance of each robot with their neighbors converges into $d_u \pm 1\%$, which is denoted as $d_{1\%}$. Figs.



(a)initial distribution (b) 2.68 [sec] (c) $d_{1\%}$: 20.66 [sec]

Fig. 7 Self-configuration results from a conglomerate formation with 100 robots



(a) initial distribution (b) 3.37 [sec] (c) $d_{1\%}$: 23.91 [sec]

Fig. 8 Self-configuration results from a random distribution with 100 robots

7 and 8 show that 100 robots configure themselves into a uniform density pattern from different initial conditions. Therefore, collecting the local geometries can globally reach $\sum_{i=1}^{n} \mathbb{E}_i$ without centralized control schemes. From a practical point of view, each robot interacts with only two neighbors, which ensures that the motion of the robot is less affected than other approaches that employ a larger number of neighbors. Accordingly, the computational load decreases.

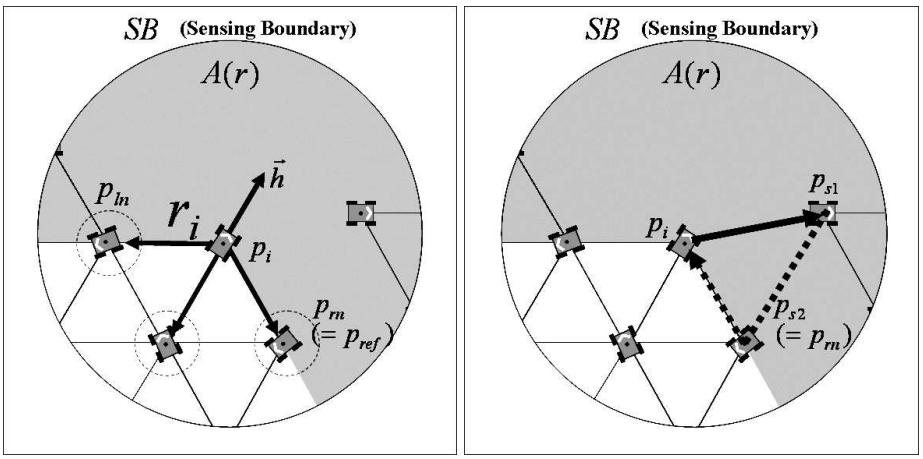
4.3 Extended Algorithm

Now, we introduce Algorithm-3, called the self-reparation algorithm, that enables robots to deploy themselves in a uniform spatial density without such irregularities as holes based on the local interaction algorithm. As observed in Figs. 7-(c) and 8-(c), the self-configuration algorithm may yield holes in a converged distribution. Specifically, Algorithm-2 involves only two neighbors for direct interaction. Each robot determines their direction of movement when selecting the neighbors at each time. Also, from the obliviousness condition, robots neither remember their previous neighbors nor estimate the neighbors' motion. Therefore it is often the case that unintended holes remain.

The above-mentioned problem can be solved by updating the neighbor selection rules. r_i checks whether $\mathbb{T}_i$ is equal to $\mathbb{E}_i$. If the condition is correct, r_i finds the new neighbors within the area to be repaired. Let P_u denote the set of positions of the robots located within the range of d_u. r_i defines its heading $\mathbf{h}$ with respect to the

Algorithm-3 Self-Reparation

FUNCTION $\varphi_{reparation}(O_i, p_i)$
IF $(\mathbb{T}_i = \mathbb{E}_i)$ **THEN**
  $\mathbf{h}$:= heading direction
  P_u := {set of robot positions located in the range of d_u }
  $p_{ref} := \min_{p_u \in P_u} [|ang(\mathbf{h}, \overrightarrow{p_i p_u})|]$
  p_{rn} := farthest position in the right-hand dir. of $\overrightarrow{p_i p_{ref}}$
  p_{ln} := farthest position in the left-hand dir. of $\overrightarrow{p_i p_{ref}}$
  $A(r)$:= reparation area
  P_r := { set of positions of robots located in $A(r)$ }
  $p_{s1} := \min_{p \in P_r - \{p_{rn}, p_{ln}\}} [dist(p_i, p)]$
  $p_{s2} := \min_{p \in \{p_{rn}, p_{ln}\}} [dist(p_{s1}, p) + dist(p, p_i)]$
END IF



(a) area of self-reparation (b) neighbor selection

Fig. 9 Illustration of Algorithm-3

local coordinates. Let $ang(\mathbf{m}, \mathbf{n})$ denote the angle between two arbitrary vectors $\mathbf{m}$ and $\mathbf{n}$. As shown in Fig. 9-(a), r_i selects the reference neighbor p_{ref} in P_u such that the value of $ang(\mathbf{h}, \overrightarrow{p_i p_{ref}})$ is minimized. r_i then checks if any neighbor exists in the area obtained by rotating $\overrightarrow{p_i p_{ref}}$ 60 degrees clockwise. If there exists one, r_i checks the next neighbor by sweeping another 60 degree clockwise. r_i continues to check until it finds a hole, then the last neighbor is defined as p_{ln}. Similarly, r_i attempts to find neighbors by rotating $\overrightarrow{p_i p_{ref}}$ counterclockwise and locate the last neighbor p_{rn}. Now the reparation area $A(r)$ is defined as the area between $\overrightarrow{p_i p_{rn}}$ and $\overrightarrow{p_i p_{ln}}$ in SB, where no element of P_u exists. As illustrated in Fig. 9-(b), r_i selects the first neighbor located the shortest distance away from p_i in $A(r)$ as p_{s1}. The second position is defined such that the total distance from p_{s1} to p_i can be minimized through either p_{rn} or p_{ln}. As a result, p_{ti} can be determined by $\varphi_{interaction}$.

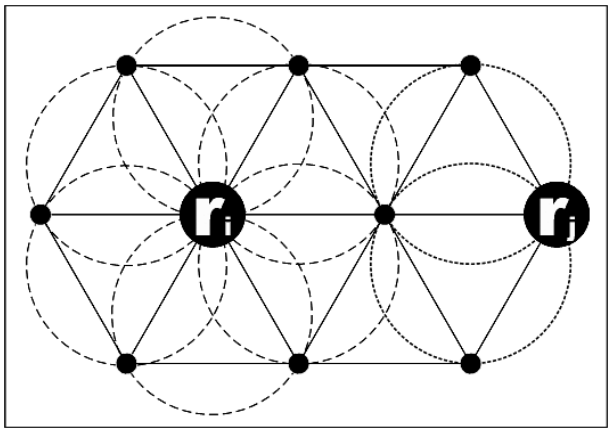


Fig. 10 Illustration of the maximum number of neighboring robots (In this figure, r_i and r_j form and maintain $\sum_{j=1}^{6}(\mathbb{E}_i)_j$ and $\sum_{j=1}^{2}(\mathbb{E}_i)_j$, respectively.

4.4 Convergence of Extended Algorithm

As illustrated in Fig. 10, like the surface tension of liquids caused by intermolecular forces, the self-reparation aims to increase the number of neighboring robots. In other words, r_i attempts to reach the maximum possible number of desired configurations given by

$$\texttt{max}[\sum_{j=1}^{s}(\mathbb{E}_i)_j] \tag{18}$$

where s is greater than or equal to one and is less than or equal to six. Here, s will not exceed a maximum of six since the desired configuration is a hexagon composed of six equilateral triangle lattices.

We define the scalar function related to self-reparation with respect to r_i as $f_{sr,i}$. As described in Algorithm-3, to obtain $\texttt{max}[\sum_{j=1}^{s}(\mathbb{E}_i)_j]$, our self-reparation algorithm shuffles the neighbors when r_i has converged into $\mathbb{E}_i$ with holes at each instant in time. Therefore, we can express the relation equation using the following scalar function:

$$f_{sr,i} = \begin{cases} \sum_{j=1}^{\hat{s}}(f_{l,i})_j + f_{sc,i} & \text{if } \mathbb{T}_i = \mathbb{E}_i \\ f_{sc,i} & \text{otherwise} \end{cases} \tag{19}$$

where $f_{sc,i}$ is given in (15) and $\hat{s}$ is less than $\texttt{max}$[s]. Using (19), (18) can be rewritten as follows:

$$f_{sr,i} = \texttt{min}[\sum_{j=1}^{s}(f_{l,i})_j]. \tag{20}$$

Here, (20) enables r_i to reach the minimum energy level by maximizing the number of $\mathbb{E}_i$.

Next, let the internal energy of r_i, increasing or decreasing during self-configuration, be denoted as q_i given by

$$q_i = \sum_{j=1}^{\hat{s}}(f_{l,i})_j. \tag{21}$$

When $\mathbb{T}_i$ is equal to $\mathbb{E}_i$, q_i forces r_i to change its neighbors toward forming another equilateral triangular lattice. We assume that q_i starts with any nonnegative value $q_i(0)$ and evolves according to the following equation:

$$\dot{q}_i = \sum_{j=1}^{\hat{s}} (\dot{f}_{l,i})_j. \tag{22}$$

Note that, by approaching new neighbors, (21) implies that q_i forces r_i to minimize $f_{sc,i}$. If q_i decreases, we can predict that r_i becomes stable, namely, $\texttt{min}[\sum_{j=1}^{s}(f_{l,i})_j]$. By doing this repeatedly, the holes in the network will be eliminated.

LEMMA 4. *Under* ALGORITHM-3*, a robot can have a maximum of six neighboring robots positioned at d_u while forming $\mathbb{E}_i$.*

proof: We use Lyapunov's theory and show the convergence of r_i using (19) and (21), defined as:

$$f_{sr,i} = f_{sc,i} + q_i. \tag{23}$$

Recall that $q_i(0)$ was initialized to a nonnegative value and evolved according to (22). Moreover, q_i was defined in such a way that it increases when $f_{l,i}$ lacks. Whenever $\mathbb{T}_i = \mathbb{E}_i$, q_i is set to $q_i(0)$. On the other hand, $f_{sc,i}$ is positive definite by Lemma 2. Therefore, Since $f_{sc,i} > 0$ and $q_i > 0$, it is clear that $f_{sr,i} > 0$.

Differentiating $f_{sr,i}$ gives

$$\dot{f}_{sr,i} = \dot{f}_{sc,i} + \dot{q}_i. \tag{24}$$

$$(\dot{f}_{sr,i} = \dot{f}_{sc,i} + \textstyle\sum_{j=1}^{\hat{s}}(\dot{f}_{l,i})_j = \dot{f}_{l,i} + \sum_{j=1}^{\hat{s}}(\dot{f}_{l,i})_j)$$

which can be simplified to:

$$\dot{f}_{sr,i} = \sum_{j=1}^{s} (\dot{f}_{l,i})_j. \tag{25}$$

Now it is easy to see that $\dot{f}_{sr,i}$ is negative definite based on Lemma 2. Similar to Lemma 2, the scalar function $f_{sr,i}$ is radially unbounded since it tends to infinity as $\| \mathbf{x} \| \rightarrow \infty$ even though q_i remains as a positive constant. Therefore, based on Lyapunov's theory, the motion of r_i under self-reparation converges into $\sum_{j=1}^{s}(\mathbb{E}_i)_j$. □

Now we prove the convergence of self-reparation for a swarm of n robots.

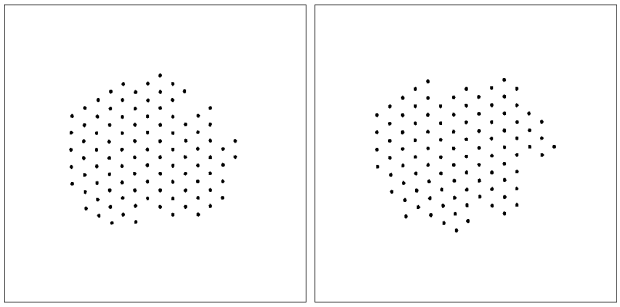
LEMMA 5. *Under* ALGORITHM-3*, all robots can have a maximum of six neighboring robots positioned at d_u while forming $\sum_{i=1}^{n} \mathbb{E}_i$.*

proof: Using Lemma 4, the n-order scalar function $\mathbf{F}_{sr}$ is defined as

$$\mathbf{F}_{sr} = \sum_{i=1}^{n} f_{sr,i} = \sum_{i=1}^{n} f_{sc,i} + \sum_{i=1}^{n} q_i. \tag{26}$$

It is straightforward to verify that $\mathbf{F}_{sr}$ is positive definite. Next, differentiating $\mathbf{F}_{sr}$ gives

$$\dot{\mathbf{F}}_{sr} = \sum_{i=1}^{n} \dot{f}_{sc,i} + \sum_{i=1}^{n} \dot{q}_i. \tag{27}$$



(a) reparation of Fig. 7 (b) reparation of Fig. 8

Fig. 11 Reconfiguration with self-reparation for each case in Fig. 7 and Fig. 8

$\dot{\mathbf{F}}_{sr}$ is negative definite and radially unbounded since it tends to infinity as t approaches infinity. Consequently, a swarm of n robots converges into $\sum_{i=1}^{n}\left(\texttt{max}[\sum_{j=1}^{s}(\mathbb{E}_i)_j]\right)$. □

THEOREM 6. *The self-configuration algorithm together with Algorithm-3 enables all robots to have a maximum of six neighboring robots positioned at d_u while forming $\sum_{i=1}^{n}\mathbb{E}_i$.*

proof: By Lemma 3, each robot can be located in $\mathbb{E}_i$, and, using Algorithm-3, the swarm can form $\sum_{i=1}^{n}\left(\texttt{max}[\sum_{j=1}^{s}(\mathbb{E}_i)_j]\right)$(Lemma 5). Consequently, these algorithms solve the self-configuration problem. □

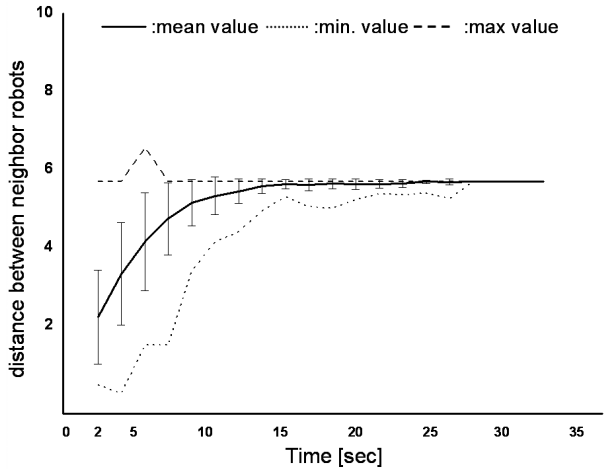
Fig. 11 presents the effects of self-reparation. Compared with Figs. 7-(c) and 8-(c), Fig. 11 present no holes and improved the connectivity between neighboring robots.

5 Simulation Results and Discussion

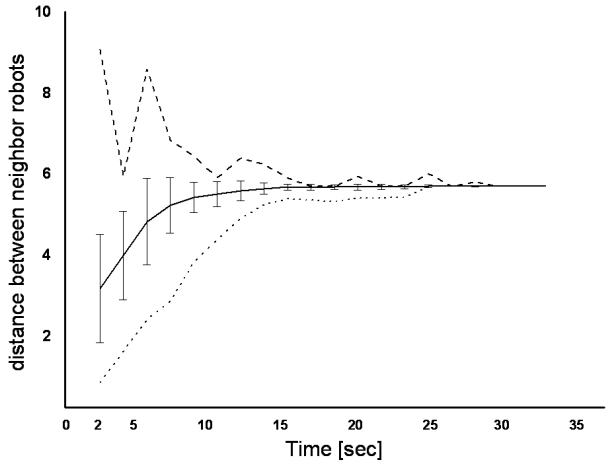
To validate our self-configuration approach, we performed extensive simulations demonstrating convergence and robustness of the algorithms.

CONVERGENCE Fig. 12 is the simulation results performed by 100 robots from a conglomerate state, where distance variations between each robot and their N_i are plotted according to time. Here, the bold line, dotted line, and dashed line indicate the mean value, minimum value, and maximum value, respectively. The error bars represent the 95% confidence intervals. As can be seen from the variation trends, each robot could converge into $\mathbb{E}_i$.

Now we investigate the effects of self-reparation under the termination condition of $d_{1\%}$. With the same initial condition as Fig. 12, Fig. 13 indicates the number of neighbors positioned d_u away from each robot according to time. Figs. 13-(a) and 13-(b) illustrate the differences when configuring the swarm without and with the self-reparation algorithm, respectively. We can largely divide their variation into four time periods. First, during the first 10 sec., each robot generated an equilateral triangle of side length d_u with their neighbors, which resulted in a significant increase of the number of robots having two neighbors at distance d_u. Secondly, from 10 sec. to 20 sec., the number of robots accompanied by zero or one neighbor decreased, while the number of robots accompanied by four and six neighbors increased in Fig. 13-(a). In



(a) distance variation between r_i and $s1$

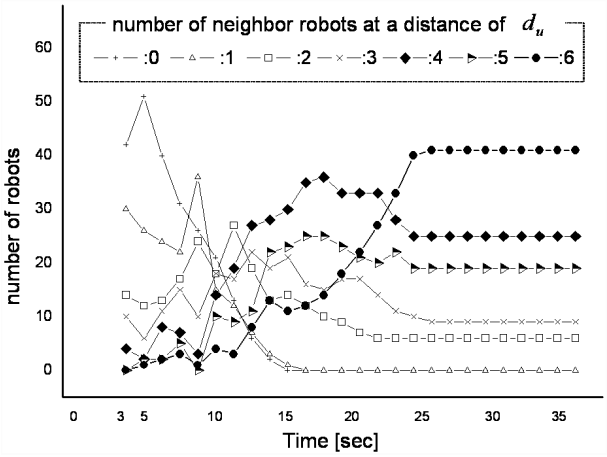


(b) distance variation between r_i and $s2$

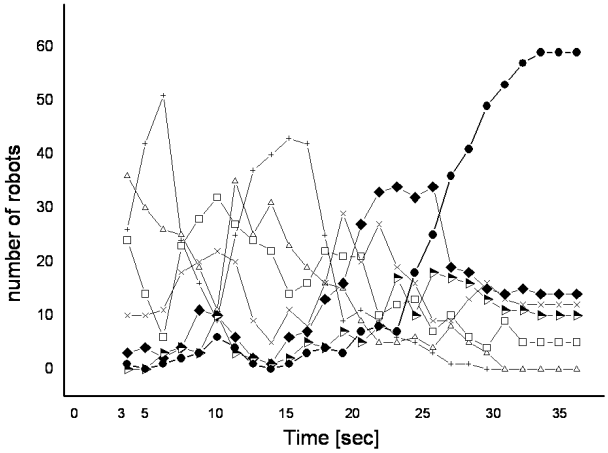
Fig. 12 Distance variations between r_i and N_i during self-configuration

contrast, Fig. 13-(b) is exceedingly complex. The lines for zero and one neighbor show regions of significant increase and decrease that later flatten out. During this period, the other lines maintain the number of neighbors almost constantly. This is caused by self-reparation that allows robots to interact with new neighbors and attempt to increase the number of neighbors. Thirdly, from 20 sec. to 25 sec., the number of neighbors gradually increased as expected. After that, there is little change. Like Fig. 11-(a), robots eventually constructed a single swarm without holes. In addition, Table I shows the comparison data for the number of robots at distance d_u after the deployment is complete with and without self-reparation. We verified that self-reparation increased the number of robots positioned at d_u by repairing holes.

From another point of view, the deployment time was examined with and without self-reparation. Table II indicates the average deployment time for 30 kinds of initial distributions of 100 robots for respective termination conditions of 1%, 0.1%, and



(a) self-configuration without self-reparation



(b) self-configuration with self-reparation

Fig. 13 Changes in the number of neighbors at d_u during self-configuration

Table 1 The number of neighbors located d_u away from each robot

No. neighbor at d_u	deployment results			
	Fig. 7	Fig. 11-(a)	Fig. 8	Fig. 11-(b)
2	4	4	3	3
3	7	4	21	12
4	18	14	22	13
5	16	12	15	12
6	55	60	39	60

0.01%. Fig. 14 shows the statistical results for the deployment time with self-reparation (scratched box) and without self-reparation (empty box). The boxes represent the inter-quartile range of the data, while the horizontal bars inside the boxes mark the median

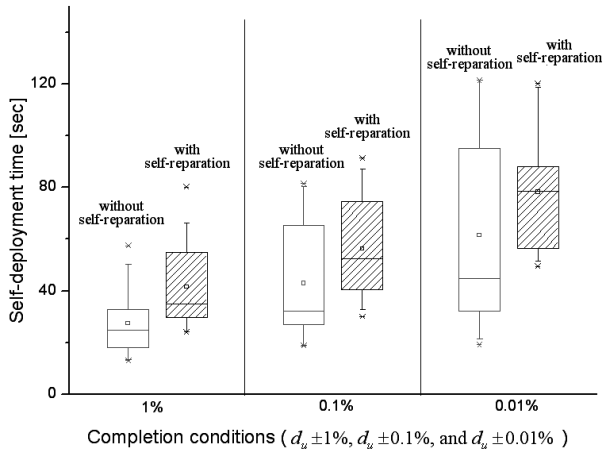


Fig. 14 Deployment time without (empty box) or with (scratched box) self-reparation for 30 kinds of different distributions

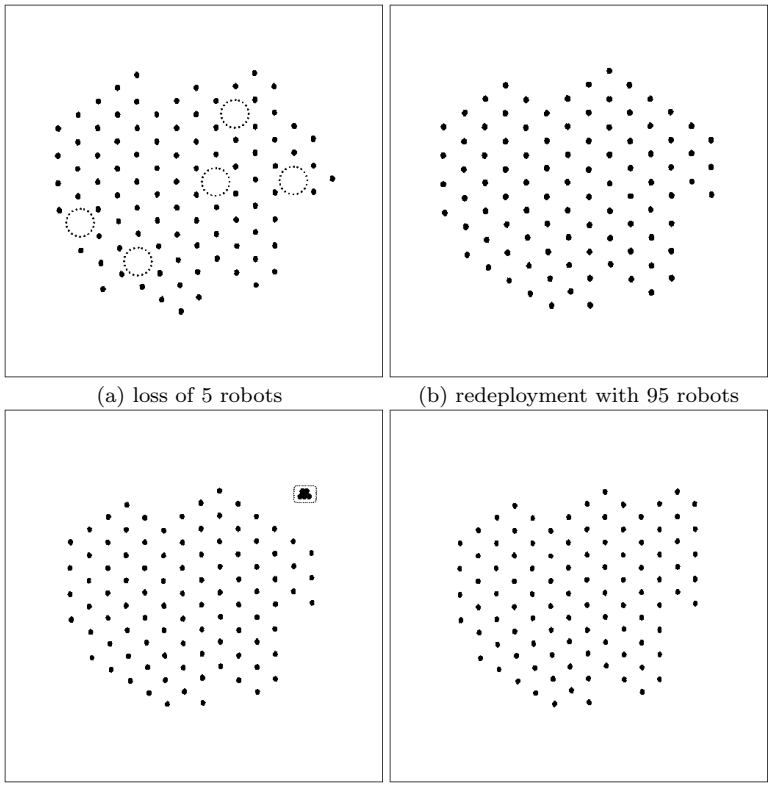
Table 2 Deployment time executing Algorithm-2 with/without self-repairing [sec]

average configuration time	Algorithm-2 without self-reparation	Algorithm-2 with self-reparation
within 1 %	27.36	41.54
within 0.1 %	42.86	56.28
within 0.01 %	61.48	78.22

values. From the results, we see that self-reparation does not require more time with respect to any particular termination condition.

ROBUSTNESS Robustness is verified against disappearances of robot members due to accidental failures. The initial condition is equal to Fig. 8-(a) with 100 robots. From the result of Fig. 11-(b), 5 robots and 10 robots disappeared in Fig. 15 and Fig. 16, respectively, and accordingly the same number of holes appeared. Each robot checks the existence of holes within *SB*. If there is a hole around it, it executes Algorithm-2 together with Algorithm-3. By the algorithms, robots approached new neighbors, and then holes disappeared. Fig. 15-(b) and Fig. 16-(b) present the results of re-deployment with 95 robots and 90 robots. In addition, we replaced the failed robots in Fig. 15-(c) and Fig. 16-(c). Fig. 15-(d) and Fig. 16-(d) show the results of re-deployment when new robots are included. From the simulation results, Algorithm-2 with self-reparation has proven to be effective in improving the robustness of robotic sensor networks against robot failures.

Four primary emphases discriminate our geometric approach from other works. First, an equilateral triangle lattice network is built with a partially connected mesh topology. Among all the possible types of regular polygons, the equilateral triangle lattices can reduce the computational burden and become less influenced by other robots, due to the limited number of neighbors. It is also highly scalable. Secondly, each robot utilizes only distance information of other robots. On the contrary, many related works require the computation of relative velocities or accelerations to calculate attractive or repulsive forces. Thirdly, robots compute the target position without



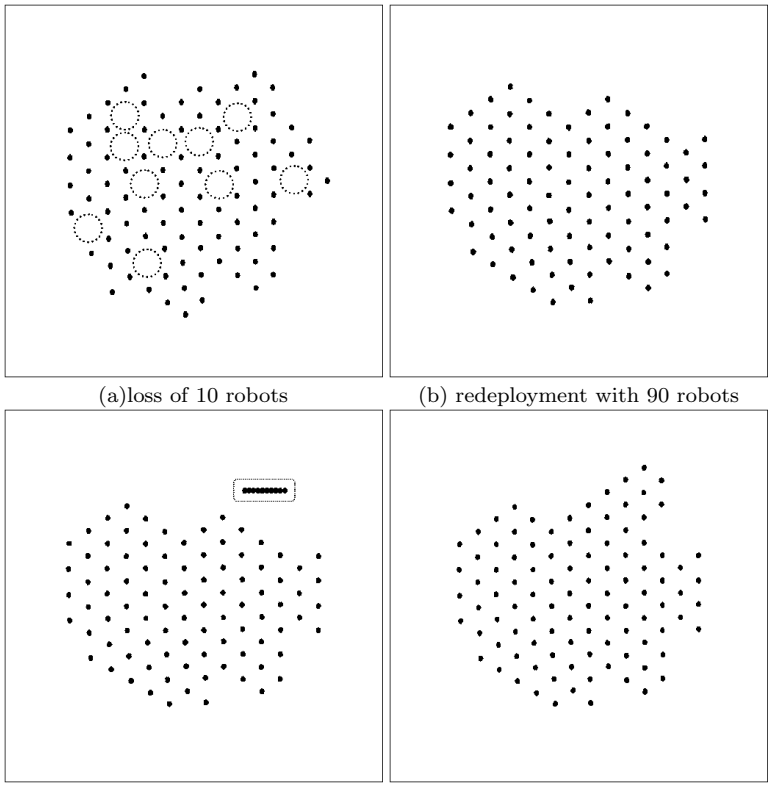
(a) loss of 5 robots (b) redeployment with 95 robots

(c) replaced 5 robots (d) redeployment with 100 robots

Fig. 15 Simulation results for robustness against loss of 5 robots

requiring memories of past actions or states. This oblivious algorithm can effectively cope with transient errors. It is known that nonoblivious algorithms might not work in situations where robots are activated asynchronously, or robots are newly added to or removed from swarms at any time. Most importantly, we attempt to solve the swarm configuration problem by eliminating such major assumptions as robot identification numbers, common coordinates, and global orientation, often made in other works.

To enable the successful configuration based on our minimal model, the proposed algorithms rely on the fact that robots can exactly sense the positions of neighboring robots, using, for instance, sonar sensors [42] or infrared sensors [19]. Moreover, it requires robots to be able to distinguish other robots from various objects in the environment. As an alternative way, when direct communications are employed, robots need to have *a priori* knowledge such as individual identification numbers or global coordinates [43][44]. Thus, direct communications may also be faced with many difficulties including the limited bandwidth, range, and interference. From a practical standpoint, the observation problem will remain open.



(a)loss of 10 robots (b) redeployment with 90 robots

(c) replaced 10 robots (d) redeployment with 100 robots

Fig. 16 Simulation results for robustness against loss of 10 robots

6 Conclusion

In this paper, we addressed the self-configuration problem of a swarm of autonomous mobile robots in a two-dimensional plane based on a robotic model with minimum functionality. For the solution to the problem, we proposed a geometric approach whereby robots could be configured into equilateral triangular lattices by locally interacting with neighboring robots in a selective way. Specifically, robots were allowed to dynamically select and interact only with two neighbors. Collecting this local behavior, the swarm could be self-deployed with uniform spatial density in a coordinated manner. Moreover, to cope with possible holes in the deployment, the self-reparation capability enabled the robots to increase the number of neighboring robots positioned at the uniform distance. This will allow the robot swarm to improve the network connectivity. The proposed algorithm features decentralization, self-organization, self-stabilization, and robustness, which was proved mathematically and verified through extensive simulations. Our analysis and simulation results showed that the proposed self-configuration method is a simple and efficient approach to the deployment of robotic sensor devices.

References

1. G. Beni: "From swarm intelligence to swarm robotics" in SAB 2004: E. Sahin, W. M. Spears (eds.): Swarm Robotics. Lecture Notes in Computer Science. Vol.3342. Springer-Verlag (2005) 1-9
2. E. Sahin: "Swarm robotics: from sources of inspiration to domains of application" in SAB 2004: E. Sahin, W. M. Spears (eds.): Swarm Robotics. Lecture Notes in Computer Science. Vol.3342. Springer-Verlag (2005) 10-20
3. H. Choset: Coverage for robotics-a survey of recent results. Annals of Mathematics and Artificial Intelligence. Vol.31 (2001) 113-126
4. J. Cortes, S. Martinez, T. Karatas, F. Bullo: Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation. Vol.20, No.2 (2004) 243-255
5. I. Suzuki, M. Yamashita: Distributed anonymous mobile robots: formation of geometric patterns. SIAM Journal of Computing. Vol. 28, No. 4 (1999) 1347-1363
6. X. Defago, A. Konagaya: Circle formation for oblivious anonymous mobile robots with no common sense of orientation. Proc. 2nd ACM Intl. Workshop on Principles of Mobile Computing (2002) 97-104
7. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer.: Pattern formation by autonomous robots without chirality. Proc. SIROCCO (2001) 147-162
8. S. Carpin, L. E. Parker: Cooperative leader following in a distributed multi-robot system. Proc. IEEE International Conference on Robotics and Automation (2002) 2994-3001
9. Z. Cao, M. Tan, S. Wang, Y. Fan, B. Zhang: The optimization research of formation control for multiple mobile robots. Proc. 4th World Congress on Intelligent Control and Automation (2002) 1270- 1274
10. F. Mondada, G.. C. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M.Gambardella, M. Dorigo: Swarm-bot: a new distributed robotic concept. Autonomous Robots. Vol.17, No.2-3 (2004) 193-221
11. G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, S. Nolfi: Self-organized coordinated motion in groups of physically connected robots. IEEE Transactions on Systems, Man, and Cybernetics - Part B. Vol.37, No.1 (2007) 224-239
12. Y. Ikemoto, Y. Hasegawa, T. Fukuda, K. Matsuda: Graduated spatial pattern formation of robot group. Information Sciences. Vol.171, No.4 (2005) 431-445
13. A. Ishiguro, T. Kawakatsu: Self-assembly through the interplay between control and mechanical systems. Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) 631-638
14. M. Shimizu, T. Mori, A. Ishiguro: A development of a modular robot that enables adaptive reconfiguration. Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) 174-179
15. B. Werger, M. J. Mataric: From insect to internet: situated control for networked robot teams," Annals of Mathematics and Artificial Intelligence. Vol.31 (2001) 173-198
16. T. Balch, M. Hybinette: Social potentials for scalable multi-robot formations. Proc. IEEE International Conference on Robotics and Automation (2000) 73-80
17. A. Howard, M. J. Mataric, G. S. Sukhatme: Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. Proc. 6th International Symposium on Distributed Autonomous Robotic Systems (2002) 299-308
18. W. M. Spears, F. D. Gordon: Using artificial physics to control agents. Proc. IEEE International Conference on Information, Intelligence, and Systems (1999) 281-288
19. W. Spears, D. Spears, J. Hamann, R. Heil: Distributed, physics-based control of swarms of vehicles. Autonomous Robots. Vol.17, No.2-3 (2004) 137-162
20. K. Fujibayashi, S. Murata, K. Sugawara, M. Yamamura. Self-organizing formation algorithm for active elements. Proc. 21st IEEE Symposium on Reliable Distributed Systems (2002) 416- 421
21. J. McLurkin, J. Smith: Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. Proc. 7th International Symposium on Distributed Autonomous Robotic Systems. (2004) 831-890
22. B. Shucker, T. Murphey, J. K. Bennett: Switching rules for decentralized control with simple control laws. Proc. American Control Conference (2007) 1485-1492
23. J. Reif, H. Wang: Social potential fields: a distributed behavioral control for autonomous robots. Robotics and Autonomous Systems. Vol.27, No.3 (1999) 171-194
24. Y. Zou, K. Chakrabarty: Sensor deployment and target localization based on virtual forces. Proc. IEEE Infocom Conference (2003) 1293-1303

25. N. Heo, P. K. Varshney: A distributed self spreading algorithm for mobile wireless sensor networks. Proc. IEEE Wireless Communication and Networking Conference (2003) 1597-1602
26. R. Cohen, D. Peleg: "Local algorithms for autonomous robots systems," in SIROCCO 2006: P. Flocchini, L. Gasieniec (eds.): Lecture Notes in Computer Science. Vol.4056. Springer-Verlag (2006) 29-43
27. G.-L. Wang, G. Cao, T. L. Porta: Movement-assisted sensor deployment. Proc. IEEE Infocom Conference (2004) 2469-2479
28. E. Martison, D. Payton: "Lattice formation in mobile autonomous sensor arrays" in SAB 2004: E. Sahin, W. M. Spears (eds.): Swarm Robotics. Lecture Notes in Computer Science. Vol.3342. Springer-Verlag (2005) 98-111
29. S. Ghosh, K. Basu, S. K. Das: An architecture for next-generation radio access networks. IEEE Network. Vol.19, Is.5 (2005) 35-42
30. T. Fowler: Mesh networks for broadband access. IEEE Review. Vol.47, Is.1 (2001) 17-22
31. P. C. Gurumohan, J. Hui: Topology design for free space optical networks. Proc. 12th International Conference on Computer Communications and Networks (2003) 576-579
32. V. Gazi, K. M. Passino: Stability analysis of swarms. IEEE Transaction on Automatic Control. Vol.48, No.4 (2003) 692-696
33. R. Olfati-Saber, R. Murray: Distributed cooperative control of multiple vehicle formations using structural potential Functions. Proc. 15th IFAC World Congress on Automatic Control (2002) 21-26
34. P. Ogren, E. Fiorelli, N. E. Leonard: Formations with a mission: stable coordination of vehicle group maneuvers. Proc. 15th International Symposium on Mathematical Theory Networks and Systems (2002)
35. J. E. Slotine, W. Li: Applied nonlinear control. Prentice-Hall (1991)
36. C.-T. Chen: Linear system theory and design. 3rd edn. Oxford University Press (1999)
37. S. Dolev: Self-Stabilization. MIT Press (2000)
38. M. Schneider: Self-stabilization. ACM Computing Survey. Vol.25, No.1 (1993) 45-67
39. R. Lyengar, K. Kar, S. Banerjee: Low-coordination topologies for redundancy in sensor networks. Proc. ACM MobiHoc. (2005) 332-342
40. I. Stojmenovic, J. Wu: "Broadcasting and activity scheduling in ad hoc networks": S. Basagini, M. Conti, S. Giordano, I. Stojmenovic (eds.): Mobile Ad Hoc Networking. IEEE press (2004) 205-229
41. F. Harary: Graph Theory. Addison-Wesley (1972)
42. G. Lee, N. Y. Chong: Decentralized formation control for a team of anonymous mobile robots. Proc. 6th Asian Control Conference (2006) 971-976
43. M. Lam, Y. Liu: ISOGIRD: an efficient algorithm for coverage enhancement in mobile sensor networks. Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) 1458-1463
44. J. Nembrini, A. Winfield, C. Melhuish: Minimalist coherent swarming of wireless networked autonomous mobile robots. Proc. International Conference on Simulation of Adaptive Behavior (2002) 373-382
45. R. Prakash, R. Baldoni: Architecture for group communication in mobile systems. Proc. 17th IEEE Symposium on Reliable Distributed Systems (1998) 235-242