

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



# Dispersion and Dispatch Movement Design for a Multi-Robot Searching Team Using Communication Density

Feng-Li Lian, You-Ling Jian and Wei-Hao Hsu  
*Department of Electrical Engineering, National Taiwan University  
 Taipei, Taiwan*

## 1. Introduction

In recent years, since the cooperation of multiple robots provides numerous advantages in engineering applications, the research topics on multi-robot systems have received great attention. Due to the complexity and/or difficulty of a task, the performance by one single robot is usually limited. However, the performance limitation could be overcome by the cooperation of multiple robots. Compared with one single robot, a multi-robot team could provide advantages such as better efficiency, robustness, execution time saving, and so on, which are mainly achieved because of the characteristics of multi-robot cooperation. In a multi-robot team, the possible abilities of each robot can be usually classified as the following four types: communication, sensing, computation, and actuation. Among these abilities, the communication between robots is the most important factor for the cooperative characteristics of a multi-robot team. If every robot is considered as a networking node, the communication links between any pair of robots form a communication network. In order to guarantee the cooperative characteristics of a multi-robot team, the method of maintaining a good communication network thus becomes a critical issue.

Considering the robot mobility, environmental limitation, user requests or other factors, many moving algorithms driving robots to form a good communication network are previously proposed. In most of these algorithms, robots are assumed to be well equipped with advanced sensing or positioning devices such as global positioning system, radar and/or camera. These equipments could provide enough information for the moving algorithms and hence output good performance. Sometimes, because of the computation complexity of the algorithm, other advanced processing units are also needed. These assumptions, however, are not easily implemented in real applications. Moreover, cost or retrieve would be another important design issue for the searching task by a robot team.

To achieve the cooperative characteristics and information sharing, the team of robots must maintain a good communication connection while moving or performing tasks. In real implementation, sharing enough information quantity among multiple robots could be a challenging problem. In this chapter, inspired by the bacteria mobility behavior, a dispersion movement design for a multi-robot team is presented. The goal of the dispersion algorithm is to maintain the communication network among robots and enlarge the coverage area as widely as possible. Unlike other previously proposed algorithms, the

Source: Recent Advances in Multi-Robot Systems, Book edited by: Aleksandar Lazinica, ISBN 978-3-902613-24-0, pp. 326, May 2008, I-Tech Education and Publishing, Vienna, Austria

design of the dispersion algorithm is distributed and only utilizes simple information. The communication density, or the number of communication links, is the only information needed in the algorithm. Although more information brings better efficiency, it also causes problems like cost and retrieve. Hence, in the proposed algorithm, the position information is not required that greatly reduces the implementation cost on positioning devices. This information could be easily obtained without complex equipments such as global positioning system, camera or radar. Hence the advantage of low information quantity can be achieved. Moreover, the bacteria foraging behavior along with attraction and repulsion model of an organism is also integrated into the algorithm. These biomimic behaviors help the robot team overcome the limitations caused by lacking of position information.

For a base station that releases robots to search for targets, a dispatch rule is also designed based on the dispersion algorithm. Because of the communication and sensing limitations, the coverage area by robots is limited. In order to enlarge the coverage area, "the base station needs to add new robots. Two key factors decided by the base station are "when to release robots" and "how many robots to be released." Therefore, with the dispatch rule, the base station could flexibly re-supply robots to enlarge the coverage area adaptively.

### 1.1 Related Work

Multi-robot systems are nowadays a popular research topic. In many applications, exploring an unknown environment is an important research task. Robots equipped with sensors form a so-called "sensor network" to explore or map the environment. By considering energy consumption, communication quality and scalability, the communication topology of a team of robots is an important issue related to the sensor network system. Wireless sensor networks are generally used in environment monitoring or exploring. The robots are equipped with sensors such as thermal, pressure or moisture sensors. With information collected by the networked sensors, the status about the environment could be completely obtained. The robots in wireless sensor networks are generally static and distributed. Due to the issue of signal transmission degradation, these robots should select a proper set of neighbors for establishing communication links. This kind of mechanism is called topology control. Topology control for a static sensor network is widely studied and many protocols are proposed: (Santi, 2005; Rodoplu & Meng, 1999; Li et al., 2005; Blough et al., 2003). However, a multi-robot system is usually mobile. Hence the deployment of a team of robots becomes an important issue in a mobile sensor network (Heo & Varshney, 2005; Chellappan et al., 2007).

Mobile sensor networks are not fully considered in many proposed topology control protocols. In the robotics research area, movement algorithms for the mobile multi-robot systems have been significantly studied. Based on the theory of graph theory or potential field, a variety of algorithms are proposed for achieving different movement purposes. For exploring or monitoring the environment, a team of robots need to be distributed uniformly in an environment, and/or they need to form a certain formation for improving the operation efficiency. To achieve these requirements, many movement algorithms are proposed. For example, Tan proposed a distributed model for a multi-robot system by utilizing Delaunay Triangulation and an autonomous self-deployment algorithm is then discussed (Tan, 2005). Cortes et al. proposed a coverage control by focusing on the Voronoi cell maintenance of each robot. A gradient descent algorithm for a class of utility functions is then proposed to encode the optimal coverage and sensing policy (Cortes et al., 2004). Burgard et al. focused on the target points and set cost functions to decide the movement of robots (Burgard et al., 2005).

Poduli and Sukhatme considered the problem of maximizing the total sensor coverage area with the constraint of at least  $K$  neighbors linked with every robot. Two virtual forces of attraction and repulsion are defined between robots to force them to move until at least  $K$  neighbors is within the neighbor set (Poduli & Sukhatme, 2004).

In addition to the algorithms developed based on the analytic method of potential field and graph theory, each robot in a multi-robot team can be considered as an organism, and the network formed by robots can be viewed as an organism aggregation. The aggregation behavior in nature is widely studied in biological research area. Typical force components of an aggregation model include locomotory, aggregative, arrayal and random (Flierl et al., 1999; Parrish et al., 2002; Czirok et al., 1996; Gueron et al., 1996). For the aggregative force, long-range attraction and short-range repulsion is the most typical model (Gueron et al., 1996; Gazi & Paasino, 2003; Gazi & Passino, 2004). When the neighbors get too close to the individual, a repulsion force is caused to part the individual from those too-closed neighbors. On the other hand, the neighbors that are a little far from the individual will cause an attraction force to the individual. The concept of the attraction-repulsion force model is later applied into the multi-robot dispersion movement algorithm to maintain the communication network.

In nature, the *E. coli* bacteria have a foraging mechanism for searching for food and avoiding noxious substances. The flagellum of bacteria causes two kinds of actions: (1) the swimming action, which is to make the cell move forward, and (2) the tumbling action, which is to make the cell change direction. When in the isotropic homogeneous environment, the bacteria alternate swim with tumble, and, hence, they move in random walks. This enables the bacteria to search for nutrients. While the bacteria encounter a nutrient gradient, they spend more time swimming and less time tumbling. The direction is biased towards increasing nutrient gradient, and the nutrient concentration data of past steps are used to decide how long to run. In other words, although the bacteria do not have any positional information about the environment concentration, they use the “time density” instead of “spacial density” to make movement decisions (Passino, 2002). In the target search problem studied, to maintain a proper communication network, one robot need to keep a certain number of communication links. In other words, the “communication density” is desired to reach a certain value. This is similar to the bacteria searching of nutrient. Hence the bacteria foraging behavior is applied as a driving force for robots to move to the area with a desired communication density.

Many algorithms utilizing large information quantity are proposed for multi-robot systems applied to environment exploration, sensor network, formation control and other topics. In this chapter, the multi-robot deployment movement problem with a dispatch mechanism is discussed. The design of the dispersion algorithm is distributed and only utilizes simple information. Although more information brings better efficiency, it also causes problems of high cost and retrieving difficulty. Hence, in the proposed algorithms, the position information is not required that greatly reduces the implementation cost on positioning devices. Moreover, the bacteria foraging behavior along with the attraction and repulsion model of organism is also integrated into the algorithm. These biomimic behaviors help the robot team to overcome limitations for lack of position information.

## 1.2 Outline

The rest of the chapter is organized as follows. Section 2 formulates the dispatch and dispersion problem. Section 3 presents the dispersion movement algorithm of multi-robot

systems inspired by the bacteria foraging behavior and the aggregation of creatures. Section 4 presents the control rule of dispatching robots at the base station. Section 5 discusses the scenario and the statistics of related simulation results. Finally Section 6 summarizes this chapter.

## 2. Problem Formulation

In the section, a target search problem combined with a multi-robot dispersion movement and a robot dispatch rule in a base station is defined. First, the unknown environment is assumed to be on a two-dimensional plane without boundary and obstacles. Certain target exists at unknown position in the environment, and robots are released from a base station for searching for the target. Since the position of the target is unknown, an appropriate number of robots are required to be able to cover a certain area and the number is not decided in advance. Hence the base station needs to release robots continuously until the target is found. Once the target is sensed by any of the robots, the target search task is finished.

Figure 1 shows a schematic diagram of the target search problem. The square in the center of the plane denotes the base station which is responsible for releasing robots. The circular patterns denote the robots which have been released, and the star symbol denotes the target. Since the position of the target is unknown, the size of required coverage area by a team of robots could not be estimated in advance. That is, the appropriate number of robots that are able to sense the target can not be decided in advance. To solve this problem, the base station needs to release robots continuously for enlarging the coverage area until the target is found. Once the target is sensed by one of the robots, the target search task is finished.

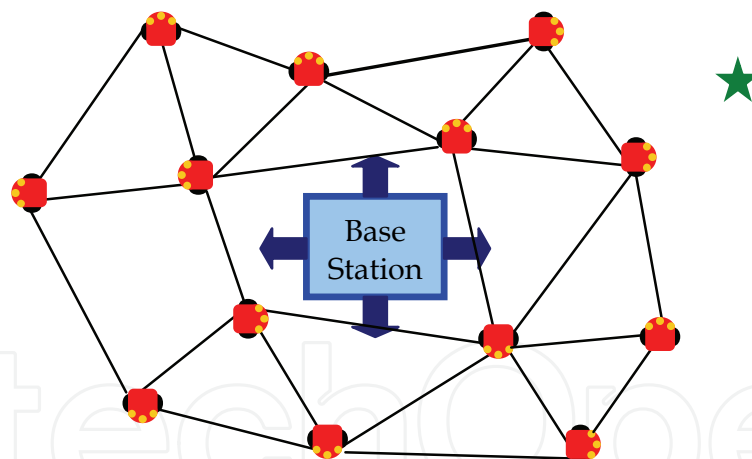


Figure 1. Schematic diagram showing the target search problem

For a multi-robot system, information sharing is one of the key advantages compared with a single-robot system. In general, the communication between robots has a limited range, defined as the “*communication range*” of the robots. A schematic diagram is shown in Figure 2. The black dots denote the robots. The large circle denotes the communication range of Robot 1. Robot 2 is inside the communication range of Robot 1. Hence Robot 2 is called a “*neighbor*” of Robot 1. On the other hand, Robot 3 is outside the communication range of Robot 1. Therefore Robot 3 is not a neighbor of Robot 1. The sensible range of robots for the target is also limited. The range is defined as “*sensing range*”, which is denoted by the small gray circle in Figure 1.

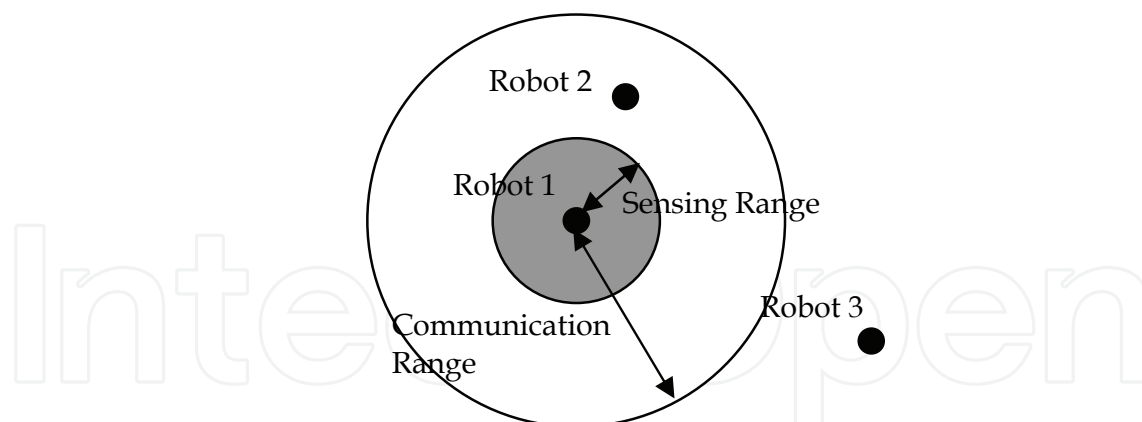


Figure 2. Schematic diagram showing the sensing and communication ranges of a robot  
For the target search problem, there are two major tasks: (1) the dispatch manner at the base station, and (2) the dispersion manner on each individual robot.

### 2.1 Dispatching at the Base Station

The duty of the base station is to release robots until the target is found. Since there is no prior information of the location of the target, and these robots need time to spread out, the base station should decide the following two issues: “*when* to release new robots” and “*how many* new robots to be released.” To make a proper decision, the base station needs some information of the dispersion manner of the robots in advance, or it can receive real-time information from the robots.

### 2.2 Dispersion of Robots

Once robots are released from the base station, they start to move. To simplify the problem, these robots are set to move along the following four directions: go forward, go backward, move right and move left, as shown in Figure 3. Also the velocity of each robot is assumed to be the same. During the movement, the robots should spread out as widely as possible while keeping the communication network connected. Many movement algorithms are developed previously and can provide good efficiency. However, they are usually designed based on rich information such as the position of robots with specialized equipments. Sometimes a leader is also needed to direct other robots. The main objective of this chapter is to develop a dispersion algorithm for robots that only uses simple and easily obtained information. Hence, simple distributed computation can be implemented on each robot.

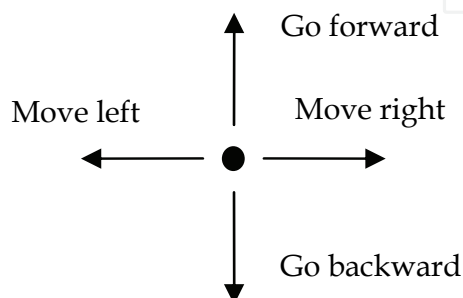


Figure 3. Action definition of a single robot

### 3. Dispersion Algorithm

In the target search problem studied, to maintain a proper communication network, robots need to keep a certain number of communication links. In other words, the “communication density” is desired to reach a certain value. This is similar to the bacteria searching of nutrient. Hence the bacteria foraging behavior is applied as a driving force for robots to move to the area with a desired communication density. Moreover, the advantage of not using the position information is achieved. This helps decrease the implementation cost and solve problems like retrieving. Simply speaking, the algorithm works as follows. When the neighbors of the robot get too close to the robot, a repulsion force is caused to separate the robot from those too-closed neighbors. On the other hand, the neighbors that are a little far from the robot generate an attraction force to the robot. This attraction-repulsion concept is combined together with the bacteria foraging behavior to design the dispersion algorithm.

Symbol	Definition
$i$	Robot ID
$k$	Time step index
$t$	Releasing index of base station
$R_s$	Radius of sensing range
$R_c$	Radius of communication range
$R_r$	Radius of repulsion range
$n_c^i(k)$	Number of neighbors in communication ring range of Robot $i$ at Step $k$
$n_r^i(k)$	Number of neighbors in repulsion range of Robot $i$ at Step $k$
$N_c$	Desired number of neighbors in communication ring range
$A^{total}(k)$	Real coverage area of total robots at Step $k$
$A_{est}^{total}(k)$	Estimated coverage area of total robots at Step $k$
$R_a^{total}(k)$	Radius of real coverage area of total robots at Step $k$
$R_{a,est}^{total}(k)$	Radius of estimated coverage area of total robots at Step $k$
$A^i(k)$	Real coverage area of Robot $i$ at Step $k$
$A_{est}^i(k)$	Estimated coverage area of Robot $i$ at Step $k$
$R_{a,eff}$	Radius of effective coverage area of a single robot
$T_{strait}$	Length of steps compared in Case Strait
$T_{normal}$	Length of steps compared in Case Normal
$T_{escape}$	Length of moving steps in Case Strait
$p(k)$	Stop percentage of robots at Step $k$
$P$	Desired value of stop percentage of robots
$T_t$	Releasing time of the $t^{th}$ releasing of base station
$u_t$	Number of releasing robots at the $t^{th}$ releasing of base station
$\Delta R$	Desired increasing radius of total coverage area
$N(k)$	Number of total robots at Step $k$
$D^i(k)$	Moving direction of Robot $i$ at Step $k$

Table 1. Parameter Definitions of Dispersion Algorithm and Dispatch Rule

The parameters of the dispersion algorithm are defined in Table 1. These parameters are grouped as indexes, the radiuses of different ranges, the number of neighbors, the definitions of different areas and radiuses, the lengths of time steps, the stop percentage of

robots, the two releasing parameters of base station, and other parameters. All of these parameters will be explained in detail later when mentioned in this discussion.

A new range, called “*repulsion range*”, is defined as shown in Figure 4. It is for designing the repulsion force for the neighbors of the robot. Moreover, the communication range excluding the repulsion range is called the “*communication ring range*,” inside which its neighbors cause an attraction force to keep the network from being partitioned. Let the number of neighbors inside the communication ring range be  $n_c^i(k)$ , where  $k$  denotes the time step and  $i$  is the robot ID. A desired value of  $n_c^i(k)$  is set as  $N_c$ . Also let the number of neighbors inside the repulsion range be  $n_r^i(k)$ , and the desired value of  $n_r^i(k)$  is zero because of the repulsion. An example of  $n_c^i(k) = 3$  and  $n_r^i(k) = 1$  is shown in Figure 4.

The flowchart of the dispersion algorithm is shown in Figure 5. First the robot needs to decide whether it should move or not. Since it wants to keep  $n_c^i(k)$  equal to  $N_c$  and  $n_r^i(k)$  equal to zero, then the *moving condition* is defined as follows.

$$n_c^i(k) \neq N_c \parallel n_r^i(k) \neq 0 \quad (1)$$

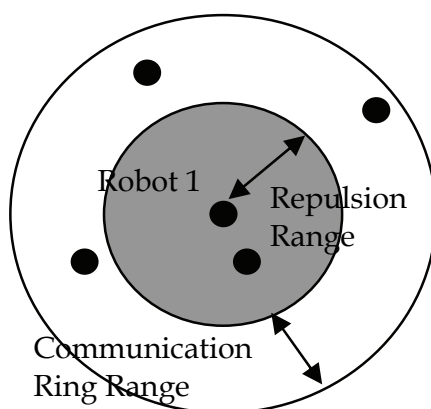


Figure 4. Schematic diagram showing the repulsion and communication ranges of a robot

However, in order to avoid the partition of the communication network, a slight adjustment is applied to the moving condition. The probability of network partition increases when the moving robots have only a few neighbors within the communication range. Therefore they should take a conservative action to avoid partition; that is, they should stop at such situation. Hence the moving condition is further adjusted as follows.

$$(n_c^i(k) \neq N_c \ \&\& \ n_c^i(k) > \frac{N_c}{2}) \parallel n_r^i(k) \neq 0 \quad (2)$$

For the condition that  $n_c^i(k) \neq N_c$  and  $n_c^i(k) > N_c/2$ , it is defined as the “*Phase A*,” indicating the attraction force stage. For the condition that  $n_r^i(k) \neq 0$ , it is defined as the “*Phase R*,” indicating the repulsion force stage. Since keeping the network connected is of most important, the attraction force is considered prior to the repulsion force.

Once the moving condition is satisfied, the robot starts to move. For simplicity, the robot moving velocity is set to be the same. Hence the only one variable that the robot needs to



decide is the movement direction. Here the bacteria foraging behavior is applied for choosing the direction.

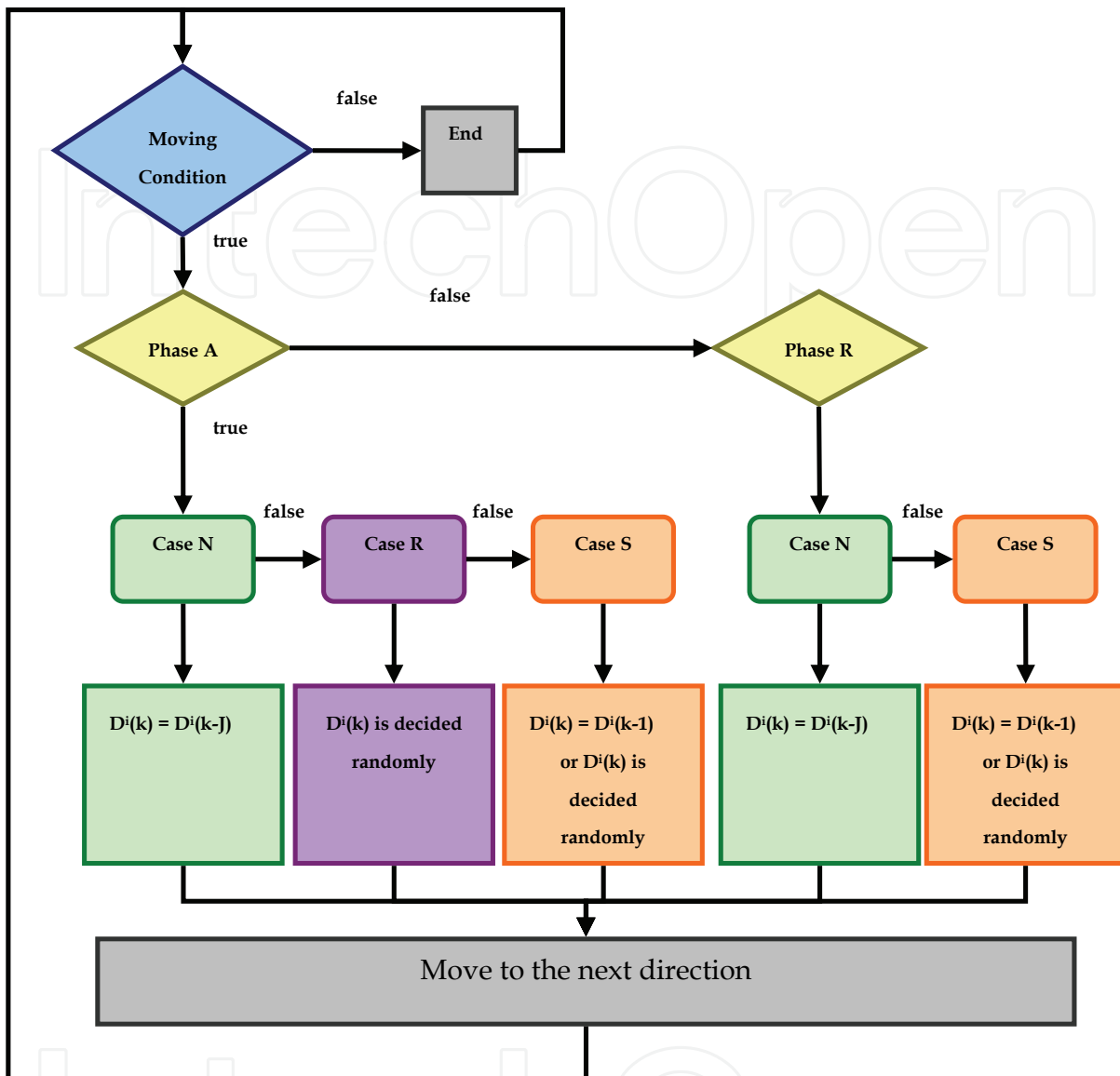


Figure 5. Flowchart of dispersion algorithm

Let the moving direction of Robot  $i$  at Step  $k$  be  $D^i(k)$ . First, every robot records  $D^i(k)$ ,  $n_c^i(k)$ , and  $n_r^i(k)$  over the last  $T_{strait}$  steps. Moreover, a small value of buffer length  $T_{normal}$  is defined.  $T_{strait}$  is set as several times of  $T_{normal}$ . According to the recorded values of  $n_c^i(k)$  and  $n_r^i(k)$ , three cases, namely "Case Normal", "Case Strait", and "Case Random" are specified.

Each case also includes Phase A and Phase R, respectively. The scenario in Phase A is discussed in the following, and minor adjustment is given for that used in Phase R.

**Case Normal:** The values of  $n_c^i(k)$  recoded in the  $T_{normal}$  buffer are not all the same.

In Case Normal, the robot implements the behavior of climbing nutrient gradient of the E. coli bacterium. The direction of the E coli bacterium is biased toward increasing nutrient gradient. The bacterial cell compares the concentration of the environment over the past 1

time step with the concentration over the last 3 time steps before that. Hence it uses the nutrient concentration data of the past 4 time steps to decide its next movement.

$n_c^i(k)$  here is similar to the concentration of nutrition. Therefore,  $n_c^i(k)$  is compared with those of the last  $T_{normal}$  steps. The direction  $D^i(k-j)$  at Step  $k-j$  ( $j \in \{1, \dots, T_{strait}\}$ ) where  $n_c^i(k-j)$  is close to  $N_c$  is believed to be the direction leading  $n_c^i(k)$  to  $N_c$ . So the robot chooses  $D^i(k-j)$  as the next direction. That is,

$$D^i(k) = D^i(k-j) \quad j = \underset{t=1-\text{buffer}}{\text{arg min}} |n_c^i(k-t) - N_c| \quad (4)$$

An example of the recorded buffer of  $n_c$  at Step  $k$  is shown in Table 2.  $T_{normal} = 10$  and  $D$  and  $n_c$  recorded in the last  $T_{normal}$  steps are shown in Table 2. If  $N_c$  is assumed to be 5,  $n_c$  in Step  $k-4$  and Step  $k-9$  are both closest to the desired value of  $N_c$ . Since Step  $k-4$  is the later step, the direction of Step  $k-4$  is then adopted and the next moving direction is determined as "Forward".

Step Index	$k-1$	$k-2$	$k-3$	$k-4$	$k-5$	$k-6$	$k-7$	$k-8$	$k-9$	$k-10$
$n_c(k-j)$	4	3	4	5	6	6	7	6	5	3
$D(k-j)$	Left	Right	Left	Forward	Back	Back	Right	Left	Left	Forward

Table 2. An example of Case Normal

**Case Strait:** The values of  $n_c^i(k)$  recoded in the  $T_{strait}$  buffer are all the same.

In Case Strait, the robot implements the escaping behavior of the E. coli bacterium from a homogeneous concentration environment. If the concentration of the environment does not change for a long time, the bacterium increases the mean run length and decreases the mean tumble time. In other words, the bacterium runs a longer distance with a fixed direction.

Once  $n_c^i(k)$  recorded in the  $T_{strait}$  buffer are all the same, the robot randomly selects a direction and runs for  $T_{escape}$  steps.  $T_{escape}$  is about the order of  $T_{strait}$  and is defined by users.

An example of the recorded buffer of  $n_c$  at Step  $k$  is shown in Table 3.  $T_{normal} = 5$ ,  $T_{strait} = 10$ ,  $T_{escape} = 20$  and  $D$  and  $n_c$  recorded in the last  $T_{strait}$  steps are as shown in Table 3. Since all data of  $n_c$  in the last  $T_{strait}$  steps are all the same, the robot will choose a random direction and run for  $T_{escape}$  steps. For instance, the robot may keep going forward for the next 20 steps.

Step Index	k-1	k-2	k-3	k-4	k-5	k-6	k-7	k-8	k-9	k-10
$n_c(k-j)$	4	4	4	4	4	4	4	4	4	4
$D(k-j)$	Left	Right	Left	Forward	Back	Back	Right	Left	Left	Forward

Table 3. An example of Case Strait

**Case Random:** The values of  $n_c^i(k)$  recoded in the  $T_{normal}$  buffer are all the same, but the  $T_{strait}$  buffer are not all the same, or the robot does not have enough data of  $n_c^i(k)$ .

In this case, it is assumed that the robot does not have enough information about the environment. Therefore, it chooses a random direction for the next movement. The action is continued until there is enough information about the environment and the situation then changes to Case Normal or Case Strait.

An example of the recorded buffer of  $n_c$  at Step  $k$  is shown in Table 4.  $T_{normal} = 5$ ,  $T_{strait} = 10$ ,  $T_{escape} = 20$  and  $D$  and  $n_c$  recorded in the last  $T_{strait}$  steps are as shown in Table 4. Since  $n_c$  are the same in the last five steps but different in the last six to last ten steps, the robot then chooses a random direction as the next moving direction. For instance, the robot may move right in the next direction and again compare  $n_c$  in the last ten steps to decide the next direction.

Step Index	k-1	k-2	k-3	k-4	k-5	k-6	k-7	k-8	k-9	k-10
$n_c(k-j)$	4	4	4	4	4	6	7	6	5	3
$D(k-j)$	Left	Right	Left	Forward	Back	Back	Right	Left	Left	Forward

Table 4: An example of Case Random

These three cases are also applied to Phase R with minor adjustment. Note that the repulsion range is smaller than the communication range. Once  $n_r^i(k)$  does not change for a period of time, it is likely to be in the strait situation. Hence, Case Random is omitted in Phase R, and  $T_{strait}$  in Case Strait is changed to  $T_{normal}$ .

With this dispersion algorithm, the final coverage area of the communication network would be mainly decided by the communication range and the repulsion range. With a larger communication range, a larger coverage area is formed. Also, with a larger repulsion range, the distance between robots increases. Hence the coverage area again gets larger. Let the radius of the communication range be  $R_c$ , and the radius of the repulsion range be  $R_r$ . Also, the *radius of effective coverage area*,  $R_{a,eff}$ , is defined as the radius of the average coverage area. Then,  $R_{a,eff}$  is a linear function of  $R_c$  and  $R_r$ , and can be expressed as follows:

$$R_{a,eff} = k_c R_c + k_r R_r \quad (5)$$

where  $k_c$  and  $k_r$  are weighting constants of  $R_c$  and  $R_r$ , respectively. Eqn. (5) can be rearranged as follows:

$$\frac{R_{a,eff}}{R_c} = k_r \frac{R_r}{R_c} + k_c \quad (6)$$

That is, the ratio of  $R_{a,eff}$  and  $R_c$  and the ratio of  $R_r$  and  $R_c$  would have a linear relationship.

#### 4. Dispatching at the Base Station

The task at the base station is to release robots into the environment for searching for unknown targets. Because of the communication and sensing limitations, the coverage area by one robot is limited. Therefore, the base station needs to add new robots to enlarge the

coverage area. Two questions decided by the base station are “when to release robots” and “how many robots to be released.” For the first question, when most robots are stopped, the coverage area could approach to a limit value. Hence, the base station should release more robots when the percentage of stop robots exceeds a certain value. Assume  $p(k)$  is the percentage of stopped robots at Step  $k$ , and  $P$  is a decision value defined by the user. Then, the releasing time  $T_t$  can be defined as follows,

$$T_t = \text{sign}(p(k) - P) \times k \quad (7)$$

where  $t = \{1, 2, 3, \dots\}$  is the releasing index, and

$$\text{sign}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (8)$$

For the second question, the goal of releasing new robots is to enlarge the coverage area. Thus,  $u_t$ , the number of additional robots at time  $T_t$ , is defined as follows:

$$u_t(k = T_t) = \frac{\Delta A_t}{\pi R_{a,eff}^2} \quad (9)$$

where  $\Delta A_t$  is the desired additional area to be increased at time  $T_t$ . To determine  $\Delta A_t$ , the coverage area  $A^{total}(k)$  is regarded as a circle, and the desired increasing radius  $\Delta R$  is also defined. Then  $\Delta A_t$  is the ring area as shown in Figure 6.

However, without the position information, the total coverage area  $A^{total}(k)$  is unable to compute and consequently  $\Delta A_t$  is unable to determine. Hence, two methods are proposed to estimate  $A^{total}(k)$ , namely, (1) the feedforward estimation, and (2) the feedback estimation.

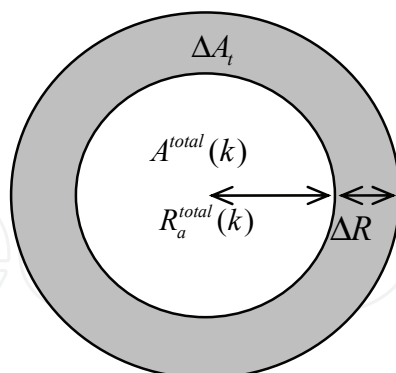


Figure 6. The diagram of  $\Delta A_t$

#### 4.1 Area Estimation with Prior Information

Figure 7 shows the diagram of the feedforward estimation. While the percentage of stopped robots is computed, the total coverage area is estimated by the base station itself. Since most robots are stopped at the releasing times, each of them should have reached a certain coverage area. The effective coverage area of single robot is estimated by utilizing  $R_{a,eff}$  and

the total coverage area is then estimated by summing these single effective coverage areas. Let  $A_{est}^{total}(k)$  be the estimated total coverage area at Step  $k$ , then

$$A_{est}^{total}(k = T_i) = N(k)\pi R_{a,eff}^2 \tag{10}$$

where  $N(k)$  is the number of robots at Step  $k$ . This estimate does not need any information returned by robots and is determined only with former information  $R_{a,eff}$ . Hence it is named as “feedforward estimation.”

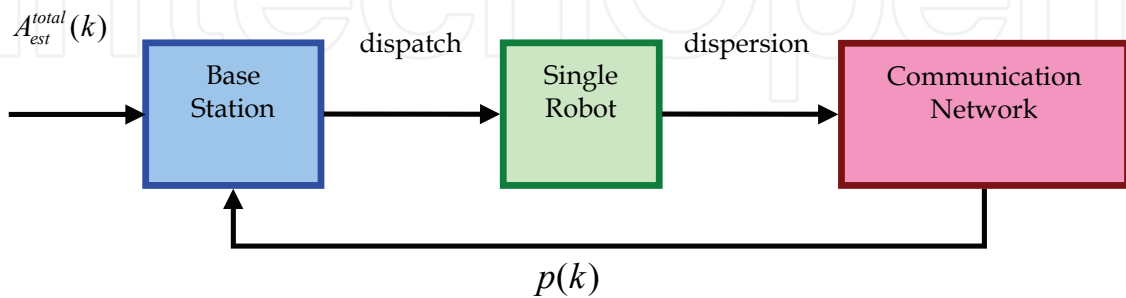


Figure 7. The flowchart of the dispatch-dispersion mechanism with feedforward estimation

#### 4.2 Area Estimation with Feedback Information

Although the actual effective coverage area of each robot is unable to obtain, each robot can still estimate it. Figure 8 shows the diagram of the feedback estimation. Same as the feedforward estimation, the percentage of stopped robots is returned by robots. Also the effective coverage area is estimated by each robot itself and then returned to the base station. Then the total coverage area could be estimated by summing these estimated effective areas. The following estimation is then used:

$$A_{est}^i(k) = \frac{\pi R_c^2}{n_c^i(k) + 1} \tag{11}$$

where  $A_{est}^i(k)$  is the estimated effective coverage area of robot  $i$  at step  $k$ . Therefore the total coverage area is computed as follows:

$$A_{est}^{total}(k) = \sum_{i=1}^{N(k)} A_{est}^i(k) \tag{12}$$

With  $A_{est}^{total}(k)$  substituted for  $A^{total}(k)$ ,  $\Delta A_t$  and  $u_t$  can then be computed.

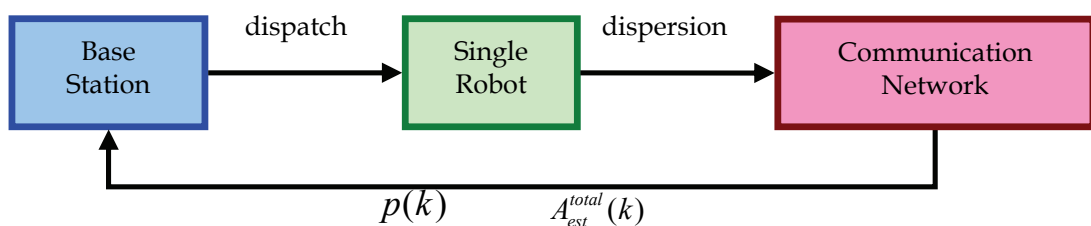


Figure 8. The flowchart of the dispatch-dispersion mechanism with feedback estimation

## 5. Simulation Study

In this section, several simulation studies are conducted to demonstrate the performance of the proposed movement algorithm. First, the stability of the dispersion algorithm with one single robot moving in static environment is studied. Second, the dispersion algorithm used in a multi-robot system is demonstrated and the statistics of partition rate, coverage area, spending time and stop rate are summarized. Finally, the dispatch rule at the base station is combined to execute the scenario of a target search problem.

### 5.1 Static Environment: Dispersion of One Robot

Figure 9 shows the simulation results on a static environment. In this simulation, the following parameters are used.  $R_c = 100$ ,  $R_r = 40$ ,  $N_c = 6$ ,  $T_{normal} = 10$ ,  $T_{strait} = 20$  and  $T_{escape} = 50$ .

The light (purple) dots in Figure 9 denote stationary robots. With these stationary robots, the equilibrium region that only one single moveable robot can satisfy the requirement of  $n_c^i(k) = 6$  and  $n_r^i(k) = 0$  are computed. The equilibrium region is denoted by the dark (red) hexagon symbols as the outer circular region and the inner circular region near the center as shown in Figure 9(a). The two similarly circular regions in Figure 9(b) are the final stop positions of one single moveable robot with the same requirement using 1500 simulations with initial positions set at (100,100) or (-100,-100). From the two plots, it can be observed that the two set of dark (red) regions are almost the same. The reason that the inner part of the hexagon symbols of Figure 9(a) is missing in Figure 9(b) is that the robot stops right as the requirements are satisfied. Hence, no robots stop inside the inner circle. Therefore, the dispersion algorithm indeed leads the robot to the regions that satisfy the requirement.

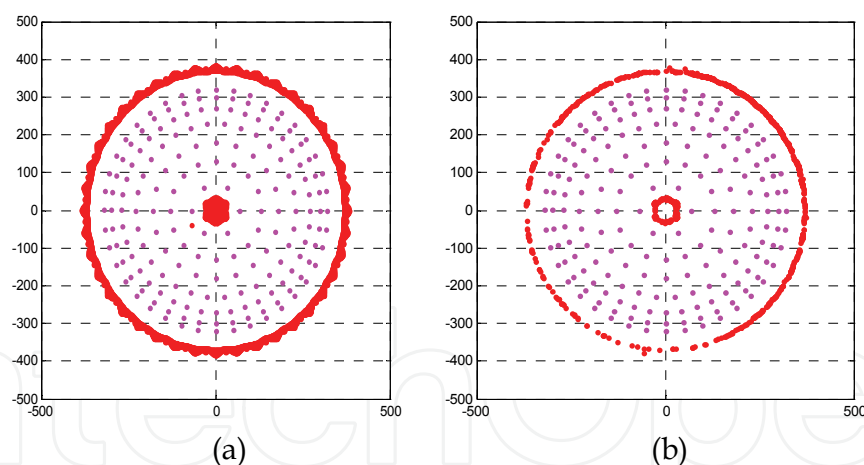


Figure 9. Static environment simulation. (a) The analytic region of equilibrium points. (b) The simulation result of stop region

Figure 10 is another static environment simulation result. All the parameters are the same as the previous example. The dark (red) circular region in Figure 10(a) is the analytic regions of equilibrium points that satisfy the requirements. It is an open region which is slightly different from the regions in Figure 9(a). In Figure 10(b) the dark (red) circular region is the stop positions of one single moveable robot in 1000 simulations. The dark (red) region is almost similar to the one in Figure 10(a). Moreover, although the equilibrium region is open, the robot still stops at the right positions and does not wander to the faraway positions. Hence the algorithm indeed leads the robot to the desired position.

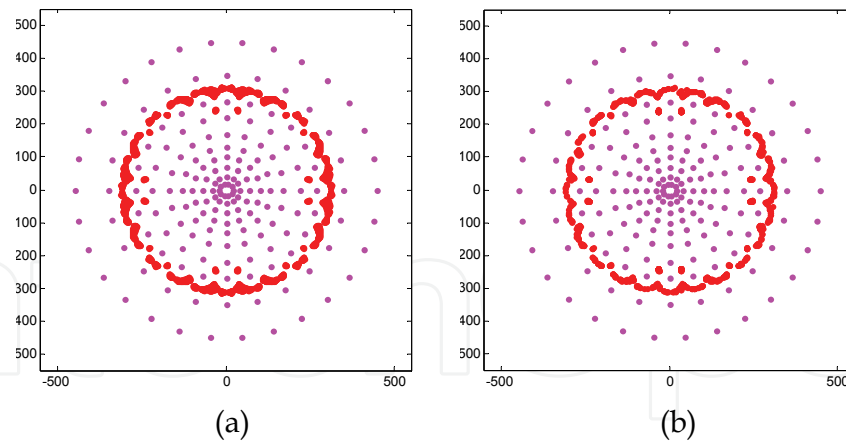


Figure 10. Static environment simulation. (a) The analytic region of equilibrium points. (b) The simulation result of stop region

## 5.2 Multi-Robot System: Dispersion of n Robots

Figure 11 shows the dispersion progress of a group of 60 robots. In this simulation, the following parameters are used:  $N = 60$ ,  $R_c = 100$ ,  $R_r = 30$ ,  $N_c = 6$ ,  $T_{normal} = 10$ ,  $T_{strait} = 20$  and  $T_{escape} = 50$ . The final balanced distribution forms a good communication network. It can be seen that, due to the repulsion force of Phase R, there are no robots staying too close and hence the coverage area has been enlarged to a certain value. Related statistic analysis of the dispersion algorithm is discussed in the following.

### 5.2.1 Partition Rate

The zero desired value of  $n_r^i(k)$  provides a repulsion force to robots. If  $R_r$  is set too large, the network partition is very likely to happen. But with a small  $R_r$ , the repulsion mechanism would not be obvious. It is important to choose an appropriate value of  $R_r$ .

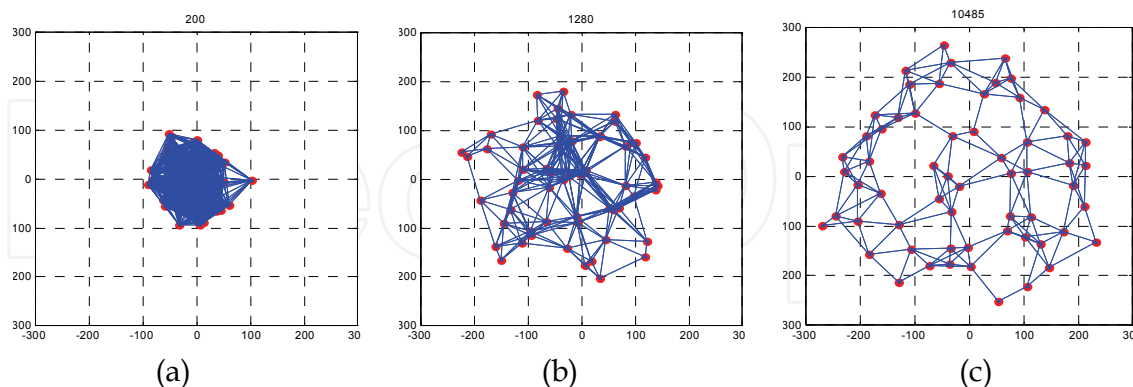


Figure 11. The dispersion of 60 robots. (a) Step 200: The dispersion has just begun, and the robots still stay close. (b) Step 1280: The network has dispersed obviously. (c) Stop 10485: The final result of the dispersion. The robots form a dispersed communication network

Figure 12 shows the relation between the partition rate and  $R_r / R_c$ . From this figure, it can be seen that, when  $R_r$  is below about 0.4 times  $R_c$ , the partition hardly happens. However,

when  $R_r$  is set too large to about 0.6 times  $R_c$ , the probability of partition is almost equal to 100%. Hence, the best value of  $R_r$  is set to be about 0.4 times  $R_c$ .

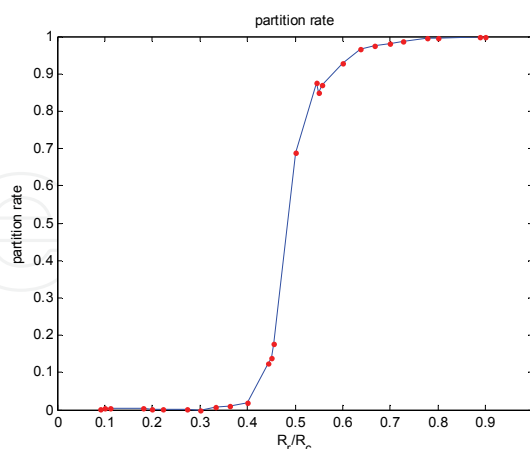


Figure 12. The partition rate

### 5.2.2 Coverage Area/Effective Area Radius

In addition to the partition rate, the value of  $R_r/R_c$  also affects the effective coverage area radius  $R_{a,eff}$ . As shown in Eqn. (6),  $R_r/R_c$  and  $R_{a,eff}/R_c$  should have a linear relationship.

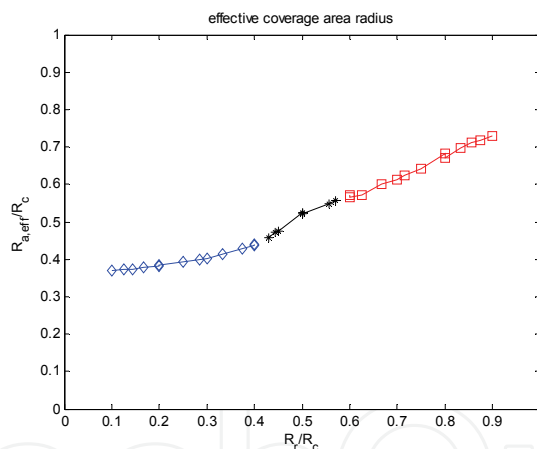


Figure 13. The effective area radius

Figure 13 is a statistical result of 2000 simulations. The final average coverage area of different values of  $R_c$ ,  $R_r$  and  $N$  are computed. The slope of the average area and  $N$  with fixed  $R_c$  and  $R_r$  is considered as the effective coverage area of a single robot, and then  $R_{a,eff}$  is derived. In Figure 13, it can be seen that, when  $R_r/R_c$  is below about 0.4,  $R_r/R_c$  and  $R_{a,eff}/R_c$  indeed have a perfect linear relationship. Moreover, they have another linear relationship when  $R_r/R_c$  exceeds about 0.6. The two regions of  $R_r/R_c$  that  $R_r/R_c$  and  $R_{a,eff}/R_c$  have good linear relationships are just the regions that the partition rate is almost equal to 0 or 100% as shown in Figure 9. This indicates that the linear relationship of  $R_c$ ,  $R_r$  and  $R_{a,eff}$  exists for both that the communication network is not partitioned, and that the



communication network is completely partitioned. In this case,  $k_r$  is 0.228 and  $k_c$  is 0.340 when  $R_r/R_c$  is below 0.4. These values indicate that the weighting of attraction is larger than that of repulsion since the network is pulled together by the attraction. When  $R_r/R_c$  is above 0.6,  $k_r$  is 0.563 and  $k_c$  is 0.225. These values indicate that the weighting of repulsion is larger than that of attraction since the network is partitioned due to the repulsion force.

### 5.2.3 Spending Time

The spending time is defined as the time when all robots are stopped. Figure 14 shows the relationship of the spending time and the number of robots with  $R_c = 100$  and  $R_r = 10, \dots, 40$ . From the figure, it can be observed that the spending time is roughly proportional to the number of robots. Moreover, the slope is only determined by  $R_r$ . Figure 15 shows the value of slopes with different  $R_r$  and  $R_c$ . It can be seen that even with different  $R_c$ , the slopes of the spending time and the number of robots are almost the same as long as  $R_r$  is the same. This statistics indicates that the time is mainly spent on repulsing too-close neighbors.

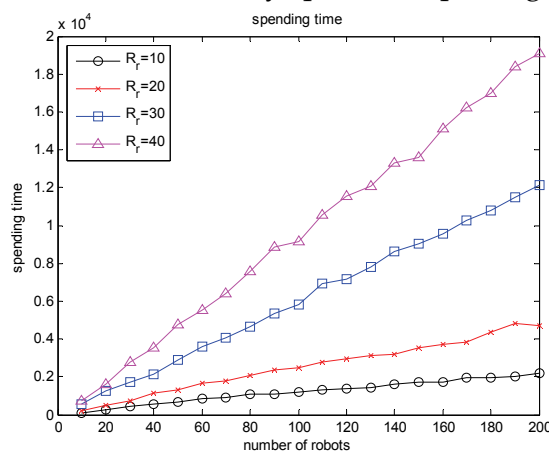


Figure 14. The spending time. With fixed  $R_r$  and  $R_c$  the spending time is roughly proportional to the number of robots

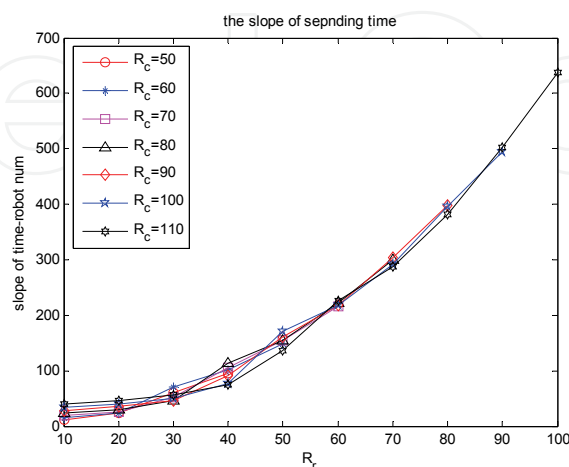


Figure 15. The slope of spending time and number of robots which is determined only by  $R_r$

### 5.2.4 Stop Rate

The average spending times are also recorded when the stop rate exceeds 10%, 20%, ..., 100%. The result is shown in Figure 16. The vertical axis is the percentage of the time spent, and the horizontal axis is the stop rate. It can be seen that these two ratios have a roughly exponential relationship regardless of the values of  $R_r$  and  $R_c$ . This combined with the spending time would be useful information for estimating the time when the stop rate exceeds a certain value. Later this information is used in the dispatch rule.

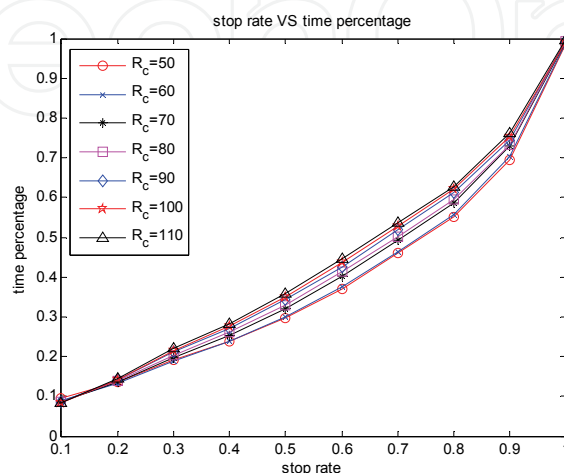


Figure 16. Stop rate and time percentage. They have a rough exponential relationship regardless of  $R_r$  and  $R_c$ .

### 5.3 Target Search: Dispersion and Dispatch

In this subsection, the dispatch rule is combined with the dispersion algorithm to execute the target search problem. Figure 14 shows the progress of a target search problem utilizing the feedforward estimation dispatch. The initial number of robots is 10.  $R_c$ ,  $R_r$ ,  $N_c$ ,  $T_{normal}$ ,  $T_{strait}$  and  $T_{escape}$  are set as the values in previous subsection.  $\Delta R = 50$ ,  $P = 0.8$  and the target position is at (200, 200).  $k_r$  and  $k_c$  are set as 0.228 and 0.340, respectively.

New robots are released from the center of the plane when  $p(t)$  exceeds  $P$ , as shown in Figure 17(b), (d), and (f). The target is found after the three dispatches as shown in Figure 17(g).

The objective of the dispatch at the base station is to enlarge the coverage area timely. Figure 18 shows the coverage area versus the time of two simulations. The longer (red) line is a simulation result of feedforward estimation, and the shorter (blue) line is a simulation result of feedback estimation. It can be seen that the coverage area increases almost linearly with time, which indicates that by using the dispatch rule the base indeed releases appropriate number of robots at right time.

Moreover, the final spending time when the stop rate  $p(t)$  reaches  $P$  can be estimated in advance with the simulation statistics. Hence another dispatch rule is studied where the coverage area and the stop rate of robots are both estimated by the base station. The flowchart is shown in Figure 19. Figure 20 is a simulation result of applying the statistics of the spending time and the stop rate. It can be seen that the coverage area increases roughly

linear with time as well as the two former estimation methods. Hence this estimation provides a good performance. Moreover, no information returned by robots is needed, therefore the base station can even determine  $u_k$  and  $T_i$  before the task starts. This is a very beneficial advantage.

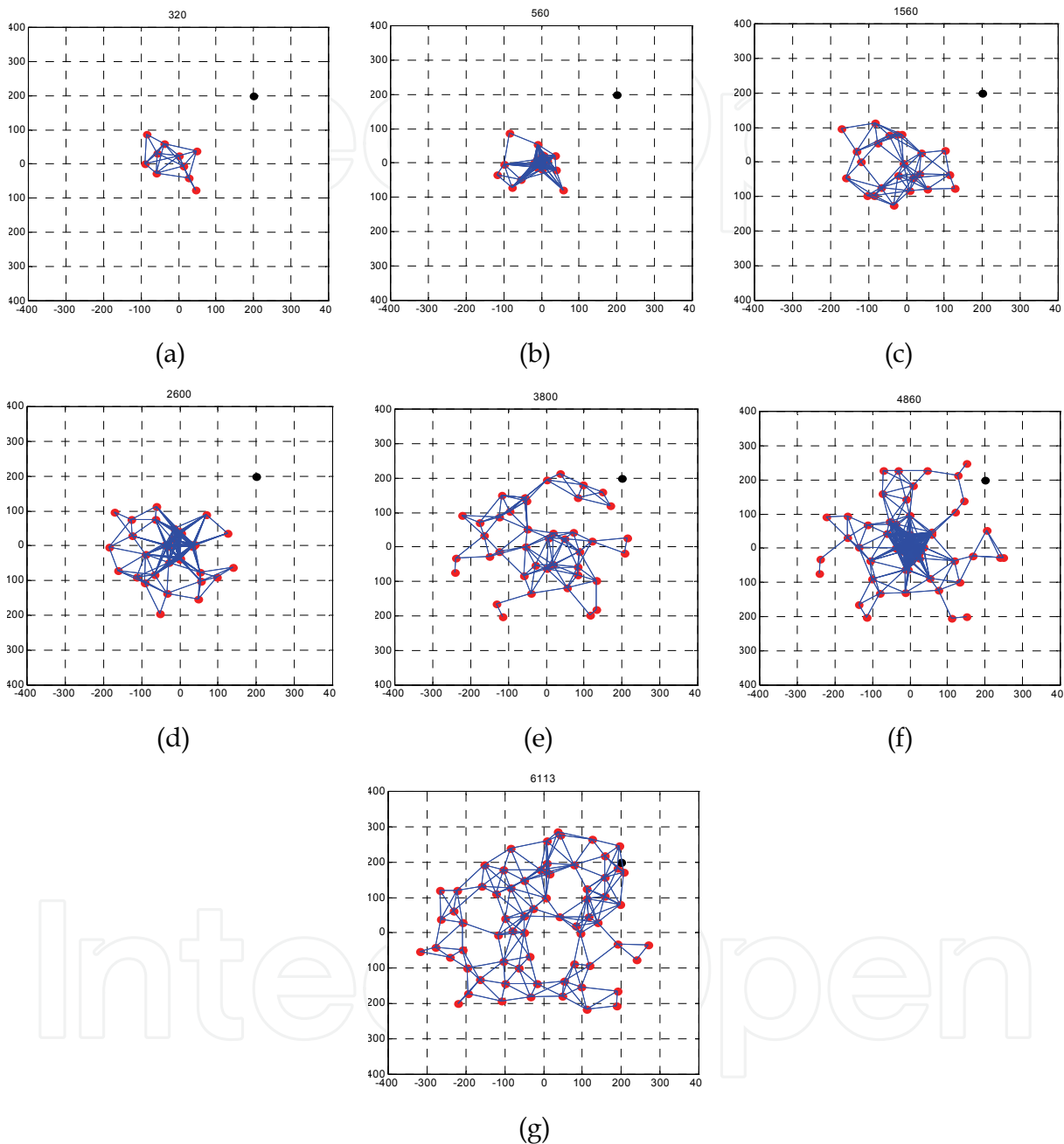


Figure 17. The progress of target search. (a) Step 320: Initially, the network formed by 10 robots which is too small to cover the target. (b) Step 560: After the stop rate reaches 0.8, new robots are released. (c) Step 1560: The network is enlarged but still unable to cover the target. (d) Step 2600: The second releasing. (e) Step 3800: The network is enlarged again. (f) Step 4860: The third releasing. (f) Step 6113: After 3 times of releasing, the target is successfully found by the enlarged network

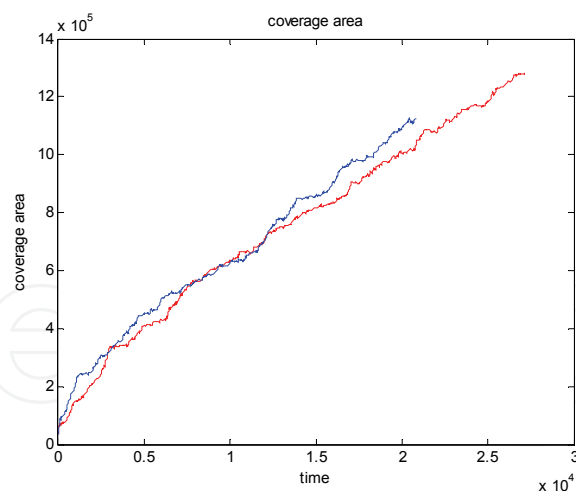


Figure 18. Coverage area versus time. The coverage area increases roughly linear with time

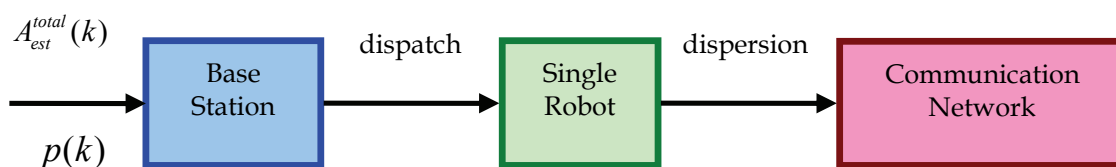


Figure 19. Flowchart of the dispatch rule utilizing area and stop percentage estimation with prior information

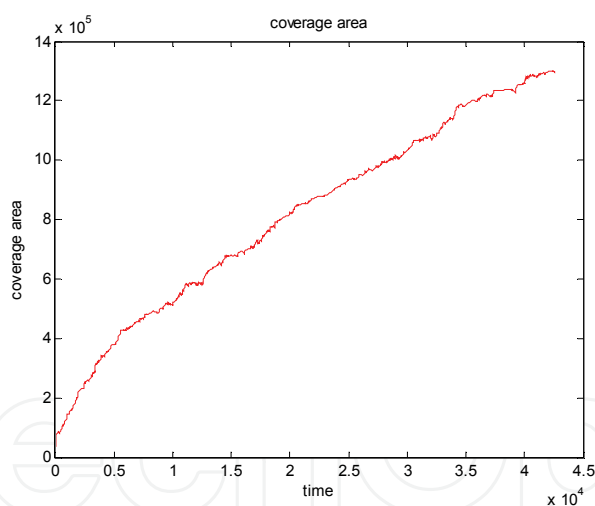


Figure 20. Coverage area versus time

### 5.3 Estimation Accuracy of Dispatch Rule

In the target search task, the base station decides the releasing time and the number of released robots by estimating the coverage area and the stop percentage of robots with three kinds of methods, namely the “area estimation with prior information”, “area estimation with feedback information” and “area and stop percentage estimation with prior information.” For these three estimations, the accuracy is one of major concerns. Figure 21 and Figure 22 show the estimation accuracy of area and stop percentage of the three

methods. The following parameters are used:  $R_c = 100$ ,  $R_r = 40$ ,  $N_c = 6$ ,  $T_{normal} = 10$ ,  $T_{strait} = 20$  and  $T_{escape} = 50$ , and  $P$  is set as 0.8 and the initial number of robots is set to 10. Robots are released from the origin of the plane, and the target is set at (350, 350). 50 simulations for each estimation method are done to compute the average values. The statistics of spending time, stop percentage and coverage area presented in Section 5.2 are used in the dispatching. Figure 21(a) and Figure 21(b) shows the ratio of the estimated area and the real area at each releasing time with "area estimation with prior information" and "area estimation with feedback information," respectively. From the figures it can be observed that the estimation accuracy becomes better as the time increases. The two results of estimation accuracy are similar to each other, but the one with prior information has better accuracy in the beginning. Moreover, it remains a value between 0.9 and 1 in the later period while the other one may exceed 1.

Figure 21(c) and Figure 22 show the estimation accuracy of "area and stop percentage estimate with prior information." Compared with Figure 21(a) and Figure 21(b), Figure 21(c) shows that the accuracy of area estimation is a little worse than the one of the former two estimations but still remain a value larger than 0.5. And for the estimation of stop percentage, the estimation value is between 0.6 and 0.95, which is about the range of  $0.8 \pm 0.2$ . Hence the estimation of stop percentage shows a good result.

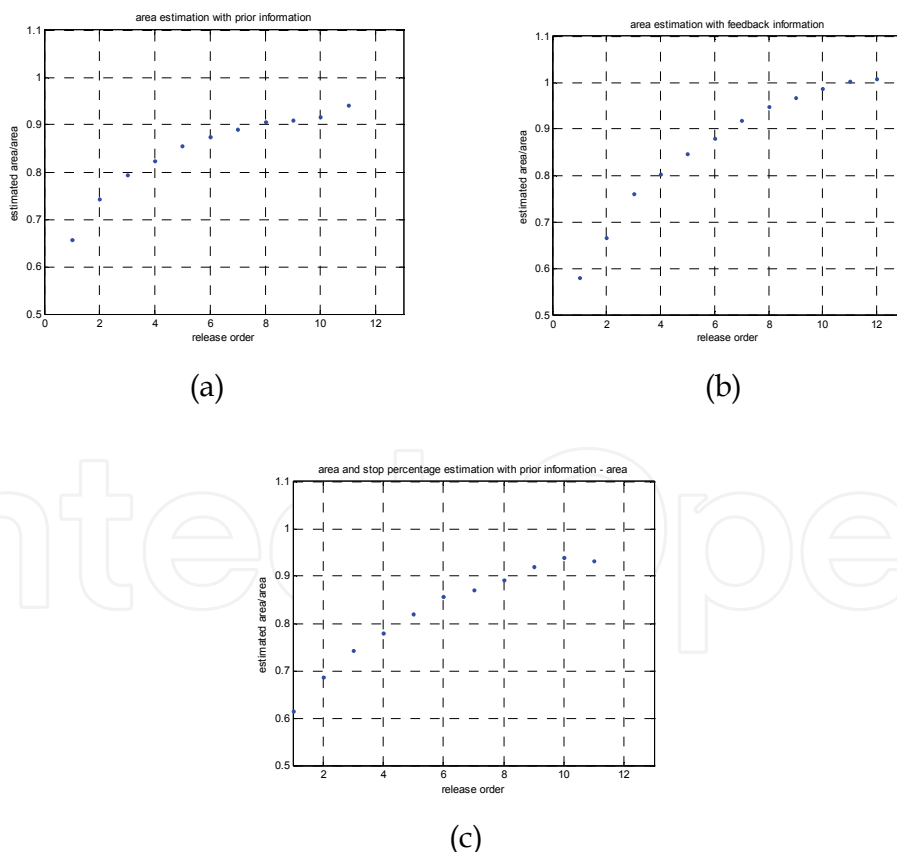


Figure 21. Ratio of estimated area and real area at releasing times with (a) area estimation with prior information; (b) area estimation with feedback information; (c) area and stop percentage estimation with prior information

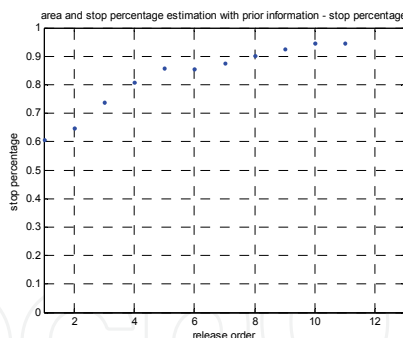


Figure 22. Estimated stop percentage at releasing times with “Area and stop percentage estimation with prior information”

## 6. Conclusions and Future Work

In this chapter, a dispersion movement algorithm for multi-robot systems with simple computation and easily obtainable information is proposed. The only information needed is the communication density, i.e., the number of communication links of each individual robot. In addition, a dispatch control rule is proposed based on the dispersion algorithm. With some parameters known in advance, the base station could then estimate an appropriate time to release new robots. The dispersion and dispatch control rules are easy to implement for a practical multi-robot system to act like a natural creature system. The dispersion movement algorithm itself still executes as a natural system. And with the dispatch rule, the dispersion algorithm can be used in tasks with more variety. Simulation results of the dispersion and dispatch control rules are presented, and statistics of the coverage area, partition rate, spending time and stop rate show the advantage of these algorithms. In the future, the research will focus on the mechanism of adaptively adjusting of the dispatch control rule, and the theoretical analysis of the algorithm performance. The implementation of the algorithm on practical robots and further applications are also under planning.

## 7. Acknowledgement

This work was supported in part by the National Science Council, Taiwan, ROC, under the grants: NSC 95-2221-E-002-303-MY3, and NSC 96-2218-E-002-030, and by DOIT/TDPA: 95-EC-17-A-04-S1-054.

## 8. References

- Blough, D.M.; Leoncini, M.; Resta, G. & Santi, P. (2003) The K-Neigh Protocol for Symmetric Topology Control in Ad Hoc Networks, *Proceedings of ACM MobiHoc 2003*, pp. 141-152, ISBN: 1-58113-684-6, Jun. 2003, Association for Computing Machinery, Annapolis, MD, USA
- Burgard, W.; Moors, M.; Stachniss, C. & Schneider, F.E. (2005) Coordinated Multi-Agent Exploration, *IEEE Transactions on Robotics*, Vol. 21, No. 3, (Jun. 2005) pp. 376-386, ISSN: 1552-3098

- Chellappan, S.; Bai, X.; Ma, B.; Xuan, D. & Xu, C. (2007) Mobility Limited Flip-Based Sensor Networks Deployment, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 2, (Feb. 2007) pp. 199-211, ISSN: 1045-9219
- Cortes, J.; Martinez, S.; Karatas, T. & Bullo, F. (2004) Coverage Control for Mobile Sensing Networks, *IEEE Transactions Robotics and Automation*, Vol. 20, No. 2, (Apr. 2004) pp. 243-255, ISSN: 1042-296X
- Czirok, A.; Ben-Jacob, E.; Cohen, I. & Vicsek, T. (1996) Formation of complex bacterial colonies via self-generated vortices, *Physical Review E*, Vol. 54, No. 2, (Aug. 1996) pp. 1971-1801, ISSN: 1063-651X
- Flierl, G.; Grunbaum, D.; Levin, S. & Olson, D. (1999) From Individuals to Aggregations: the Interplay between Behavior and Physics, *Journal of Theoretical Biology*, Vol. 196, No. 4, (Feb. 1999) pp. 397-525, ISSN: 9911-4082
- Gazi, V. & Passino, K.M. (2003) Stability Analysis of Swarms, *IEEE Transactions on Automatic Control*, Vol. 48, No. 4, (Apr. 2003) pp. 692-697, ISSN: 0018-9286
- Gazi, V. & Passino, K.M. (2004) A class of attractions/repulsion functions for stable swarm aggregations, *International Journal of Control*, Vol. 77, No. 18, (Dec. 2004) pp. 1567-1579, ISBN: ISSN: 0020-7179
- Gueron, S.; Levin, S.A. & Rubenstein, D. I. (1996) The Dynamics of Herds: From Individuals to Aggregations, *Journal of Theoretical Biology*, Vol. 182, No. 1, (Sep. 1996) pp. 85-98, ISSN: 9911-4082
- Heo, N. & Varshney, P.K. (2005) Energy-Efficient Deployment of Intelligent Mobile Sensor Networks, *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, Vol. 35, No. 1, (Jan. 2005) pp. 78-92, ISSN: 1083-4427
- Li, L.; Halpern, J.Y.; Bahl, P.; Wang, Y. & Wattenhofer, R. (2005) A Cone-Based Distributed Topology Control Algorithm for Wireless Multi-Hop Networks, *IEEE/ACM Transactions on Networking*, Vol. 13, No. 1, (Feb. 2005) pp. 147-159, ISSN: 1063-6692
- Parrish, J.K.; Viscido, S.V. & Grunbaum, D. (2002) Self-Organized Fish Schools: An Extraction of Emergent Properties, *The Biological Bulletin*, Vol. 202, No. 3, (June 2002) pp. 296-305, ISSN: 0006-3185
- Passino, K.M. (2002) Biomimicry of Bacterial Foraging for Distributed Optimization and Control, *IEEE Control Systems Magazine*, Vol.22, No.3, (June 2002) pp. 52-67, ISSN: 0272-1708
- Poduri, S. and Sukhatme, G.S. (2004) Constrained Coverage for Mobile Sensor Networks, *Proceedings of IEEE 2004 International Conference on Robotics and Automation*, pp. 165-171, ISBN: 0-7803-8232-3, Apr. 2004, Institute of Electrical and Electronics Engineers, New Orleans, LA, USA
- Rodoplu, V. & Meng, T.H. (1999) Minimum Energy Mobile Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, (Aug. 1999) pp. 1333-1344, ISSN: 0733-8716
- Santi, P. (2005) *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, ISBN: 978-0470094532, Hoboken, NJ, USA
- Tan, J. (2005) A Scalable Graph Model and Coordination Algorithms for Multi-agent Systems, *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1529-1534, ISBN: 0-7803-9046-6, Jul. 2005, Institute of Electrical and Electronics Engineers, Monterey, CA, USA



## **Recent Advances in Multi Robot Systems**

Edited by Aleksandar Lazinica

ISBN 978-3-902613-24-0

Hard cover, 326 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, May, 2008

**Published in print edition** May, 2008

To design a team of robots which is able to perform given tasks is a great concern of many members of robotics community. There are many problems left to be solved in order to have the fully functional robot team. Robotics community is trying hard to solve such problems (navigation, task allocation, communication, adaptation, control, ...). This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field. It is focused on the challenging issues of team architectures, vehicle learning and adaptation, heterogeneous group control and cooperation, task selection, dynamic autonomy, mixed initiative, and human and robot team interaction. The book consists of 16 chapters introducing both basic research and advanced developments. Topics covered include kinematics, dynamic analysis, accuracy, optimization design, modelling, simulation and control of multi robot systems.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Feng-Li Lian, You-Ling Jian and Wei-Hao Hsu (2008). Dispersion and Dispatch Movement Design for a Multi-Robot Searching Team Using Communication Density, Recent Advances in Multi Robot Systems, Aleksandar Lazinica (Ed.), ISBN: 978-3-902613-24-0, InTech, Available from:

[http://www.intechopen.com/books/recent\\_advances\\_in\\_multi\\_robot\\_systems/dispersion\\_and\\_dispatch\\_movement\\_design\\_for\\_a\\_multi-robot\\_searching\\_team\\_using\\_communication\\_density](http://www.intechopen.com/books/recent_advances_in_multi_robot_systems/dispersion_and_dispatch_movement_design_for_a_multi-robot_searching_team_using_communication_density)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen