

Nanorobotic Agents Communication Using Bee-Inspired Swarm Intelligence

Rodney Mushining, Francis Joseph Ogwu

Computer Science Department, University of Botswana, Gaborone, Botswana
Email: rodneyushiningar@gmail.com, ogwufj@mopipi.ub.bw

Received August 12, 2013; revised September 12, 2013; accepted September 19, 2013

Copyright © 2013 Rodney Mushining, Francis Joseph Ogwu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

The main goal of this paper is to design nanorobotic agent communication mechanisms which would yield coordinated swarm behavior. Precisely we propose a bee-inspired swarm control algorithm that allows nanorobotic agents communication in order to converge at a specific target. In this paper, we present experiment to test convergence speed and quality in a simulated multi-agent deployment in an environment with a single target. This is done to measure whether the use of our algorithm or random guess improves efficiency in terms of convergence and quality. The results attained from the experiments indicated that the use of our algorithm enhance the coordinated movement of agents towards the target compared to random guess.

Keywords: Nanorobots; Agents; Nanites; Swarms; Nanotechnology; Beelike Nano-Machine

1. Introduction

The problem of controlling large groups of agents towards a specific target is not new. Genetic algorithms (GA) and other evolutionary programming techniques have been dominantly proposed for tackling this problem in the past [1]. In this manuscript, the main goal of this paper is to design nanorobotic agent communication mechanisms which would yield coordinated swarm behavior. This target is perceived to represent a foreign body in the human body like environment which the swarm is aimed at locating and destroying. We precisely investigate communication control issues in swarms of bee-like nanorobotic agents. Medical nanotechnology has promised a great future including improved medical sensors for diagnostics, augmentation of the immune system with medical nano-machines, rebuilding tissues from the bottom up and tackling the aging problem [2]. Proponents claim that the application of nanotechnology to nano-medicine, will offer ultimate benefit for human life and the society eliminating all common diseases and all medical suffering [3]. This work is primarily motivated by the need to contribute to this common goal. Most of the researches that have been completed so far focus mostly on the control of a single agent [4]. Increased efforts have begun towards addressing systems that are composed of multiple autonomous mobile robots [5]. In some cases, local in-

teraction rules maybe sensor based, as in the case of flocking birds [6]. In other cases these local interactions may be stigmergic [7]. In this work, the bee agents we propose have no leader to influence other bees in the swarm to fulfill their low level agenda. Each bee agent responds to some form of local information which is made available through the local environment and direct message passing.

2. Related Works

Communication is a necessity in multi agent emergent systems as it increases agent's performance. Many agent communication techniques assume an external communication method by which agents may share information with one another. Direct agent communication includes waggle dancing, a technique used by bees when they are communicating the source of food and signaling [8]. In our context a swarm can be defined as a collection of interacting nanorobotic agents. The agents will be deployed in an environment which is a substrate that facilitates the functionalities of an agent through both observable and unobservable properties. Given the above view swarm intelligence is the self organizing behavior of cellular robotic systems [9]. Swarm does not have a centralized control system and individual responds to simple and local information that allows the whole system to

function [10]. Bee agents in this research will be represented by nanorobotic agents, defined as artificial or biological nanoscale devices that can perform simple computations, sensing or actuation [11]. In this research, we will interchangeably use the terms nanorobots, nanites and nano-robotic agents to refer to tiny and autonomous devices that are built to work together towards achieving a collaborated solution, the same way natural bee swarms.

Foraging is a task of locating and acquiring resources and has to be performed in unknown and possibly dynamic environments [12]. When the foraging bee discovers the nectar source, they commit to memory information on direction in which the nectar is found, the distance from the hive and its quality rating [13]. On return to the hive they perform waggle dance on the dance flow [14]. The dance expresses the information of the direction, distance and quality of the nectar source. Onlooker bees assess the information delivered and make a decision to follow, remain in the hive or randomly wonder. The recruitment among bees is associated with the quality of the nectar source [15]. A nectar source with a lot of nectar and near the hive is recognized as more promising and would attract more followers [16]. In the event that the bees want to relocate due to some reasons, the queen bee and some of the bee in a colony leave their hive and form a cluster on a nearby branch [17]. Upon returning to the cluster the searching bees perform a waggle dance [18]. In a dance commonly known as waggle dance, agents can communicate the distance and direction of the target to observing novice agents [14]. The dance communication techniques in bee agents have been extended to applications in database querying, particularly evaluating the similarities between the recommender systems to the user query [19]. In [15], the travelling salesman problem was proposed. In their version a number of nodes dotted in the network and the hive was located at one of the nodes. To collect as much nectar, the artificial bee agents had to fly along a certain link and need to locate the shortest path to the source. The Bee Colony Optimization (BCO) was proposed to solve combinatorial optimization problems [20]. Two main elements of BCO is forwards pass and backwards pass. A partial solution was generated when the bees performed a forwards pass which was accomplished by the combination of individual exploration and collective experience for the past. A backward pass was performed when they returned to the hive. The step was followed by decision making process. It was inspired by a swarm of virtual bees where it began with bees wandering randomly in the search environment [14]. Virtual Bee Algorithm (VBA) initially created a population of virtual bees, where each bee was associated with a memory bank. Then, the functions of the optimization were converted into virtual

food source. The direction and distance of the virtual food were defined. Population update is through waggle dance.

This strategy combines recruitment and navigation [14]. Recruitment strategies were employed to communicate the search experience to the novice bees in a colony. Strategies included such thing as dancing processes, communicating information of distance and direction of the nectar source. The other half of the algorithm, navigation is mainly worried about discovering undiscovered areas. In [14], the non pheromone algorithms proposed by Lammens in his recruitment and navigation model was presented. The algorithm has three (3) functions namely ManageBeesActivities, CalculateVector and DaemonAction.

The models use mathematical model adopted from [21] which is inspired by the way bees forage in search of food. In addition to already discussed algorithms, we reviewed Minimal communication [22], Knowledge-based-multi-agent communication [23], sensing communication [24] and flocking [6]. As the agents move towards the target they communicate their proximity, direction and velocity. The kind of communication employed in these kinds of scenarios is direct communication and the agents will be communicating with the neighboring agents to maintain the same speed and distance between agents. In [25] Packer describes experiments of a group of simulated robots which are required to keeping a side by side line formation whilst moving towards a goal. This algorithm provides a one on group communication. The idea proposed in a minimal communication algorithm [22] has been improved in this algorithm and this will go a long way in solving our multi nanorobotic agents. This algorithm brings about the issue of proximity for communication to be more successful when they mention sense of sight.

3. Methodology

The agents must be able to assemble themselves to repair damaged vessels and it must be able to scan the environment in search of the target, In addition to what an agent can do us also need to consider the makeup of an agent that will enable it to execute the above mentioned tasks.

The structure can be represented as a class with the attributes.

3.1. Agent Class

```
{int x;
int y;
intdy, dx;
int state; // flagging, 1 , 2, .....
```

In this paper, the agents were deployed in an environ-

ment through a single entry point just like the way an injection is administered. The point of entry was defined and considered in terms of the y and the x axis. As soon as our nanorobotic agents are deployed into the environment they start communicating among themselves and their environment. The goal for communicating dispersion in nanorobotic agents is to achieve positional configurations that satisfy some user defined criteria. This behavior steers a nanorobotic agent to move to avoid crowding its local flocking mates.

Algorithm

If evaluate (pos) == unsatisfied then

Move

Else Do not move

End if

This algorithm can be diagrammatically represented as follows (See **Figure 1**).

From **Figure 1**, unsatisfied situations can mean that nanites are too closed or too far apart from each other while satisfied situation should be the normal flocking position.

The goal of each nanorobotic agent is to achieve and maintain a constant minimum and maximum distance from its neighboring nanorobotic agents. To enhance swarming, we provided an algorithm which makes all nanites to come closer to each other to maintain a swarm. We use the intuitive idea where each nanites move towards the centre of mass (COM) of all other nanorobotic agents where COM of n points $p^1 \dots\dots p^n$ is defined by

$$\bar{p} = \frac{1}{n} \sum_i^n p_i \tag{1}$$

3.2. Algorithm

If $|\Delta x| < 1/2$ and $|\Delta y| < 1/2$ then

Do not move

Else

Choose dimension $d \in \{x, y\}$ for each $|\Delta d| \geq 1/2$

Move one step towards the COM.

And representation diagrammatically as in **Figure 2**.

In **Figure 2**, the black dots are nanorobotic agents, the tiny circle is COM and the arrows signify possible movement choices for each moving nanorobotic agent. Our nanorobotic agents will move towards the average destination neighbors keeping the swarm in alignment and moving together towards the same general heading. To align them the nanorobotic agents will need to

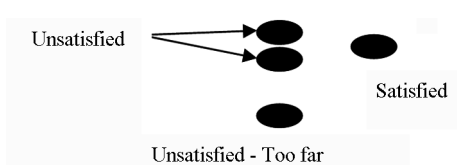


Figure 1. An example of the dispersion algorithm in action.

communicate velocity of the neighboring agent and adjust its speed to suit the rest of the swarm. The main and final aspect of this research as mentioned earlier is to communicate the location of the target. As soon as one of the nanorobotic agents get in contact with the target in the environment it will change color and make some movements (waggle) observed by those within the same proximity. Agents within the proximity will also change color to red to show that they have received the message and they will move towards the target. In our simulator all user options are dynamic and may be switched during execution. Target deployment: The user can only deploy a sing target anywhere in the environment by assigning values for x and y axis in the target information panel. The agents in our simulator can be deployed in batches, starting form 2, 5, 10, 20 up to 50. Agent deployments: Agents enter the environment through a single entry point as defined by the user. The user will also enter the point of entry through accepting the x and the y axis value. Agent speed control: Speed of the nanorobotic agents can either be decreased on increased through a button under the agent information panel.

Figure 3 shows the nanorobots swarming towards the target at closed formation. Here, the screen is the environment while the control panel is the agent and target information panel. Our approach, we kept a

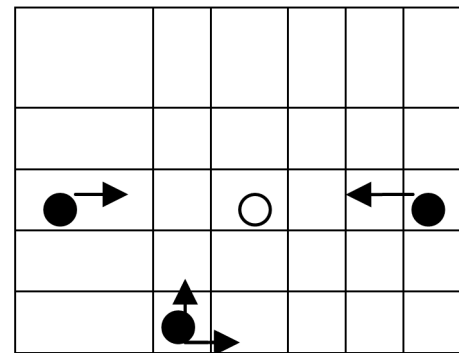


Figure 2. Cohesion diagrammatic representation.

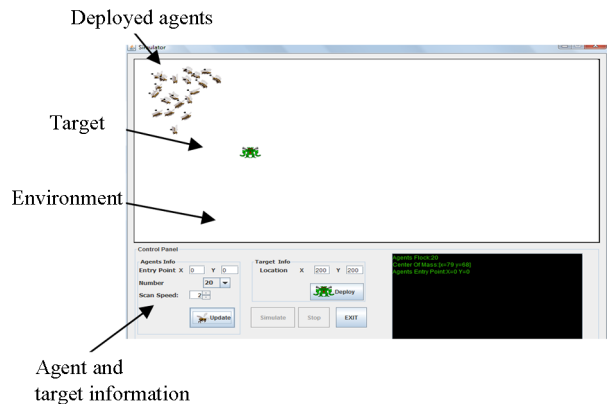


Figure 3. Simulator interface.

closer look on and avoid the following as our nanorobots will be sending a message to its neighbors; send wrong message or fail to send message. In our approach our nanorobotic agents are capable of observing and decoding message being sent by the wagging nanorobotic agent. After observation our agents will select the best waggle dance considering distance, direction and target location. We logically present our algorithm as follows:

- a) Deploy target
- b) Deploy n agents
- c) Explore environment
- d) Communicate swarm direction
- e) Cohesion
- f) Agent collision avoidance
- g) If found = null: go back c.
- h) Communicate target location
- i) Coordinate movement towards communicated location.

4. Simulation Scenarios

The simulation environment should provide an accurate estimate of nanorobotic agent performance in the real human like environment. The potential metrics for nanorobotic agent's communication and coordination are convergence speed and quality. Convergence speed was measured in terms of time and distance. Time was measured in iterations. Quality was measured in terms of frequency and time. In order to examine the performance of the proposed communication control algorithm we first consider the efficiency of the following experiments: Dispersion, Direction coordination, Cohesion control, and Target location. If the agents collide it may take a certain delay time to reassemble and that affect systems efficiency in performance. In the event that the agents are about to collide the separation algorithm will be called to disperse the near collision agents. We started by deploying 2 agents in the environment to run at a speed 0.4. In the screen shot given above 10 nanorobotic agents have been deployed at position 0, 0 in terms of x and y axis. We observed the agents and noted that they are moving in a single coordinated direction. As the agents will be scanning the environment each nanorobotic agents need information about the speed of neighboring agent.

5. Results and Discussion

The results obtained from the simulation are presented below.

Convergence speed: In this experiment two simulation setups have been chosen to compare and study performance and scalability of the proposed algorithm in terms of convergence speed. The setups were carried out in the same environment where number of agents was changed

with the target remaining at the same position. The first simulation setup was made for reference case model followed by the coordinated simulation model.

5.1. Parameter Justifications

As shown in **Table 1**, we used three batches of 5, 10, and 20 agents for both reference case model and coordinated agents model in order to ascertain the results in the event of doubled agents' deployment. Simulation time has to be set such that we have the maximum limit to avoid running the simulation to infinite. The environment has been given as 884×418 following the measure of the canvas space given when we developed the simulator. The target was deployed at the midpoint for it not too far or too close to the boundary to enable more observation time. Agent deployment point was kept constant at point 0, 0 to enable consistent results. **Figure 4** given in the following shows summarized results obtained after running some experimental simulation where agents are not capacitated with any control algorithm. The results shows that using random guess, the agents were able to locate the target after so many iterations. For testing purposes we ran the simulation three times with the same agent density in order ascertain how varied the results would be. The non-effectiveness of reference case model has shown high iteration values implying that there is low convergence speed.

An effective "hit" occurs when 50% on the nanorobotic agents find the target. The convergence properties were measured by running the model for three times on a group of 5, 10 and 20 respectively. In all runs, the reference model achieved reaching the target after high iteration values. To evaluate the effectiveness of our algorithm in terms of convergence speed, we made several simulation runs. We fixed the target at mid point as justified in simulations above and changed the number of agents from 5, 10 and 20 respectively. The results indicate that the agents were able to locate the target after far less iteration as compared to reference case model. **Figure 5** is the graphical representation for the results obtained.

Table 1. Parameters for random guess.

Description	Values
Number of agents	5, 10, 20
Simulation duration (time step)	100
Number of trials	N
Environment	884×418
Target position in terms of (X, Y)	Varied
Agents entry point	442,209

5.2. Performance Measures of Our Agent Control Algorithm

We have successfully deployed the agents in the environment in groups of 5, 10 and 20 respectively. We observed that the agents within the proximity will change color to red and move to the target instantly decreasing the iterations towards the target. Agents controlled by our algorithm achieved getting to the target in less iteration. We also observed that the more the agents the high the convergence speed though the variance is not that distinct. In this research we drew up analogies between agent density and convergence speed. We have observed that the two are greatly related in that an increase in number of agent causes an increase in convergence speed. Another observation made is that as soon as one of the agents get informed all those agents within proximity get the information about the target location and they will immediately move towards the target meaning as the number increases the rate of confidence and convergence is increased. During the simulation some initially non informed agents had the opportunity to make new observation that converts them into informed agents. We recommend the deployment of many coordinated agents to enhance high convergence speed when using our algorithm. Not mentioning other factor the major or basic concept that enables high convergence speed is that we allow non informed agents to acquire informed state through interactions with neighbors. This will result in

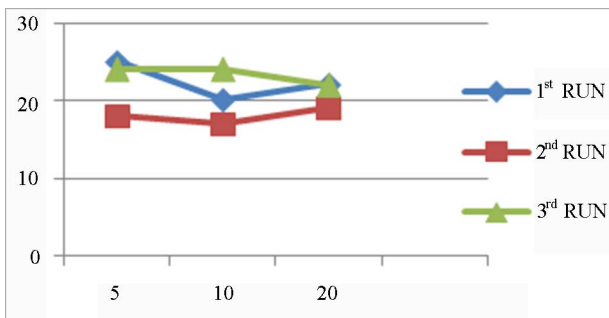


Figure 4. Reference case model graph.

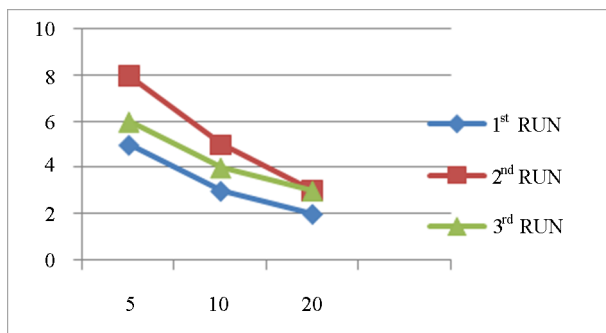


Figure 5. Results given different target position and varied number of agents.

increasing in the number of informed agents hence high convergence speed. Random guess results have shown no consistence in terms of number of iterations. Conclusively we can safely say that convergence speed increases when agents are coordinated as compared to random guess and the number of agents takes a pivotal role convergence speed. Our algorithm allows all agents to follow dispersion, alignment, and cohesion rules to reach common decision in search of a target without complex coordination mechanism.

Figure 6 shows coordinated agents in action searching for a target. In the screen shot it shows that as soon as one of the agents get in touch with the target it communicates the information about the location of the target hence move directly towards the target increasing the frequency of “hitting” the target. Ultimately, this will increase the quality of emergence when using our algorithm. To authenticate our argument Table 2 shows the results obtained after a number of simulations to test for the quality when using our control algorithm.

Figure 7 given below shows the corresponding results represented graphically.

We observed that the agents within the proximity will change color to red and move to the target instantly. Agents controlled by our algorithm achieved high quality due to target location communication.

6. Conclusions

The bee agent control algorithm generally achieves better convergence speeds and qualities of emergence than the reference case model. Performances in both cases improve with an increase in agent density. Frequency of hitting the target increases as the number of informed agents increases in our algorithm compared to reference

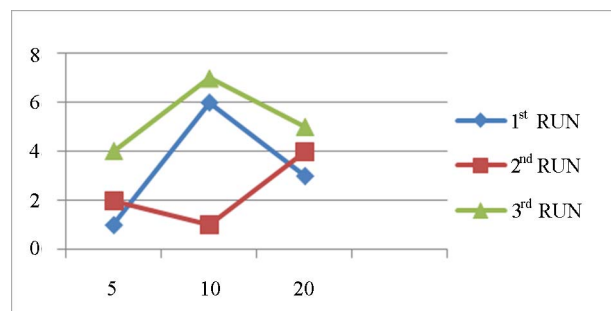


Figure 6. |Random wondering.

Table 2. Coordinated agent results.

	1 st run	2 nd run	3 rd run
5	18	16	15
10	16	19	16
20	17	20	18

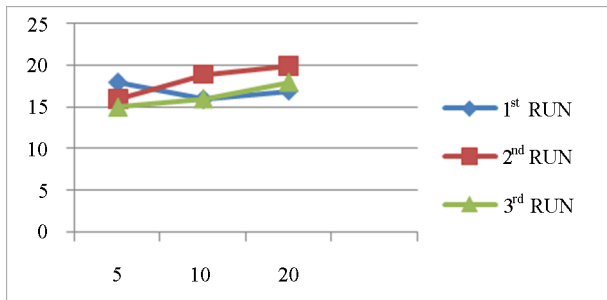


Figure 7. Graphical results for coordinated agents.

case model. Fast and coordinated swarm decisions making is prominent in bee agent control algorithm compared to reference case model.

The results presented are a proof of concept that the bee agent algorithm we propose can successfully coordinate agents towards desired targets in specific environments. As such, we contribute evidence of the potentials of similar algorithms to enhance search in human-body-like environments. In that way, connotations towards using the algorithm to control nanorobotic agents for health purposes are supported. A number of conclusions emanated from the results are reported in this work. Among these are: Swarms of bee-like agent that are deployed and controlled using the algorithm proposed in this work significantly out-performed the performances of swarms of random wandering agents that are deployed for the same purpose. We attribute these good results to a number of factors that are addressed by our algorithm. Most importantly our algorithm incorporate mechanisms in which the agents of the deployed swarm make use of local interactions and information to decide the direction to follow in each step. This alone fosters the speed of convergence, and hence the quality of emergence behavior that arises.

REFERENCES

- [1] A. Cavalcanti and R. A. Freitas, Jr., "Nanorobotics Control Design: A Collective Behaviour for Approach for Medicine," *IEEE Transactions on Nanobioscience*, Vol. 4, No. 2, 2005, pp. 134-139.
- [2] A. Cavalcanti, R. A. Freitas Jr. and L. C. Krety, "Nanorobotics Control Design: A Practical Approach Tutorial," ASME Press, Salt Lake City, 2004.
- [3] R. A. Freitas Jr., "Current Status of Nanomedicine and Medical Nanorobotics," *Computational and Theoretical Nanoscience*, Vol. 2, No. 1, 2005, pp. 1-25.
- [4] A. Galstyan and T. Hogg, "Kristina Lerman Modelling and Mathematical Analysis of Swarms of Microscopic Robots," Pasadena, 2006, pp. 201-208.
- [5] T. E. Izquierdo, "Collective Intelligence in Multi Agent Robotics: Stigmergy, Self Organisation and Evolution," University of Sussex Press, UK, 2004.
- [6] C. W. Reynolds, "Flocks, Heads and Schools: A Distributed Behavioural Model," *Computer Graphics*, Vol. 21, No. 4, 1987, pp. 25-34.
- [7] C. Chibaya and S. Bangay, "A Probabilistic Movement Model for Shortest Path Formation in Virtual Ant like Agents," *Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, South Africa, 2007, pp. 9-18.
- [8] N. Pavol, "Bee Hive Metaphor for Web Search," *Communication and Cognition-Artificial Intelligence*, Vol. 23, No. 1-4, 2006, pp. 15-20.
- [9] G. Beni and J. Wang, "Swarm Intelligence in Cellular Systems," *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, 26-30 June 1989.
- [10] D. M. Gordon, "The Organisation of Work in Social Insect Colonies," *Nature*, Nature Publishing Group, Vol. 380, No. 6570, 1996, pp. 121-124.
- [11] R. A. Freitas Jr., "Current Status of Nanomedicine and Medical Nanorobotics," American Scientific Publisher, USA, 2005.
- [12] J. R. Krebs and D. W. Stephen, "Foraging Theory," Monographs in Behavior and Ecology, Princeton University Press, Princeton, 1986.
- [13] K. V. Frisch, "The Dancing Bees: Their Vision, Chemical Senses and Language," Cornell University Press, Ithaca, 1956, pp. 36-39.
- [14] Lammens, S. Dejong, K. Tuyls, A. Nowe, *et al.*, "Bee Behaviour in Multi-Agent Systems: A Bee Foraging Algorithm," *Proceedings of the 7th ALAMAS Symposium*, Maastricht, 2007.
- [15] D. Teodorovic, "Swarm Intelligence Systems for Transportation Engineering: Principles and Applications," Elsevier, Vol. 16, No. 6, 2008, pp. 651-667.
- [16] M. Beekman and F. L. W. Ratniek, "Long Range Foraging by the Honey Bee," *Apis Mellifera*, Vol. 14, No. 4, 2000, pp. 490-496.
- [17] K. M. Passino and T. D. Seeley, "Modelling and Analysis of Nest-Site Selection by Honeybee Swarms: The Speed and Accuracy Trade-Off," Springer-Verlag, Berlin, 2006, pp. 427-442.
- [18] T. D. Seeley and S. C. Buhraman, "Group Decision Making in Swarms of Honey Bees," *Behavioral Ecology and Sociobiology*, Vol. 45, No. 1, 1999, pp. 19-31.
- [19] F. Lorenzi, F. Arreguy, C. Correa, A. L. C. Bazzan, M. Abel, F. Ricci, *et al.*, "A Multi Agent Recommender System with Task Based Agent Specialisation," Springer, Berlin, 2010.
- [20] C. S. Chong, A. I. Sivakumar, M. Y. H. Low and K. L. Gay, "A Bee Colony Optimisation Algorithm to Job Shop Scheduling," *Winter Simulation Conference*, California, 3-6 December 2006, pp. 1954-1961.
- [21] N. Pavol, "Bee Hive Metaphor for Web Search," *Communication and Cognition—Artificial Intelligence*, Vol. 23, 2006, pp. 15-20.
- [22] G. Enee and C. Escabut, "A Minimal Model of Commu-

- nication for Multi-Agent Classifiers System,” *4th International Workshop on Advances in Learning Classifier Systems*, Berlin-Heidelberg, 2002, pp. 32-42.
- [23] E. Verbrugge and R. van Baars, “Knowledge Based Algorithm for Multi-Agents Communication,” *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory*, 2006, pp. 227-236.
- [24] T. O. T. Kuroda and Y. Hoshinot, “A Evaluation of Multi Agent Behaviour in a Sensing Communication World,” *IEEE Internation Workshop on Robot and Human Communication, Conference Proceeding*, Tokyo, 3-5 November 1993, pp. 302-307.
- [25] G. A. Bekey, “From Biological Inspiration to Implementation and Control,” 1993.