

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

6,900

Open access books available

186,000

International authors and editors

200M

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)



# Adaptive Sensor-Network Topology Estimating Algorithm Based on the Ant Colony Optimization

Satoshi Kuriharam, Hiroshi Tamaki, Kenichi Fukui and Masayuki Numao  
Osaka University  
Japan

## 1. Introduction

Studies on ubiquitous-computing and/or ubiquitous-network systems have recently been very attractive research topics along with interest in rapid developments in information and communications technologies [2]. All kinds of information devices such as computers, cellular phones, electrical appliances, and various sensors will be connected in future ubiquitous-network systems. Consequently, everyone will be able to make use of all kinds of information without stress, anywhere and at any time.

However, we need to bring the idea of "context-awareness" as an essential element into reality to construct ubiquitous-network systems. "Context" refers to the situation in which ubiquitous-network devices are embedded, and "awareness" refers to the recognition of "context" with the ubiquitous network. One goal of context-awareness was to acquire and utilize information about the context to provide services that were appropriate to particular people, places, times, and events. Ubiquitous-network systems were also expected to implement context-awareness by using a sensor network, which was made up of many sensors.

Information about adjacent relationships between sensors is necessary [6] to extract human motion in a sensor-network system. Adjacent relationships indicate the physical connectivity between sensors from the point of view of a person's movement in a sensor network (Fig. 1). This information is usually sent to the system manually [3]. However, investigating and inputting information on the sensor topology becomes much too difficult in proportion to the scale of the sensor-network system. In addition, once the structure of the network changes, e.g., by adding/removing sensor units and by rearranging the network, we have to manually reinvestigate adjacent relationships. Incessantly repetitive investigations and inputs are onerous, and increase the possibility of making mistakes. There is no room for

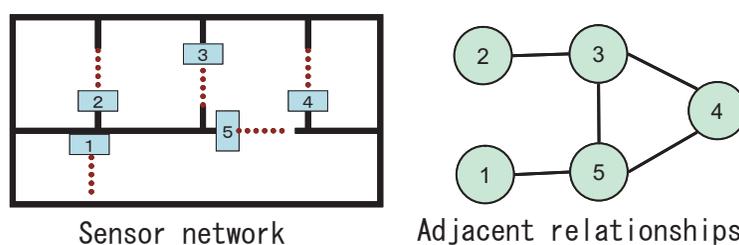


Fig. 1. Adjacent relationships in sensor network

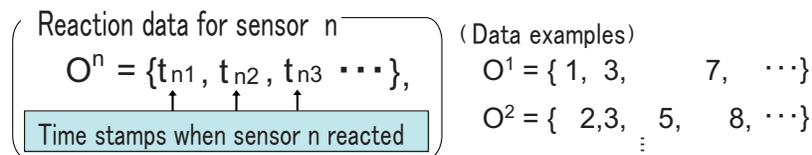


Fig. 2. Format for sensor reaction data

investigating modified topologies, especially in emergency situations. Consequently, it is necessary for sensor-network systems to be able to acquire their own structures automatically without any previous knowledge.

Marinakos and Dudek constructed adjacent relationships in a sensor network and the traffic patterns of people in the network using stochastic expectation maximization [4]. They succeeded in constructing a flexible algorithm independent of any previous knowledge and it was only reliant on information from data observed by all the sensors. However, as the performance of their algorithm strongly depended on the number and traffic patterns of people in the environment, it was based on the assumption that the number of people in the network and their traffic patterns were constant. Moreover, their algorithm required a great deal of analysis with various estimates as to the number of people and traffic patterns to derive the number of persons. It took too much time for it to be of practical use.

This paper proposes an algorithm that could predict sensor relationships by only using real or simulated sensor data. It had to have at least three capabilities: (1) Anytime characteristics - it had to be able to output optimal results through analysis over a finite period of time even if the network scale or the frequency of movement increased. (2) Adaptability - the algorithm had to be free of special tuning to accommodate all types of environments. It also had to automatically adapt to changes in network structures. (3) Robustness - it had to maintain extremely accurate analysis results even if the ratio of noise data caused by the simultaneous movements of numerous people or sensing errors increased.

The Ant Colony Optimization (ACO) algorithm is one of the most well known models of pheromone communication derived from the swarming behavior of ants in nature [1]. ACO is recognized as being extremely robust against and adaptable to dynamic changes in the environment, and various kinds of optimization problems have been solved by using ACO-based approaches [5]. As the capabilities of ACO were very appealing to meet our needs, we constructed our algorithm based on this.

The rest of this paper is organized as follows. Section 2 presents an outline of the proposed algorithm. Section 3 describes the details. Sections 4 and 5 describe the experimental results using simulated sensor data and actual data. Section 6 presents the conclusions drawn from this study.

## 2. Pheromone communication algorithm

### 2.1 Estimates of standard travel time between sensors

We used the sensor reaction data presented in Fig. 2 in the proposed algorithm, which shows that sensor  $s_1$  reacted at time steps of 1, 3, 7, and 10, and sensor  $s_2$  reacted at time steps of 2, 3, 5, 8, and 13.

If sensors  $s_i$  and  $s_j$  are adjacent, the travel time between two sensors for any number of people is roughly the same. Consequently, when sensor  $s_i$  reacts at  $timestept_1$  and sensor  $s_j$  reacts at  $timestept_2$ , and  $|m_{i,j} - (t_2 - t_1)|$  becomes too small, where  $m_{i,j}$  is the average travel time between  $s_i$  and  $s_j$ , these two sensors are considered to have reacted to one person traveling

from  $s_i$  to  $s_j$ . Conversely, when  $|m_{i,j} - (t_2 - t_1)|$  becomes too large, these two sensors are considered to have reacted to two different people. Therefore, if  $m_{i,j}$  is calculated from a very long data log of sensor reactions and  $|m_{i,j} - (t_2 - t_1)|$  regularly becomes too small,  $s_i$  and  $s_j$  are considered to be adjacent even if we do not know the relation between both sensors in advance. At this point, we can define  $\omega_{i,j} = \frac{1}{|m_{i,j} - (t_2 - t_1)|}$  as the likelihood that sensors  $s_i$  and  $s_j$  will be adjacent to each other.

However, the methodology for estimating the adjacent relation between two sensors by using  $\omega$  suffers from several weaknesses. For example, even when different people have simultaneous reactions to either sensor and the interval time nears their standard travel time, we mistakenly think both sensors are adjacent. Of course, it is necessary to consider noise such as that from sensor mis-reactions, and it is also important to consider dynamical changes in the adjacent relations of sensors (e.g., when office and room furniture is occasionally re-arranged).

## 2.2 Necessary functions

Four functions can effectively solve these problems.

### 1. Positive feedback loop

A positive feedback loop is necessary to accelerate the process by which two such sensors having a higher likelihood of adjacency can be given a higher  $\omega$ .

### 2. Avoidance of local solutions

It is necessary to avoid mis-detections where two sensors cannot be considered to be in an adjacent relation, due to the extreme bias of adjacency likelihood. "Item 1" and this item are a trade-off.

### 3. Using other criteria to calculate the adjacency likelihood

Using other criteria is also necessary to improve accuracy and convergence, and not only using the interval time of sensor reaction, which is the basic methodology used to consider whether two sensors are in an adjacent relation.

### 4. Deletion of old information

It is necessary to delete relatively old information from the sensor-reaction data log to adapt to dynamic changes in the environment.

Although these items can all effectively be used to solve problems, it is difficult to coordinate them manually, due to the trade-offs between them. As a result, we propose a new methodology in this paper for calculating the adjacency likelihood between sensors based on ACO (Ant Colony Optimization, which is one of the best known pheromone communication models). All four items in the proposed methodology were implemented and adjustments between them were autonomously done.

The adjacency likelihood in our methodology is considered to be the amount of pheromone, and this is accumulated by using both the adjacency likelihood calculated from past sensor reaction data and the presumed distance between sensors derived from the standard travel time. The positive feedback loop where the adjacency likelihood is entirely increased is formed in sensors with adjacent relations by the accumulation mechanism. However, noise due to simultaneous sensor reactions by several persons can be deleted from non-adjacent sensors. Moreover, since pheromones gradually evaporate old information is automatically removed. This means that new information is always given priority. The coordination of items is autonomously and indirectly controlled by the interaction of agents using pheromones.

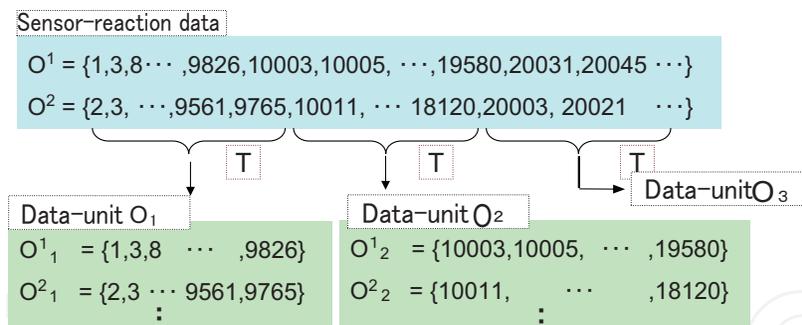


Fig. 3. Example data unit for  $T = 10000$ .

### 2.3 Virtual graph

The proposed methodology using the pheromone-communication model is executed on a virtual graph,  $G$ , where agents move between sensors and pheromone interactions occur.  $G = (V, E)$  is a virtual directed graph and each node  $v_i \in V$  corresponds to each sensor  $s_i \in S$  in a real environment. Edge  $e_{i,j} \in E$  means the edge from node  $v_i$  to  $v_j$ . As there is no information about adjacent relations,  $G$  is initially the complete graph before calculation is started.

Several kinds of pheromones are accumulated and deleted on each  $e_{i,j}$ , and we propose three kinds:  $\tau$ ,  $\omega$ , and  $\epsilon$ .

- Pheromone concerned with adjacency likelihood  $\omega$  (called "edge pheromone" after this)

The entire sensor-reaction data in the proposed methodology is divided into several data units (explained in the next section), and calculation is repeatedly done for each unit.  $\omega$  indicates the adjacency likelihood for each unit and is always initialized before the calculation for each data unit.

- Pheromone being output by agents  $\epsilon$  (called "agent pheromone" after this)

Each agent outputs  $\epsilon$  corresponding to  $\omega$  on each edge as the evaluated result for the edge.  $\epsilon$  is also always initialized before each data unit is calculated.

- Pheromone distribution data set  $\tau$  (called "pheromone distribution" after this)

$\tau$  indicates the accumulation of each  $\epsilon$ , i.e.,  $\tau$  specifies the total adjacency likelihood of reflecting the calculation results for all units.  $\tau$  is updated whenever the calculation for a new unit has finished.

Initially,  $\tau(0)$  (in this paper,  $\tau(0) = 300$ ) is assigned to whole edges.  $a$  agents move on the  $G$  and they output pheromones, depending on the number of pheromones on each edge. They are initially placed on the  $G$  homogeneously.

### 2.4 Calculation details

The entire sensor reaction data is divided into several data units  $O_t$  containing data on the interval of time  $T$  as shown in Fig. 3.

Standard travel time  $m$  and pheromone distribution data set  $\tau$  are updated whenever the calculation for a data unit has finished. Whenever the calculation for a data unit has completed, whether two sensors are adjacent is determined for all the sensors. The locations of all agents, and  $\omega$  and  $\epsilon$  are initialized to calculate the next data unit. The next data unit is calculated using  $m$  and  $\tau$  updated in the previous process. Fig. 4 shows the whole sequence for the algorithm.

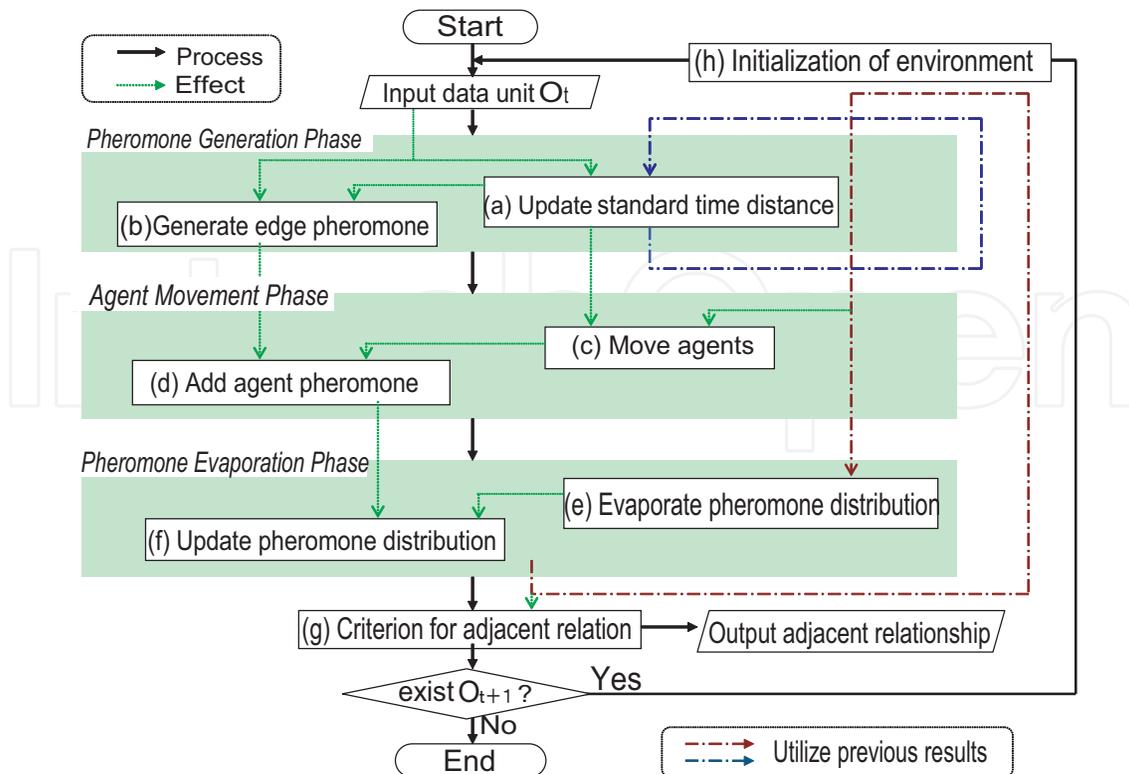


Fig. 4. Overview of proposed algorithm

### 3. Algorithm for acquisition of adjacent relations

This section describes how the algorithm and adjacent relations are determined in detail.

#### 3.1 Pheromone generation phase

This phase updates  $M(t)$ <sup>1</sup> from  $O_t$  and generates  $\omega(t)$  onto the graph.

##### (a) Update standard travel time, $M$

First, a frequency distribution,  $d_{i,j}(t)$ , for the time distance is created from  $O_t$ .  $d_{i,j}^c(t)$  represents a frequency where the time interval between two sensors,  $s_i$  and  $s_j$ , is equal to  $c$  within  $O_t$ . After an accumulated frequency distribution,  $D_{i,j}(t)$ , is updated using the current  $d_{i,j}(t)$  (Eq. (1)), standard travel time  $M(t)$  is updated using  $D_{i,j}(t)$  (Eq. (2)). Since  $D_{i,j}(t)$  is the accumulated time distance  $d_{i,j}(t)$  of each data unit,  $M(t)$  is the most frequent time distance up to now.

$$D_{i,j}^c(t) = D_{i,j}^c(t-1) + d_{i,j}^c(t) \quad (1)$$

$$m_{i,j}(t) = \arg \max_{1 \leq c \leq 50} D_{i,j}^c(t) \quad (2)$$

##### (b) Generate edge pheromone, $\omega$

$\omega_{i,j}(t)$  is represented by the sum of adjacency likelihoods  $\Delta\omega_{i,j}^c(t)$ , where each time interval  $c$  has

<sup>1</sup> $M(t)$  equals set of standard travel time  $m(t)_{i,j}$   
 $M(t) = \{m(t)_{i,j} | \forall i, j, i \neq j\}$ .

$$\Delta\omega_{i,j}^c(t) = \frac{d_{i,j}^c(t)}{|m_{i,j}(t) - c|} \quad (3)$$

$$\omega_{i,j}(t) = \sum_c \Delta\omega_{i,j}^c(t) \quad (4)$$

### 3.2 Agent movement phase

Each agent in this phase moves based on  $\tau(t)$  and  $M(t)$ , and adds  $\epsilon(t)$  onto an edge based on  $\omega(t)$ .

#### (c) Move agents

Each agent only moves once on an edge within an agent movement phase, and can move to any node. Each agent selects a route based on previous search result  $\tau(t)$  and heuristics  $\eta(t)$ .  $\eta_{i,j}(t)$  is the inverse of  $m_{i,j}(t)$  on its route. The less  $m_{i,j}(t)$  is, i.e., the shorter the distance between sensors is, the larger the value is (Eq. (5)).

$$\eta_{i,j}(t) = \frac{1}{m_{i,j}(t)} \quad (5)$$

All agents are set to prefer an edge where there is a lot of  $\tau(t)$  remaining. Therefore, they can focus their search on edges that have a high adjacency likelihood obtained by a previous search. Also, since agents are set up to prefer routes that have high  $\eta(t)$ , they can conduct more effective searches. It is highly likely that adjacent sensors will be located nearer than non-adjacent sensors. The probability,  $p_{i,j}^k(t)$ , that agent  $k$  will moves from  $v_i$  to  $v_j$  is given by:

$$p_{i,j}^k(t) = \frac{[\tau_{i,j}(t)][\eta_{i,j}(t)]^\gamma}{\sum_{j,j \neq i} [\tau_{i,j}(t)][\eta_{i,j}(t)]^\gamma} \quad (6)$$

$\gamma$  is the weight for  $\tau(t)$ , giving priority to the information for all agents. In addition, all agents are set up to select routes randomly with constant probability  $r$  not depending on  $\tau(t)$ . This prevents the discovered adjacent relation from falling into a local solution. The local solution might be obtained by the effectiveness of the feedback loop of a pheromone increasing.  $\gamma = 1$  and  $r = 0.1$  are used in this paper.

#### (d) Add agent pheromone, $\epsilon$

After moving, each agent discharges an agent pheromone,  $\Delta\epsilon_{i,j}^k(t)$ , onto its edge as the evaluation value for the route. The evaluation value reflects information on  $O_t$  using  $\omega_{i,j}(t)$ , which is generated in the pheromone generation phase. Equation (7) represents agent pheromone  $\Delta\epsilon_{i,j}^k(t)$  that agent  $k$  who passed  $e_{i,j}$  added onto  $e_{i,j}$ . Also, Eq. (8) represents the amount of agent pheromone  $\epsilon_{i,j}(t)$ , where  $n$  is the number of agents who moved to  $e_{i,j}$ .

$$\Delta\epsilon_{i,j}^k(t) = \omega_{i,j}(t) \quad (7)$$

$$\epsilon'_{i,j}(t) = \sum_{k=1}^n \Delta\epsilon_{i,j}^k(t) \quad (8)$$

$$\epsilon_{i,j}(t) = z \left\{ 1 - \left( 1 - \frac{1}{z} \right)^{\epsilon'_{i,j}(t)} \right\} \quad (9)$$

Equation (9) prevents strong bias in the pheromone distribution. When the sum of agent pheromone  $\epsilon'_{i,j}(t)$  that the agents discharged who moved to  $e_{i,j}$  is low, the amount of pheromone proportional to that value is added. However, the contribution due to the increase in agent pheromone decreases with increasing  $\epsilon'_{i,j}(t)$ .  $z$  was set to 1,000 in this paper.

### 3.3 Pheromone Evaporation Phase

$\tau(t)$  is evaporated and updated using  $\epsilon(t)$  in this phase.

#### (e) Evaporate pheromone distribution, $\tau$

$\tau_{i,j}(t)$  that is on each edge decreases according to the evaporation rate,  $\rho$ , in every evaporation phase. Equation (10) is used to calculate evaporation:

$$\tau'_{i,j}(t) = (1 - \rho)\tau_{i,j}(t) \quad (10)$$

#### (f) Update pheromone distribution, $\tau$

After being evaporated,  $\tau'_{i,j}(t)$  is updated by combining it with  $\epsilon_{i,j}(t)$ , which is generated in the agent movement phase. Equation (11) gives the formula for updating  $\tau_{i,j}(t)$  in  $O_t$ :

$$\tau_{i,j}(t+1) = \tau'_{i,j}(t) + \epsilon_{i,j}(t) \quad (11)$$

Later searching information is thereby reflected at a constant ratio by discarding old searching information. Evaporation rate  $\rho$  affects the update speed. The smaller  $\rho$  is, the more weight is given to later information.  $\rho = 0.01$  has been used in this paper. When  $\rho$  is small, a stable solution can generally be discovered; however, the convergence speed is slow. Whereas, when  $\rho$  is large, the convergence speed is fast; however, an unstable solution may be obtained depending on the data unit.

### 3.4 Determination of adjacent relations

This section describes how adjacent relations are determined after analysis is finished for  $O_t$  and  $G$  is initialized for  $O_{t+1}$ .

#### (g) Criterion for adjacent relations

$\tau(t)$  represented the adjacency likelihood as the result of total analysis up to now. An edge is determined as an adjacent relation when  $\tau(t)$  is greater than a certain threshold. The threshold is given by  $\alpha$  times the average of all pheromone distributions  $\tau(t)$  as shown in Eq. (12).  $\alpha$  has been set to 0.8 in this paper. Here,  $|V|$  is the number of nodes.

$$threshold(t) = \alpha \times \frac{\sum_{i,j} \tau_{i,j}(t)}{|V|^2} \quad (12)$$

#### (h) Initialization of environment

If the next data unit,  $O_{t+1}$ , exists after the above phases have finished, it restarts analysis from the pheromone generation phase using  $O_{t+1}$  after the environment has been initialized.

$\omega(t)$  and  $\epsilon(t)$  are discarded for initialization, and agents are also allocated evenly to each node. Only  $\tau(t+1)$ ,  $M(t)$ , and  $D(t)$  are inherited for the next analysis.

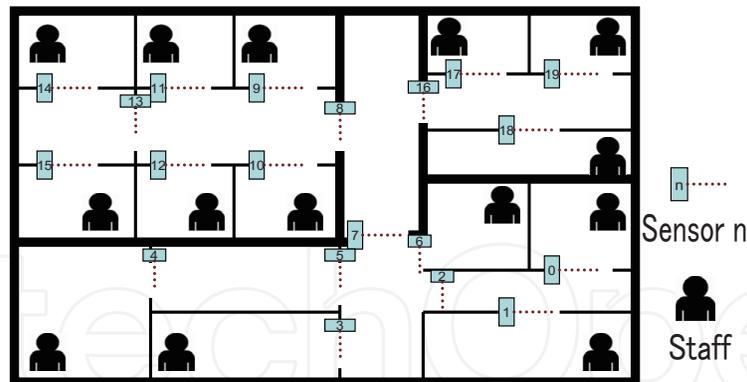


Fig. 5. Virtual environment,  $E_2$ .

#### 4. Empirical verification using simulation

We conducted simulation experiments to verify the basic effectiveness of the proposed algorithm. This section presents the results of experiments using simulated sensor data.

##### 4.1 Preparation of data for simulation

We prepared the sensor-reaction data for simulation using a virtual sensor-network environment (Fig. 5). This virtual environment was built with rooms and aisles the sensors were attached to, and in which staff moved around. Each staff member was asked to randomly repeat asynchronously moving from one room to another. In every time step, each decided whether to begin moving or not according to the probability of "movement frequency", which represented how often staff started to move. When a staff member passes in front of a sensor, the sensed data from this sensor was sent and stored with a time stamp. Mis-detection by the sensor also occurred, which depended on the probability of "mis-reaction frequency". Whenever time  $T$  passed, data unit  $O_t$  was created and analysis was carried out.

We prepared two patterns for a virtual environment in this experiment,  $E_1$  and  $E_2$ , which had 10 and 20 sensors, and also 6 and 14 staff members. The movement frequency was also set up for three patterns of  $F_{Low}$ ,  $F_{Normal}$ , and  $F_{High}$ , and the mis-reaction frequency was also set up for  $M_0$ ,  $M_{20}$ , and  $M_{50}$ <sup>2</sup>. By choosing these elements, we built a variety of virtual sensor networks. We used the following values for the algorithm parameters of  $a$  and  $T$ , depending on simulation elements  $E$  and  $F$ , i.e.,  $\{E, a\} = \{E_1, 2,000\}$ ,  $\{E_2, 4,000\}$ ,  $\{F, T\} = \{F_{Low}, 15,000\}$ ,  $\{F_{Normal}, 8,000\}$ , and  $\{F_{High}, 1,000\}$ .

##### 4.2 Verification of robustness

First, we verified the robustness of the algorithm against noise data. We divided the noise data into two types: the first was caused by movements by numerous walkers and the second by mis-reactions by the sensors. We controlled these noise probabilities by selecting simulation elements  $E$ ,  $F$ , and  $M$ .

We constructed two more algorithms to calculate adjacency likelihood for comparison, i.e., Test 1 and Test 2. In Test 1, adjacent relations were judged from adjacency likelihood  $\omega$  without being weighed. Test 2 was based on Test 1 and only adopted the  $\eta$  information to weight  $\omega$ .

We tested robustness with the noise-production probability related to the ratio of numerous people with simultaneously varying movements. Six kinds of virtual environments were prepared by pairing  $E$  with  $F$  ( $M_{20}$  was used), and for all six we calculated 1,000,000 time

<sup>2</sup> $M_x$  means that no sensors detected at a probability of  $x\%$

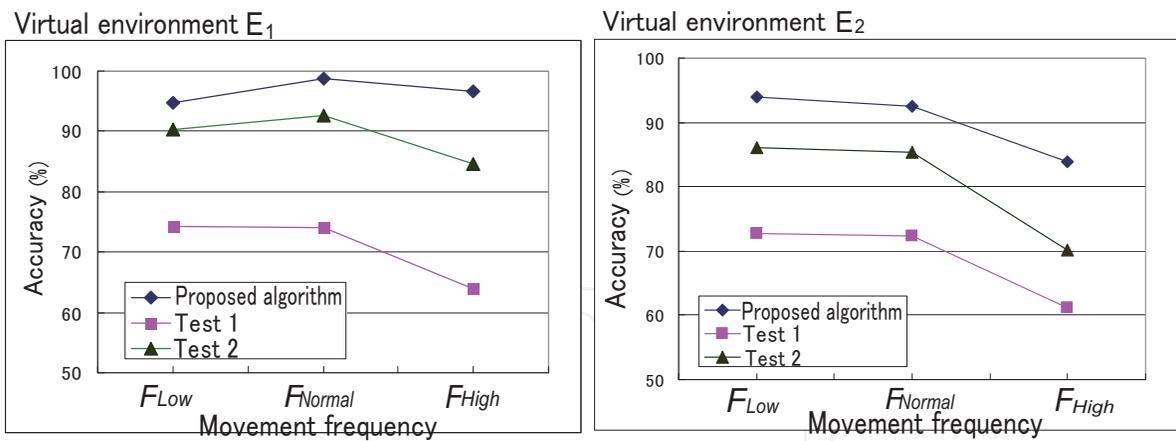


Fig. 6. Relations between  $F$  and accuracy.

steps using the proposed algorithm, Test 1, and Test 2. All the calculations were repeated ten times and the average accuracies were calculated (accuracy was the ratio of correct adjacent relationships acquired by calculation). The mean value of accuracies is plotted in Fig. 6 and the process for acquiring adjacent relationships using the proposed algorithm is shown in Fig. 7.

We next verified how robust our algorithm was against mis-reaction noise. We prepared six pattern environments by pairing  $E$  with  $M(F_{Normal}$  was used), and did experiments in the same way as for the environments using numerous people moving simultaneously. Figure 8 plots the results.

Figures 6 and 8 reveal that the proposed algorithm could analyze more accurately than Tests 1 and 2. In addition, our algorithm always required less sensor data to attain 90% accuracy than the others. Consequently, we concluded that it was more robust against noise data.

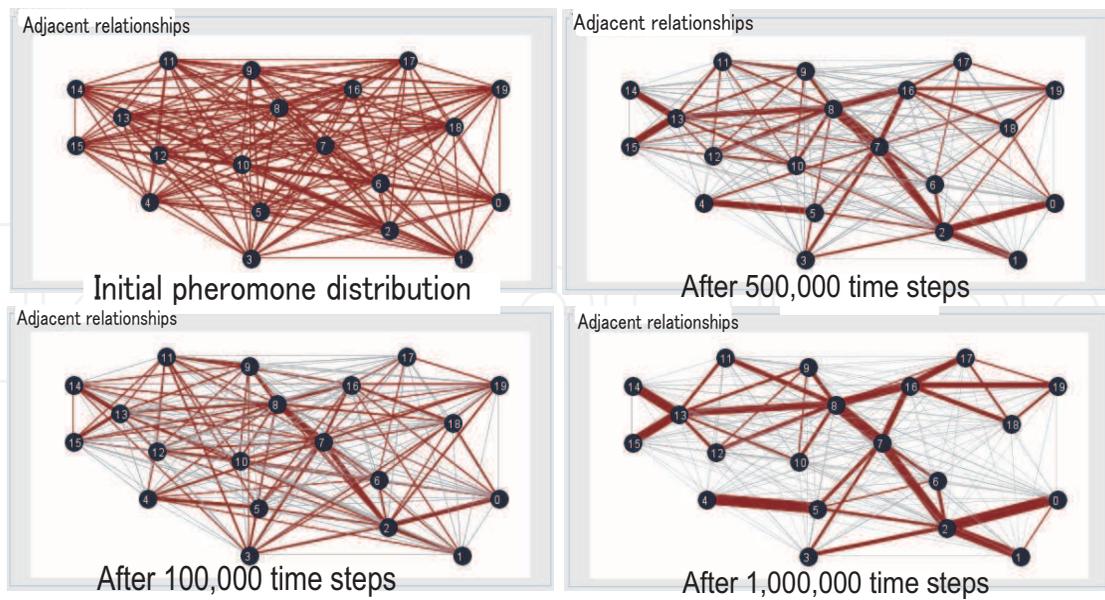


Fig. 7. Transition in acquiring adjacent relationships.

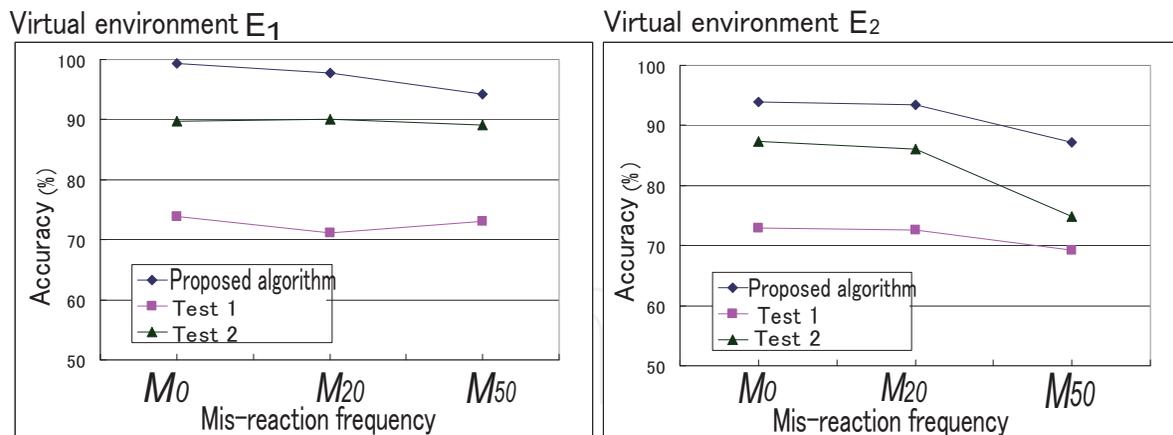


Fig. 8. Relations between  $M$  and accuracy.

#### 4.3 Verification of adaptability

Second, we verified how adaptable our algorithm was against dynamic changes in the topology of sensor-network systems. Structural changes are generated when one sensor fails and is replaced by two others.  $E_1$ ,  $F_{Normal}$  and  $M_{20}$  were selected for the simulation elements, and the network structure was shifted after 500,000 time steps had passed. The time to recover accuracy to 90% was taken as the barometer for adaptability to structural change. Figure 9 shows the process of adaptation to dynamic changes in the network structure.

As a result, the proposed algorithm needed about half the time of other algorithms not using the pheromone communication system to recover. Therefore, we concluded that it was basically adaptable to environmental changes.

#### 5. Empirical verification using real-world sensor data

This section describes our empirical verification using real-world sensor data. We used sensor-reaction data that were obtained from an infrared sensor network installed in our laboratory. A person's presence was detected by reading the reflection intensity from a sensor that had beamed an infrared laser from itself. The data were collected over a period of 30 days with 31 infrared sensors that were installed in three rooms. The proposed algorithm was

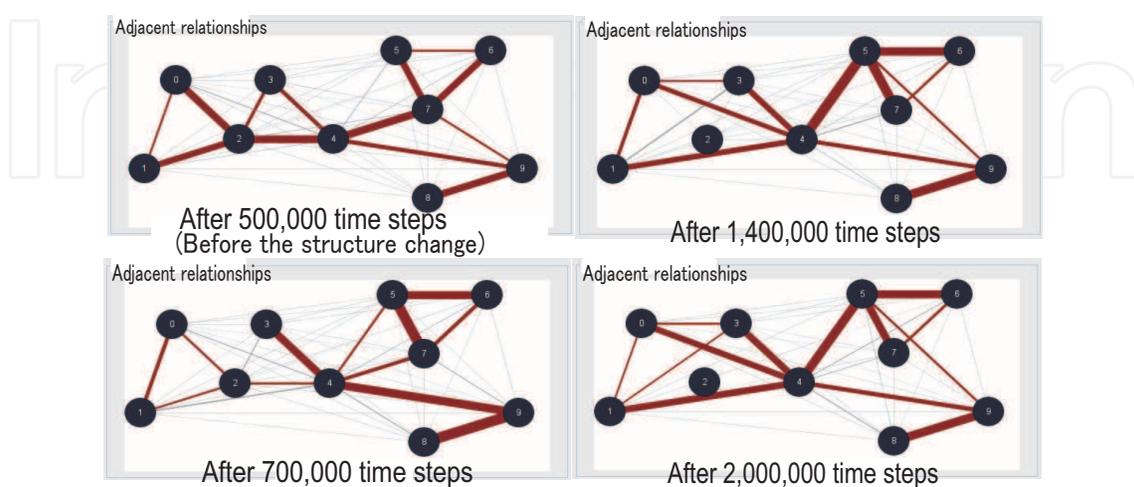


Fig. 9. Process of adaptation.

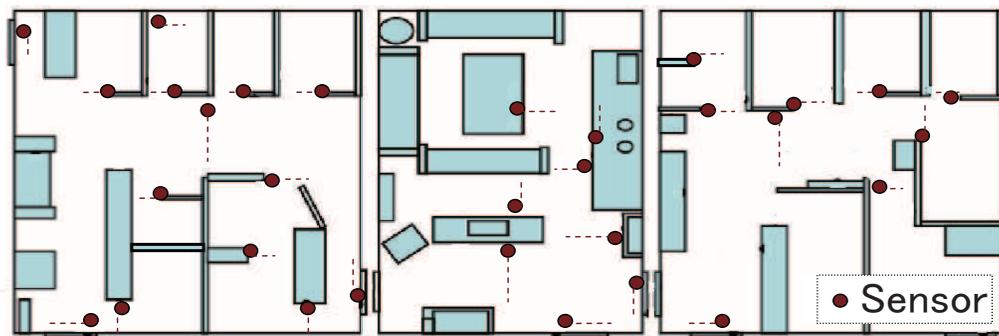


Fig. 10. Actual layout for sensor network.

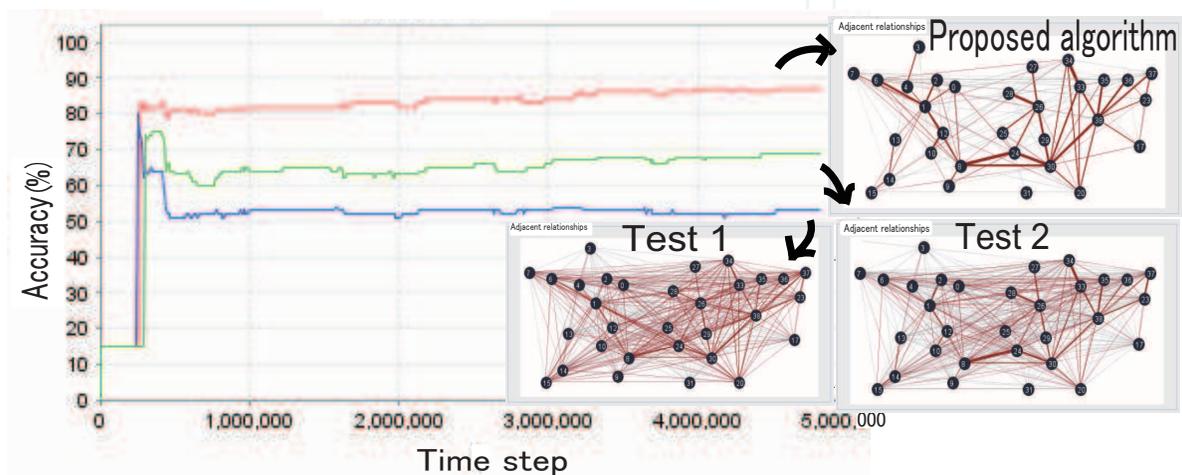


Fig. 11. Accuracy of analysis obtained by algorithms using real-world data and obtained adjacent relationships.

compared with Tests 1 and 2 that were discussed in Section 4. Figure 10 shows the layout for the sensor network.

The sensor data was collected for 24 hours, and each sensor had an average of 2143.2 reactions. The structure of the network also did not change during this period. Parameters  $T = 10,000$  and  $a = 20,000$  were used. Figures 11 shows the results discovered with the proposed algorithm, Test 1, and Test 2.

Figure 11 shows that the proposed method is superior to the test algorithms (Accuracies: 87% with proposed, 53% with Test 1, and 64% with Test 2). Note that it continues to remain accurate after approximately 300,000 time steps. Each data unit is created and analyzed every one and a half hours when converted into real-world time. There are large differences in the amount of sensor-reaction information obtained during the day, at night, or on holidays.

Moreover, when someone only makes one sensor react during their one episode of movement, this becomes noise in addition to movements by numerous people and sensor miss-reactions, because it is impossible to extract a sensor-reaction relation when only one sensor reacts. Since the proposed algorithm assumes that sensor-reaction information is the sequence of a person's movements, a single sensor reaction prevents accuracy from being improved. However, the new method also demonstrated greater robustness against this problem than the test algorithms.

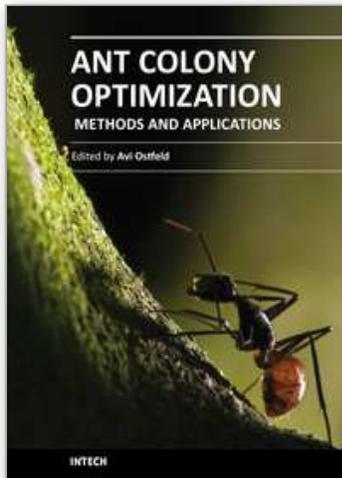
## 6. Conclusion

The basic effectiveness of the proposed methodology could be verified from our evaluations using simulations and experiments in the real world. We used our laboratory as the real environment for the experiments, where a total of 15 persons were working. The actual number of people at work differed depending on the day of the week and the time. The habitual behavior of individuals was of course different as they had their own weekly and daily schedules. The methodology we propose based on the pheromone-communication model could be adapted to the subjects' personal diversity, but unfortunately this might be difficult using the methodology suggested by Marinakis and Dudek. Our methodology has the following features:

- (1) A positive feedback loop is used to accelerate the process by which two sensors having a higher adjacent likelihood can be given a higher likelihood autocatalytically.
- (2) When an agent selects a node that it wants to move to, random selection is done at a constant rate. This mechanism, which is a heterogeneous factor, is very effective.
- (3) Also using other criteria, i.e., the standard traveling time,  $m$ , and not only the interval time of sensor reaction, which is the basic methodology that is used to consider whether two sensors are in an adjacent relation, very effectively improves accuracy and convergence.
- (4) Since pheromones gradually evaporate, old information is automatically removed, i.e., new information is always given priority. These features support all of the necessary functions discussed in Section 2, and demonstrate the effectiveness of the pheromone-communication model.

## 7. References

- [1] Marco Dorigo and Gianni Di Caro, *The Ant Colony Optimization Meta-Heuristic, New ideas in optimization* (McGraw-Hill), 1999, pp. 11–32.
- [2] Satoshi Kurihara, Shigemi Aoyagi, Toshihiro Takada, Toshio Hirotsu, and Toshiharu Sugawara, Agent-Based Human-Environment Interaction Framework for Ubiquitous Environment, INSS2005, pp. 103–108.
- [3] Seiichi Honda, Ken-ichi Fukui, Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao. Extracting Human Behaviors with Infrared Sensor Network, INSS 2007, pp. 122–125.
- [4] Dimitri Marinakis and Gregory Dudek, Topological Mapping through Distributed Passive Sensors, IJCAI 2007, pp. 2147–2152.
- [5] John A. Sauter, Robert Matthews, CH. Van Dyke, Parunak, and Sven A. Brueckner, Performance of Digital Pheromones for Swarming Vehicle Control, CAAMAS 2005, pp. 903–910.
- [6] Taisuke Hosokawa and Mineichi Kubo, Person tracking with infrared sensors, KES 2005, pp. 682–688.



## **Ant Colony Optimization - Methods and Applications**

Edited by Avi Ostfeld

ISBN 978-953-307-157-2

Hard cover, 342 pages

**Publisher** InTech

**Published online** 04, February, 2011

**Published in print edition** February, 2011

Ants communicate information by leaving pheromone tracks. A moving ant leaves, in varying quantities, some pheromone on the ground to mark its way. While an isolated ant moves essentially at random, an ant encountering a previously laid trail is able to detect it and decide with high probability to follow it, thus reinforcing the track with its own pheromone. The collective behavior that emerges is thus a positive feedback: where the more the ants following a track, the more attractive that track becomes for being followed; thus the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. This elementary ant's behavior inspired the development of ant colony optimization by Marco Dorigo in 1992, constructing a meta-heuristic stochastic combinatorial computational methodology belonging to a family of related meta-heuristic methods such as simulated annealing, Tabu search and genetic algorithms. This book covers in twenty chapters state of the art methods and applications of utilizing ant colony optimization algorithms. New methods and theory such as multi colony ant algorithm based upon a new pheromone arithmetic crossover and a repulsive operator, new findings on ant colony convergence, and a diversity of engineering and science applications from transportation, water resources, electrical and computer science disciplines are presented.

### **How to reference**

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

Satoshi Kuriharam, Hiroshi Tamaki, Kenichi Fukui and Masayuki Numao (2011). Adaptive Sensor-Network Topology Estimating Algorithm Based on the Ant Colony Optimization, *Ant Colony Optimization - Methods and Applications*, Avi Ostfeld (Ed.), ISBN: 978-953-307-157-2, InTech, Available from: <http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications/adaptive-sensor-network-topology-estimating-algorithm-based-on-the-ant-colony-optimization>

**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

### **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

[www.intechopen.com](http://www.intechopen.com)

[www.intechopen.com](http://www.intechopen.com)

IntechOpen

IntechOpen

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), 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