Electronic Preprint for Journal of Information Processing Vol.29

Regular Paper

Dynamic Swarm Spatial Scaling for Mobile Sensing Cluster in a Noisy Environment

Eiji Nii[1] Shoma Nishigami[2] Takamasa Kitanouma[3] Hiroyuki Yomo[4] Yasuhisa Takizawa[2]

Received: March 30, 2020, Accepted: November 5, 2020

Abstract: Autonomous mobile devices, such as robots and unmanned aerial vehicles, as alternatives to humans, are expected to be applied to searching for and manipulating a variety of emergent events of which the location and number of occurrences are unknown. When an autonomous mobile device searches for an event, it needs to sense a physical signal emitted by an event, such as radio waves, smell or temperature. After a device finds an event, it must manipulate the event. We previously proposed Mobile Sensing Cluster (MSC), which applies swarm intelligence to multiple autonomous mobile devices to quickly search for and manipulate multiple events using dynamically formed multiple swarms of mobile devices. However, in an environment that the physical signal emitted by an event and sensed by a device includes some random noises, the behavior of swarms in MSC becomes unstable. As a result, MSC requires a long time to search and manipulate. In this paper, we propose a dynamic swarm spatial scaling MSC for improving the tolerance of MSC against such random noises, and show its effectiveness.

Keywords: wireless sensor networks, particle swarm optimization, autonomous mobile devices

1. Introduction

In the near future, as alternatives to humans, autonomous mobile devices, such as robots and unmanned aerial vehicles, are expected to be applied to searching for and manipulating a variety of emergent events of which the location and number of occurrences are unknown [1], [2]. When an autonomous mobile device searches for such an event, it needs to sense a physical signal emitted by an event, such as radio waves, smell or temperature. In this paper, we focus on radio waves as the physical signal. After a device finds an event, it must manipulate the event. The event is defined as generalizing diverse phenomena, such as an outbreak of damage on buildings and infrastructures, an outbreak of survivors to rescue in a disaster, etc. The manipulation is, for example, repairing the damage to buildings and rescuing the survivors. Because of the nature of an event, the device is required to search for and manipulate a greater large number of events in less time, however, it is difficult for a single device to realize the requirement because of its restrictions such as sensing performance, manipulating performance, battery power, mobile speed, etc.

We previously proposed Mobile Sensing Cluster (MSC) [3] to address the above issues. In MSC, multiple devices share information through wireless communication between them and create a swarm to search for an event and manipulate the event by applying particle swarm optimization (PSO) [4] to the devices. MSC also extends PSO to form dynamic multiple swarming.

In MSC, the strength of a physical signal emitted by an event and sensed by a device is assumed to monotonically increase according to how near it approaches an event. However, in a real environment, the physical signal sensed by a device includes some random noises caused by an obstacle or interference and its strength does not always monotonically increase as a device approaches an event. In such environments, the swarms in MSC spend a long time to searching for and manipulating multiple unknown events because they are moving in the incorrect direction to an event because of random noises they are sensing.

In this paper, we propose a dynamic swarm spatial scaling for MSC in such noisy environments in which the strength of a physical signal includes random noises. The proposed method increases the spatial scale of the swarm in the search phase to improve the tolerance of MSC against random noises, decreases the spatial scale as the swarm approaches the event, then decreases the time of searching for and manipulating an event that emits a physical signal with random noises.

The rest of this paper is organized as follows. Section 2 presents related works, Section 3 explains MSC, Section 4 describes the proposed method, and Section 5 shows evaluation with simulation. Finally, Section 6 draws conclusions.

2. Related Works

2.1 Swarm Robotics

Swarm robotics [5], [6], [7] is a new approach for coordinating multi-robot systems consisting of large numbers of mostly simple physical robots. This approach is inspired by nature and is a combination of swarm intelligence and robotics. The individuals in the swarm are normally simple, small, and inexpensive. A key component of this approach is the communication between

[1] Graduate School of Science and Engineering, Kansai University, Suita, Osaka 564–8680, Japan
[2] Faculty of Environmental and Urban Engineering, Kansai University, Suita, Osaka 564–8680, Japan
[3] Organization for Research and Development of Innovative Science and Technology Kansai University, Suita, Osaka 564–8680, Japan
[4] Faculty of Engineering Science, Kansai University, Suita, Osaka 564–8680, Japan

the individuals in a swarm, which is normally local, allowing a multi-robot system to be scalable and robust.

2.2 Reynolds Flocking Model

The Reynolds flocking model [8], which simply simulates the swarming behavior of flocks of birds, was introduced in 1987. Each agent moves based on the following three rules [9], [10]:

- alignment: agents adjust their velocity to that of their neighbor agents;
- cohesion: agents are attracted to the average position of their neighboring agents; and
- collision avoidance: agents are repulsed from their neighboring agents.

The Reynolds flocking model has no function to search for an event because its algorithm maintains a swarm's form, which is organized by multiple agents. It also takes into account the formation for a single swarm.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [11], [12], inspired by the swarm behavior of flocks of birds and schools of fish, is a mathematical search model based on multiple particles. Each particle has a location and velocity, and its location is evaluated using a fitness function [13], [14], [15]. The velocity of each particle is derived from its personal and global bests. The former is the best previous location of the particle, and the latter is the best previous location of all particles.

2.4 Consensus Problem

Multiple agent systems, which cooperatively control arbitrary systems by multiple agents, are expected to be used in the field of sensor networks or to control autonomous robots. In such systems, the velocity of robots and the sensing data values converge to an arbitrary value called a consensus problem [16], [17]. Only obtaining a consensus among multiple agents, the systems address the formation of a single swarm [18].

3. Mobile Sensing Cluster

Most works in section 2 focus on the formation of single swarm, but do not investigate the division of a swarm to form multiple swarms. Therefore, they cannot search for and manipulate multiple unknown events in parrallel by forming multiple swarms.

In this section, we explain the original MSC [3] that can search for and manipulate multiple unknown events in parallel by forming multiple swarms, and that can search for and manipulate more events in a shorter time. The original MSC is composed of two mechanisms as follows.

- A search and manipulation mechanism based on PSO of unknown events using wireless communication to enable interaction between mobile devices
- A dynamic multiple-swarming mechanism that extends PSO to create the behavior of multiple swarms.

The search and manipulation mechanism is based on PSO using wireless communication to create the intelligent swarm's behavior that emerges from the collective behavior of a large number of autonomous mobile devices. The dynamic multiple-swarming mechanism divides a swarm into multiple swarms to search for and manipulate multiple events in parallel.

3.1 Precondition with MSC

An autonomous mobile device can estimate its position, and share information by wireless communication among multiple mobile devices. An event emits its physical signal which contain its identification.

If the device detects the strength of a physical signal above a threshold, it determines that it reaches an event and starts to manipulate the event. Each device can manipulate an event independently and in parallel.

3.2 Search and Manipulation Mechanism

3.2.1 Location Updating Rule

To search for and manipulate unknown events in the real world, this mechanism is operated in each mobile device to derive a location to move toward based on the following updating rule:

$$\begin{aligned} v_i(t+1) = {} & wv_i(t) + p_i(t)(x_i^{Pbest}(t) - x_i(t)) \\ & + l_i(t)(x^{Lbest}(t) - x_i(t)) + \vec{S_i} \end{aligned} \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{2}$$

where t is time, $v_i(t)$ is the velocity of device i at time t, w is the weight of the inertia vector $v_i(t)$ at time $t+1$, $p_i(t)$ is the weight of the personal best, $l_i(t)$ is the weight of the local best, x_i^{Pbest} is the personal best location, $x_i^{Lbest}(t)$ is the best location of neighbor devices, and $\vec{S_i}$ is the collision-avoidance vector of device i.

3.2.2 Personal Best and Local Best Locations

The personal best location is where each mobile device senses the physical signal strength from events, based on the personal best evaluation value, that shows the distance from an event:

- If the personal best evaluation value improves, a device randomly updates the velocity vector around the current moving direction.
- Otherwise, a device randomly updates the velocity vector around the opposite direction to the current moving direction:

$$x_i^{Pbest}(t) = \begin{cases} |v_i(t-1)|(\cos(\alpha+\beta), \sin(\alpha+\beta)) + x_i(t) \\ \quad \text{if } E_i^{Pbest}(t-1) > E_i^{Pbest}(t) \\ -|v_i(t-1)|(\cos(\alpha+\beta), \sin(\alpha+\beta)) + x_i(t) \\ \quad \text{otherwise.} \end{cases} \tag{3}$$

Here $E_i^{Pbest}(t)$ is the personal best evaluation value of device i at time t, α is an angle of $v_i(t-1)$ with x axis, β is a random angle in $[-\theta, \theta]$ and θ is a parameter defining the random number space for β.

The local best location is a site where a neighbor device is nearest to the events in the wireless communication range. The indirect distance to the nearest event from the neighbor devices uses the local best evaluation value.

3.2.3 Evaluation Value

The above updating rule uses the following three evaluation values:

- The personal best evaluation value shows the distance from the nearest event in the discovery and sensing neighboring events. This value is derived as follows:

$$E_i^{Pbest}(t) = \min_{K \in discovery_i(t)} \{E_i^K(t)\}, \tag{4}$$

where $discovery_i(t)$ is a set of discovered events by device i at t and $E_i^k(t)$ is an evaluation value showing the distance from event K based on sensing the physical-signal strength of K in device i at t.

- The local best evaluation value shows the minimum distance to an event in the neighbor devices and derived by being based on the self-evaluation value, which shows the distance to an event in each device:

$$E_i^{Lbest}(t) = \min_{j \in neighbor_i(t)} \{E_j(t)\}, \tag{5}$$

where $E_i^{Lbest}(t)$ is the local best evaluation value of device i at t, $neighbor_i(t)$ is a set of devices whose neighbor devices of device i are found at t, and $E_j(t)$ is a self-evaluation value of device j at t.

- A self-evaluation value shows the distance to an event. If the personal best evaluation value is less than the personal best evaluation values of the neighbors in the wireless communication range, the self-evaluation value is the personal best evaluation value; otherwise, it is the sum of the local best evaluation value and the distance to the local best location and is derived as follows:

$$E_i(t) = \begin{cases} E_i^{Pbest}(t) \\ \quad \text{if } E_i^{Pbest}(t) < \min_{j \in neighbor_i(t)} \{E_j^{Pbest}(t)\} \\ E_i^{Lbest} + C_i^{Lbest}(t) \\ \quad \text{otherwise.} \end{cases} \tag{6}$$

Here $E_i(t)$ is the self-evaluation value of device i at t and $C_i^{Lbest}(t)$ is the distance to the local best location of device i at t.

3.2.4 Selecting Leader

MSC chooses a device that has a minimum personal best value for an event as the leader of a swarm. The leader only moves based on the personal best value, and devices other than the leader (called followers) just move based on the local best value; that is, the leader selfishly moves to an event and the followers obey the leader to search for an event. To produce the above behavior in the swarm, the weights of the personal and local best values are derived as

$$p_i(t) = \begin{cases} 1 & \text{if } E_i^{Pbest}(t) < \min_{j \in neighbor_i(t)} \{E_j^{Pbest}(t)\} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

$$l_i(t) = \begin{cases} 0 & \text{if } E_i^{Pbest}(t) < \min_{j \in neighbor_i(t)} \{E_j^{Pbest}(t)\} \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

3.2.5 Collision-avoidance Control

MSC extends the collision avoidance in the Reynolds flocking model. All devices have collision-avoidance vectors that repulse other devices. A collision-avoidance vector is derived from the distance between it and other devices. The vector, which becomes a strong repulsion vector as a device moves closer to its neighbor, is derived as

$$\vec{S_i} = c_3^i \sum_{j \in n} \frac{\overrightarrow{V_{ji}(t)}}{|V_{ji}(t)|(d_{ij}(t))^k}, \tag{9}$$

where c_3^i is the avoidance weight of device i, $\overrightarrow{V_{ji}(t)}$ is the velocity vector to device i from device j, n is the neighbor devices of device i, d_{ij} is the distance between devices i and j, and k is the avoidance degree.

3.2.6 Search and Manipulation Phases

MSC repeatedly turns between the search and manipulation phases. In the former, as described above, devices search for events by communicating with other neighbor devices based on Eqs. (1) and (2). If the device senses the strength of the physical signal above a threshold, it determines that it has reached an event and enters the manipulation phase.

To stay within a range where the physical signal is strong above a threshold, the device decelerates and adjusts the distance among its neighbors to evenly disperse them. The velocity vector in Eq. (1) and collision-avoidance weight in Eq. (9) are derived as

$$c_3^i = \begin{cases} c_3^{Search} & \text{if } E_i > T \\ c_3^{Search}/n & \text{otherwise.} \end{cases} \tag{10}$$

$$v_i(t) = \begin{cases} \dfrac{v_i(t)}{|v_i(t)|} M^{upper} & \text{if } |v_i(t)| > M^{upper} \\ v_i(t) & \text{otherwise,} \end{cases} \tag{11}$$

where c_3^{Search} is the separation weight in the search phase, n is an integer n, T is a threshold entering the manipulation phase, and M^{upper} is the upper limit of velocity per second.

In the manipulation phase, if a device becomes unable to sense the physical signal from an event within certain a period, it determines that the manipulating of an event is completed. Then to search for other events, it discards the current evaluation values and returns to the search phase.

3.2.7 Wireless Communication among Multiple Mobile Devices

MSC uses wireless communication for sharing information among devices, which advertise the following information and share it among neighboring devices:

- self-location;
- personal best evaluation value;
- self-evaluation value.

The devices that received the above information use it to update their locations and best evaluation values and the leader's selection.

3.3 Dynamic Multiple-swarming Mechanism

MSC dynamically forms multiple swarms to search for and manipulate multiple events in parallel. To manifest the above behavior, it introduces an event-crowd degree for deriving the personal best value and a neighbor-crowd degree for deriving the local best value, divides a swarm into multiple swarms, and controls the number of devices that form a sub-swarm within each swarm.

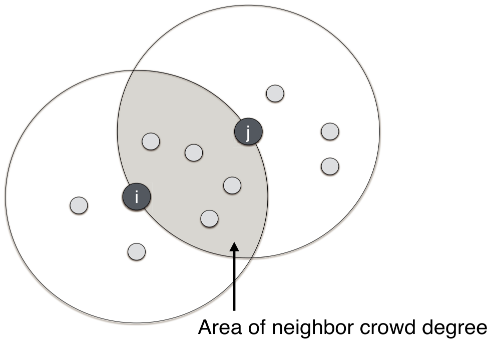


Fig. 1 Area of neighbor-crowd degree.

3.3.1 Multiple Leaders for Dividing a Swarm into Multiple Swarms

As described above, only one device is selected as a leader in a swarm. The dynamic multiple-swarming mechanism selects multiple leaders to search for and manipulate multiple events. To divide a swarm into multiple swarms based on multiple events, the weights of the personal and local best values are revised:

$$p_i(t) = \begin{cases} 1 & \text{if } E_i^{Pbest(K)}(t) < \min_{j \in neighbor_i(t)}\{E_j^{Pbest(K)}(t)\} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

$$l_i(t) = \begin{cases} 0 & \text{if } E_i^{Pbest(K)}(t) < \min\limits_{j \in neighbor_i(t)}\{E_j^{Pbest(K)}(t)\} \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

Here, $E_i^{Pbest(K)}(t)$ is a personal best evaluation value of device i for K at t.

The event-crowd degree is also introduced to derive a personal best value to control the number of devices in a swarm. The event-crowd degree for K accords with the number of neighboring devices in a swarm that is approaching K. By applying the event-crowd degree to the personal best value, since another new leader can be selected to search for other events in a swarm, a swarm is divided into multiple swarms. The event-crowd degree and personal best evaluation value that apply that degree are derived as

$$D_i^k(t) = \{x | x \in neighbor_i(t), P^k(x,t)\} \tag{14}$$

$$E_i^{Pbest(K)}(t) = \min_{k \in discovery_i(t)}\{E_i^{Pbest(k)}(t) + c_4|D_i^K(t)|\}, \tag{15}$$

where $P^K(x,t)$ is a set of devices approaching K at t, $D_i^K(t)$ is a set of the event-crowd degrees for K of device i, and c_4 is a coefficient of the event-crowd degree.

3.3.2 Impartial Swarm Size among Multiple Swarms

To optimize the search and manipulation mechanism based on multi-swarms, the swarm size, which is the number of devices that form a swarm, must be impartial among multiple swarms, To make the number of followers uniform among multiple swarms, we applied the neighbor-crowd degree to derive the local best evaluation value. The neighbor-crowd degree accords with the number of devices between a device and its neighboring devices, as shown in **Fig. 1**. If the neighbor -crowd degree for a neighbor is large, that is, the swarm among the neighbors is crowded, it follows another device with a lower neighbor-crowd degree. To enable the above behavior, we derive the local best evaluation value with the neighbor-crowd degree:

$$N_i^j(t) = \{x | x \in neighbor_i(t), x \in neighbor_j(t)\} \tag{16}$$

$$E_i^{Lbest}(t) = \min_{j \in neighbor_i(t)}\{E_j(t) + c_4|N_i^j(t)|\}, \tag{17}$$

where $N_i^j(t)$ is a neighbor-crowd degree of device i for neighbor device j at t.

4. Proposal Method

In this section, we discuss the proposed dynamic swarm spatial scaling for MSC in a noisy environment so that the device senses a physical signal emitted by an event which includes random noises.

4.1 Swarm Behavior in a Noisy Environment

In an environment where a physical signal does not include random noises, the strength of the physical signal emitted by events monotonically increases when approaching the event. The increase in the strength of the physical signal sensed by the device corresponds to a decrease in the distance between the device and the event. In MSC, the device nearest the event (**Fig. 2** (a)) is selected as a leader, which the other devices follow.

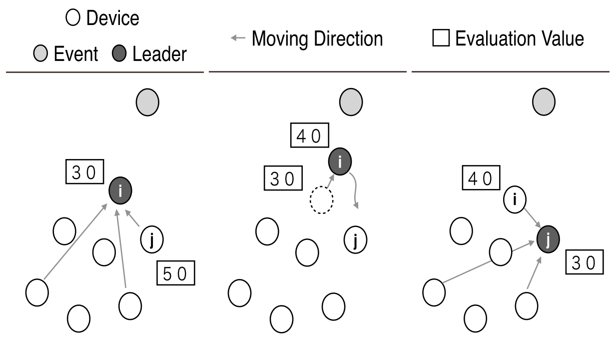
In a noisy environment, where the physical signal sensed by a device includes random noises, two incorrect behaviors emerge in an MSC swarm. One is that the leader moves in an incorrect direction to an event: when the personal best evaluation value in a leader increases, that is, when a leader moves away from an event, it turns in the opposite direction to where the personal best evaluation value increases. But as shown in Fig. 2 (b), the leader moves in an incorrect direction because the strength of the physical signal from events oscillates by random noise, and an increase in the personal best evaluation value does not always correspond to moving away from an event. Consequently, a leader may move in an incorrect direction, and MSC requires more time to search for events. The incorrect behavior is called Pbest noise.

The other is the leader-selection noise shown in Fig. 2 (c). In a noisy environment, the strength of the physical signal from events oscillates by random noise, and the decrease in the personal best evaluation value does not always correspond to the approach to an event. Therefore, a device with the smallest personal best evaluation value is not always nearest the event (device j on Fig. 2 (c)). An incorrect device may be selected as a leader in the swarm. Consequently, the swarm spends more time searching for and manipulating events.

The Pbest noise is incorrect behavior in an individual leader, who retains the nearest device to an event without any occurrence in the leader-selection noise. On the other hand, in the leader selection noise, a device that is not nearest to an event behaves as a leader, the followers approach the incorrect leader, therefore, all the devices, that is, the swarm moves in an incorrect direction. Therefore, the leader-selection noise more strongly impacts the searching manipulation performance than the Pbest noise.

4.2 Dynamic Swarm Spatial Scaling Mechanism

The dynamic swarm spatial scaling MSC avoids the leader-

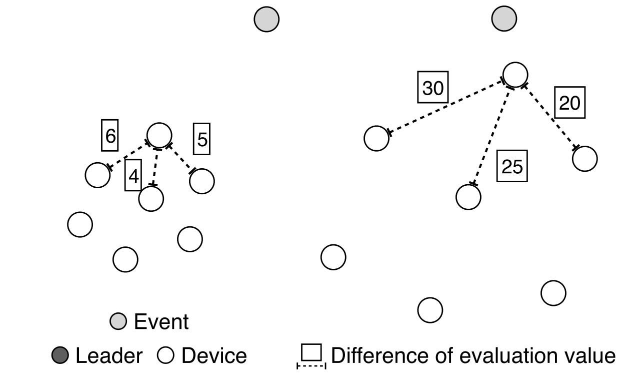


(a) Behavior of devices not in a noisy environment

(b) Pbest noise of a leader in a noisy environment

(c) Leader selection noise in a noisy environment

Fig. 2 Value, Direction, and Environment for internal consistency.



(a) Behavior of devices not in a noisy environment

(b) Pbest noise of a leader in a noisy environment

Fig. 3 Direction in evaluation value by dynamic swarm-scale control.

selection noise in a noisy environment by using a dynamic swarm spatial scaling mechanism. If the distance is small between the devices in a swarm, that is, the swarm's scale is spatially small, the devices in a swarm are located near each other, and their evaluation values that are near the event are also close to each other. Therefore, the relation between their evaluation values on spatially small-scale swarms is easily disordered by random noise in the physical signal, and leader-selection noise occurs easily (**Fig. 3** (a)). In other words, a spatially small-scale swarm sensitively reacts to random noise in the physical signal. On the other hand, if the distance between devices in a swarm is large, that is, if the swarm scale is spatially large, the devices in it are located far from each other, and their evaluation values are clearly different. Therefore, the relation between their evaluation value on a spatially large-scale swarm is difficult to disrupt by random noise in the physical signal, and leader-selection noise rarely occurs (Fig. 3 (b)). In other words, a spatially large-scale swarm can absorb the random noise in its physical signal.

As mentioned above, a dynamic swarm-scaling mechanism spatially inflates the swarm to eliminate leader-selection noise. However, if the swarm scale remains large, the number of devices reaching an event decreases, and the number of devices manipulating an event decreases. On the other hand, the random noises in the physical signal will probably decrease when approaching an event because obstacles and interferences between a device and an event decrease when approaching an event. Therefore, a dynamic swarm-scaling mechanism spatially shrinks the swarm as it approaches events and aims to obtain a sufficient number of devices in the manipulation phase. At a point far from the event, the mechanism inflates the swarm's scale to absorb random noises and shrinks its scale for more devices to manipulate when approaching an event. For the above behavior to emerge in a swarm, the mechanism utilizes the avoidance weight in Eq. (9), which represents the repulsive force among the devices:

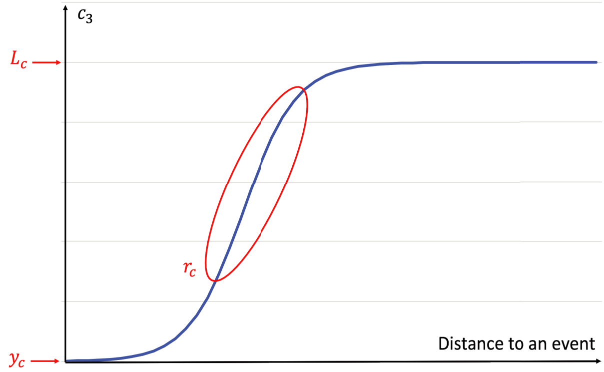


Fig. 4 Transition of c_3 with dynamic swarm-scale mechanism.

$$c_3^i(t) = \frac{y_c L_c}{y_c + (L_c - y_c)e^{-r_c E_i(t)}}, \tag{18}$$

where y_c is a lower limit of the avoidance weight, L_c is its upper limit, and r_c is its slope.

In **Fig. 4**, the x-axis shows the distance to an event, and the y-axis shows the c_3 in Eq. (18). L_c is the upper limit of c_3, and y_c is the lower limit respectively. r_c represents the slope of c_3 calculated by Eq. (18). Based on Fig. 4, c_3 inflates the scale of a swarm at a point far from an event and shrinks it when approaching an event by controlling the avoidance weight based on a device's evaluation value.

5. Evaluation

This Section shows the effectiveness of dynamic swarm spatial scaling MSC with simulation.

5.1 Simulation Specifications

The simulation parameters are listed in **Table 1**. The devices and events are defined as follows:

- A device is equipped with an IEEE802.11b interface and periodically advertises its information (Section 3.2.7).
- An event is equipped with an IEEE802.11b interface and periodically advertises a beacon including event identities as a MAC address.

Each device receives information from neighboring devices and beacons from events, which it can identify based on the received beacons. Each device also derives the following three evaluation values:

- Personal best evaluation value (E_i^{Pbest})
  Based on Eqs. (14), (15), a personal best evaluation value is defined as

$$E_i^{Pbest(K)}(t) = \min_{k \in discovery_i(t)} \{|RSSI_i^k(t)| + c_4|D_i^K(t)|\}, \tag{19}$$

Table 1 Simulation parameters.

Parameters	Values
Simulator	ns3
Simulation time (sec)	5000
Number of trials for each simulation scenario	10
Number of devices	10~30
Number of events	1,10~30
Initial location of devices (m×m)	(30, 30)
Initial location of events (m×m)	(100, 100)
Update cycle of velocity vector (sec)	0.1
Inertia weight w	0.5
Avoidance degree k	2
Coefficient of event crowd degree	−10
θ of β in Eq. (3)	30
M^{upper} in search phase (m/sec)	1
M^{upper} in manipulation phase (m/sec)	0.3
Manipulation capacity of event	300
Wireless communication	IEEE802.11b
Transmission power (dBm)	17.0206
Fading model	Rician fading
K-factor (dB)	1
Transition threshold to manipulation phase (dBm)	−50.6262
Distance to collision D_c (m)	1
y_c in Eq. (18)	5
L_c in Eq. (18)	1000
r_c in Eq. (18)	0.1 and 0.3

where $RSSI_i^k(t)$ is the Receive Signal Strength Indicator (RSSI) of a beacon that device i receives from K at t and $discovery_i(t)$ is a set of events from which device i receives beacons at t. If a device cannot receive a beacon from any event, let the personal best evaluation value be a positive infinity.

- Local best evaluation value (E_i^{Lbest})
  Based on Eqs. (16), (17), a local best evaluation value is defined as

$$E_i^{Lbest}(t) = \min_{j \in neighbor} \{E_j(t) + c_4 |N_i^j(t)|\}. \tag{20}$$

- Self-evaluation value (E_i)
  Based on Eq. (6), a self-evaluation value is defined as

$$E_i(t) = \begin{cases} E_i^{Pbest(K)}(t) \\ \quad \text{if } E_i^{Pbest(K)}(t) < \min_{j \in neighbor_i(t)} \{E_j^{Pbest(K)}(t)\} \\ E_i^{Lbest} + |RSSI_i^{Lbest}(t)| \\ \quad \text{otherwise,} \end{cases} \tag{21}$$

where $RSSI_i^{Lbest}(t)$ is the RSSI of the information that device i received from a device treated as a local best device at t.

If the distance between two devices becomes lower than threshold (D_c), it is judged that the two devices will collide. The devices stop to advertise their information and create a moving vector.

An event has manipulation capacity, which is a measure of the time taken from reaching an event to the completion of manipulating an event. The device in the manipulating phase decreases 1 manipulation capacity per sec. When the manipulation capacity of an event becomes 0, the event disappears from the simulation field.

In this simulation, it is assumed that the physical signal from an event is radio waves. We applied a radio-propagation model to the rice-fading model [19], and the RSSI included random noises.

Table 2 Comparison methods.

	c_3 in search phase	c_3 in manipulate phase
Previous Method 1	25	5
Previous Method 2	1000	5
Proposed Method 1	c_3 is controlled dynamically based on r_c 0.3	
Proposed Method 2	c_3 is controlled dynamically based on r_c 0.1	

5.2 Comparison Methods

In this simulation, we compared the four methods listed in **Table 2**. Previous methods 1 and 2 are mechanisms based on MSC. The c_3 of the previous method 1 is 25, and the c_3 of the previous method 2 is 1000. Therefore, the previous method 1 searches for an event by a constantly small-scale swarm, and the previous method 2 searches for an event by a constantly large-scale swarm. Our proposed methods 1 and 2 are mechanisms with a dynamic swarm spatial scaling mechanism. The r_c in Eq. (18) of proposed method 1 is 0.3, and that of proposed method 2 is 0.1. That is, proposed method 1 shrinks the swarm scale more rapidly than proposed method 2 when approaching an event. The four methods were evaluated by turnaround time, which is the time to finish searching for and manipulating all events. If the devices cannot complete searching for and manipulating at all events, let the turnaround time be the simulation time.

5.3 Evaluation based on Simulation Results

5.3.1 Turnaround Time

The dependence of turnaround time on the number of devices is shown in **Fig. 5**. The turnaround time of proposed method 1 was lower than that of previous methods 1 and 2 regardless of the number of devices and events. Comparing the proposed method 1 and 2 on turnaround time, they were almost equivalent on the number of events 1, but that of proposed method 1 was lower than that of proposed method 2 in the other cases. Proposed methods 1 and 2 maintain a large swarm scale when they locate far from an event, but proposed method 1 more rapidly shrinks the swarm scale than proposed method 2 when approaching an event. Therefore, to absorb random noises, maintaining a large swarm scale until approaching an event is effective. However, when approaching an event, shrinking the swarm scale is effective because of the significant decrease in the random noises from an event.

5.3.2 Dependence of Turnaround Time on the Manipulation Capacity of an Event

The dependence of the turnaround times on the manipulation capacity of an event when the manipulation capacity is varied from 100 to 500 is shown in **Fig. 6**. Proposed methods 1, 2 and previous method 1 were almost equivalent. Previous method 2 was significantly inferior to the others. Proposed methods 1 and 2 shrank the swarm scale near an event, and previous method 1 kept it small; therefore, a large number of devices could manip-

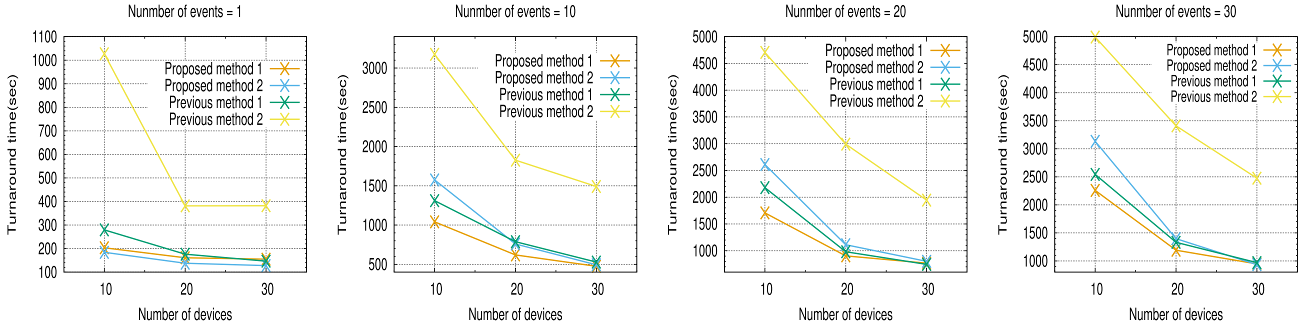


Fig. 5 Dependence of turnaround time on number of devices.

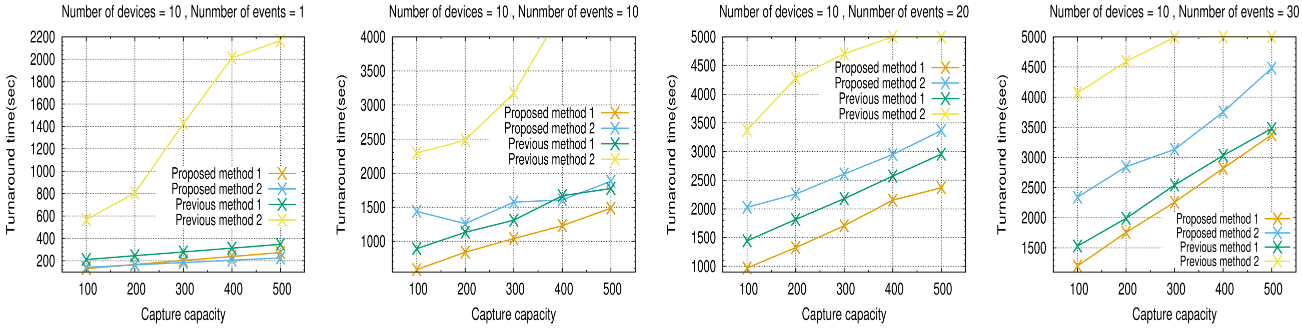


Fig. 6 Dependence of turnaround time on manipulation capacity of an event.

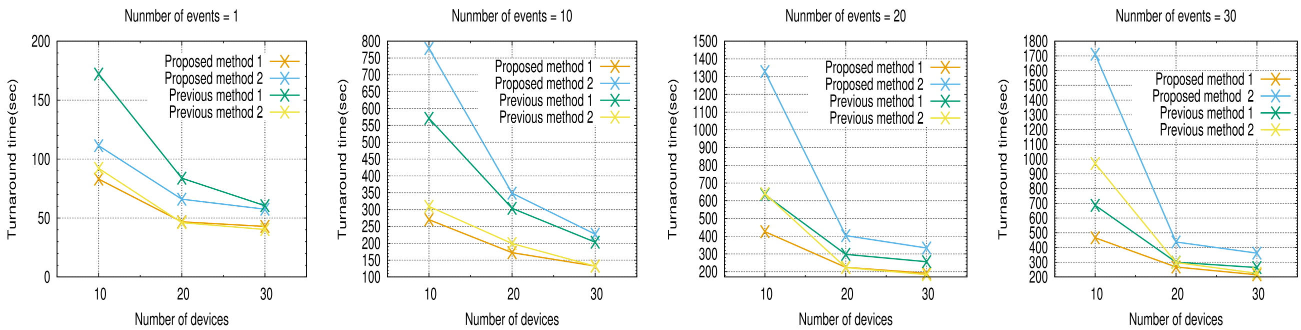


Fig. 7 Dependence of turnaround time on searching time.

ulate an event, which reduced the time. Previous method 2 kept the swarm scale large, and the number of devices near an event was small; the number of manipulating devices to the event was also small. Therefore, previous method 2 required more time to manipulate an event.

5.3.3 Dependence of Turnaround Time on Searching Time

Figure 7 shows the turnaround time of each method when the manipulation capacity was 1, therefore, it shows the comparison with search time, which is turnaround time without manipulating time, in each method. Proposed method 1 outperformed the others regardless of the number of devices and events. The swarm-scaling mechanism (Section 5.3.1) absorbed random noises.

Table 3 lists the ratios of the leader-selection noise and the pbest noise of a leader. As mentioned above, the leader-selection noise is incorrect behavior when the device, which is not the nearest to an event, is selected as a leader. The pbest noise is incorrect behavior when the leader moves away from an event due to the pbest with random noise.

Table 3 shows that the pbest noise of a leader for each method ranged from 28 to 37%, and the maximum difference among the methods was about 10%. The leader-selection noise for each

Table 3 Swarm noise.

	Leader-Selection noise	Pbest noise of Leader
Previous Method 1	74.03%	37.69%
Previous Method 2	48.86%	34.52%
Proposed Method 1	51.38%	28.77%
Proposed Method 2	57.42%	34.65%

method ranged from 48 to 74%, and the difference between previous method 1, which kept the swarm scale small, and the other three methods, which kept the swarm scale large at a point far from an event, was about 25%. The leader was correctly selected by inflating the swarm scale at a point far from an event. By inflating the swarm scale, the difference in the evaluation value between devices increased and became so large that it was not affected by the random noise. As a result, the disorder in the relations between the personal best evaluation values derived by each device decreased, and the leader was correctly selected.

5.4 Dependence of Turnaround Time on K-factor

Figure 8 shows the dependence of turnaround time on K-factor when the manipulation capacity of an event is 1, therefore, shows

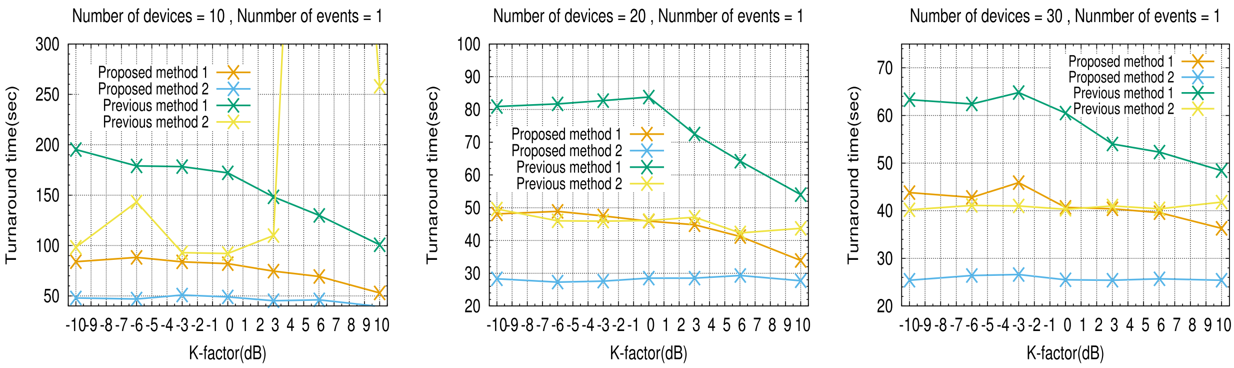


Fig. 8 Dependence of turnaround time on K-factor.

the dependence of search time on K-factor. K-factor indicates the ratio of indirect wave power to direct wave power. A high K-factor means that RSSI is mainly composed of direct wave power, and RSSI includes less random noises, on the other hand, a low K-factor means that RSSI is mainly composed of indirect wave power, and that RSSI includes many random noises. Therefore, Fig. 8 shows the tolerance to random noise in a physical signal in the search function of each method.

As shown in Fig. 8, proposal method 1 and previous method 1 have the trend that they decrease the turnaround time according to the K-factor increase, and then the turnaround time in proposal method 2 is lowest and constant regardless of the K-factor. In addition, in most of the K-factors, the proposal methods outperform the previous methods. The difference in the times between the proposal methods 1, 2 and the previous method 1 increase as K-factor decreases. The reason is that previous method 1 is more affected by random noise than the proposal methods because it increases the number of the leader-selection noises by small swarm scale.

Comparing with the turnaround time of proposed methods 1 and 2, that of proposed method 2 is constant regardless of the K-factor, and much lower than that of proposed method 1. Therefore, the following behaviors in swarm emerged by the swarm spatial scaling mechanism is effective to absorb any random noises in physical signal from an event and to decrease the turnaround time.

- When the random noise is large in a location far from an event, that is, when the physical signal from an event is inaccurate, the form of swarm becomes sparse and loose by inflating swarm scale and swarm moves to an event while spreading in multiple directions.
- When the random noise decreases on approaching an event, that is, when the physical signal from an event becomes accurate, the swarm tightly moves directly to an event by shrinking swarm scale.

Based on the above simulation results, our proposed mehcanism, which dynamically controls a spatial scale in a swarm, reduces the turnaround time even in a noisy environment.

6. Conclusion

We proposed a dynamic swarm spatial scaling mechanism for a swarm composed of multiple autonomous mobile devices in noisy environments containing a physical signal including random noises. The proposed mechanism increases the scale of a swarm in the search phase to improve tolerance against random noises and decreases the scale as the swarm approaches an event. Simulation showed that the proposed mechanism decreases the time required to search for and manipulate an event in a noisy environment with random noises.

In this paper, we assume that there are no obstacles or barriers in the simulation area, but there are many obstacles or barriers in a real environment. Therefore, we will investigate the application of MSC for space with obstacles as future work.

References

[1] Allan, C., Sibonelo, M. and Riaan, S.: Survey and requirements for search and rescue ground and air vehicles for mining applications, *M2VIP*, pp.105–109 (2012).

[2] Bellingham, J. and Rajan, K.: Robotics in Remote and Hostile Environments, Science2007, Vol.318, No.5853, pp.1098–1102 (2007).

[3] Nii, E., Kitanouma, T., Hirose, W., Yomo, H. and Takizawa, Y.: Mobile Sensing Cluster based on Swarm Intelligence with Multiple Autonomous Mobile Devices, *IPSJ Journal* (2020) (in Japanese).

[4] Qianying, P. and Hongtao, Y.: Survey of particle swarm optimization algorithm and its application in antenna circuit, *2015 IEEE ICCP*, pp.492–495 (2015).

[5] Tan, Y. and Zhong-Yang, Z.: Research Advance in Swarm Robotics, *Defence Technoloty*, Vol.9, No.1, pp.18–39 (2013).

[6] Adham, A. and David, M.W.P.: The Use of Area Extended Particle Swarm Optimization (AEPSO) in Swarm Robotics, *2010 11th International Conference on Control Automation Robotics & Vision*, pp.591–596 (2010).

[7] Siddarth, J., Manish, S. and Vijay, K.C.: Ad-hoc swarm robotics optimization in grid based navigation, *2010 11th International Conference on Control Automation Robotics & Vision*, pp.1553–1558 (2010).

[8] Reynolds, W.C.: Flocks herds and schools: A distributed behavioral model, *SIGGRAPH Comput. Graph.*, Vol.21, No.4, pp.25–34 (1987).

[9] Eversham, J., Ruiz, F.V.: Parameter analysis of Reynolds flocking model, *2010 IEEE 9th International Conference on Cybernetic Intelligent Systems*, pp.1–7 (2010).

[10] Hauert, S., Leven, S., Varga, M. and Ruini, F.: Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.5015–5020 (2011).

[11] James, K. and Russell, E.: Particle Swarm Optimization, *Proc. 1995 IEEE International Conference on Neural Networks*, pp.1942–1948 (1995).

[12] Kangtai, W. and Fupeng, L.: A dynamic chaotic mutation based particle swarm optimization for dynamic optimization of biochemical process, *ICISCE*, pp.788–791 (2017).

[13] Yuanbin, M., Hetong, L. and Qin, W.: Conjugate direction particle swarm optimization solving systems of nonlinear equations, *COMPUT MATH APPL*, Vol.57, No.11-12, pp.1877–1882 (2009).

[14] Le, Y., Dakuo, H., Qingkai, W., Jiahuan, L., Yingjie, H. and Zipeng, Z.: Particle Swarm Optimization Algorithm Based on Robust Control of Random Discrete Systems, *ICISCE*, pp.1089–1093 (2017).

[15] Soririos, S.: Particle Swarm Optimization and Application in Robotics: A Survey, *0218 9th IISA*, pp.1–7 (2018).

[16] Saber, O.R., Fax, A.J. and Murray, R.M.: Consensus and Cooperation

in Networked Multi-Agent Systems, *Proc. IEEE*, Vol.95, pp.215–233 (2007).
[17] Lingyu, L. and Weili, N.: Consensus Problems on Multi-agent Networks with Directed Dynamic Interactions, *2017 6th DDCLS*, pp.421–426 (2017).
[18] Qiang, J. and Wallce, K.S.T.: Consensus of Multi-Agents with Event-Based Nonlinear Coupling over Time-Varying Digraphs, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.65, No.12 (2018).
[19] Amirhossein, A., Saeed, F.F., Bruce, F.C. and Christian, S.: Compact Rayleigh and Rician fading simulator based on random walk processes, *IET*, Vol.3, No.8, pp.1333–1342 (2009).

Eiji Nii received his B.S., M.S., and Ph.D. degrees from Kansai University, Osaka, Japan, in 2015, 2017, 2020, respectively. His research interest is in control multiple autonomous mobile devices with swarm intelligence. He is a member of IPSJ and IEICE.

Shoma Nishigami received B.S. degree from Kansai University, Osaka, Japan, in 2020. His research interest is in control multiple autonomous mobile devices with swarm intelligence.

Takamasa Kitanouma received B.S., M.S., and Ph.D. degrees from Kansai University, Suita, Japan, in 2014, 2016, 2019, respectively. He is currently CEO of Phindex Technologies Corporation, Osaka, Japan, in 2020. His research interest is a self-organizing in wireless networks. He is a member of IPSJ.

Hiroyuki Yomo received his B.S., M.S., and Ph.D. degrees from Osaka University, Suita, Japan, in 1997, 1999, and 2002, respectively, all in communication engineering. From April 2002 to March 2004, he was a Postdoctoral Fellow with Aalborg University, Aalborg, Denmark. From April 2004 to September 2004, he was with NEC Corporation, Tokyo, Japan. In October 2004, he joined Aalborg University as an Assistant Research Professor and became an Associate Professor from February 2006 to March 2008. From April 2008 to March 2010, he was a Senior Researcher with Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. Since April 2010, he has been an Associate Professor with Kansai University, Suita, Japan. He is also affiliated with Aalborg University as a part-time Lecturer and with ATR as a Guest Researcher. His main research interests are access technologies, radio resource management, and link-layer techniques in the broad area of wireless communications.

Yasuhisa Takizawa received his B.S. degree in mechanical engineering from Kyoto Institute of Technology, Kyoto, Japan, in 1983. He received Dr. Eng. degree from Ritsumeikan University, Shiga, Japan, in 2003. From 1983 to 1990 he worked mainly for research and development of unix basic software at Nihon Unisys and also worked mainly for research of distributed operating system at Sumitomo Metals from 1990 to 1998. From 1990 to 2009, he worked as an advanced senior researcher at Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan. In 2009, he joined Kansai University, Osaka, Japan, as an Associate Professor, since 2014, he has been a Professor with Kansai University. His current research interests are in self-organizing and adaptive control for wireless networks. He is a member of IEEE and IEICE.