https://doi.org/10.20965/jaciii.2007.p1099

Paper:

Research on the Sheepdog Problem Using Cellular Automata

Yoshinobu Adachi and Masayoshi Kakikura

Tokyo Denki University, 2-2 Nishiki-cho, Kanda, Chiyoda-ku, Tokyo 101-8457, Japan
E-mail: {adachi, kakikura}@isl.d.dendai.ac.jp
[Received February 28, 2007; accepted June 14, 2007]

The simulation framework we propose for complex path planning problems with multiagent systems focuses on the sheepdog problem for handling distributed autonomous robot systems – an extension of the pursuit problem for handling one prey robot and multiple predator robot. The sheepdog problem involves a more complex issue in which multiple dog robot chase and herd multiple sheep robot. We use the Boids model and cellular automata to model sheep flocking and chase and herd behavior for dog robots. We conduct experiments using a Sheepdog problem simulator and study cooperative behavior.

Keywords: cellular automata, pursuit problem, multiple mobile robots, cooperative behavior

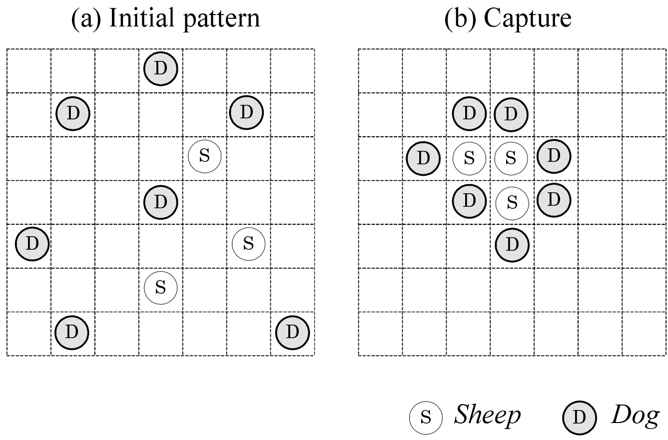

Fig. 1. Definition of sheepdog problem. (a) Initial pattern, (b) Capture.

1. Introduction

Technological improvements in robot hardware and related software application are increasing mobile robot utilization, making solutions for intelligent and efficient control of multiple mobile robot systems very hot issues. Simulation study and development for such systems is particularly relevant and important at this stage. The key issue is cooperative behavior between distributed autonomous robot systems.

The problem of cooperation between multiple robot systems is a subject of distributed artificial intelligence (DAI), which covers a vast, impressive number of studies [6, 7, 9–12]. Examples include cooperative search, hunting, and capture problems. Cellular automata are used to model DAI problems and as a powerful problem solver for multiagent systems [3–5, 8]. The approach is applied to model the complex behavior of particle elements such as gases, liquid materials, and biological cells.

The Boids model [1, 2] is another very practical, useful tool used to model cooperative movement such as birds flocking, swarming, fish schooling, and other distributed behavior.

In our work, we use boids and cellular automata to model cooperative multiple mobile robot behavior in solving the sheepdog problem. We present the definitions underlying our work in the next section.

2. Problem Definition

The sheepdog problem is an extension of the pursuit problem that handles a single prey robot and multiple predator robots. In the sheepdog problem, multiple agents (dog robots) seek to herd multiple agents (sheep robots), each agent moving vertically or horizontally on a two-dimensional (2D) lattice shown in **Fig. 1**.

The field defined by the 2D lattice is an infinite toroidal plane. Robot moving to the right for enough over the right edge of the lattice appears at the opposite side up from the left edge.

Sheep completely surrounded by dogs are captured. The number of sheep and dogs ranges from 0 to 1000 and the filed size is 100×100 grids.

The sheepdog problem requires two different cooperative behavioral models: First, a flocking model for sheep. Second, a chase and herd (capture) model for dogs.

Robots detect the presence of neighbor robots within a specific range of sight, recognizing the robot type. Robots moves individually based on local information and neighborhood robot interaction.

Performance is measured by the number or ratio of captured sheep. The ratio is the proportion of captured to total number of sheep.

© Fuji Technology Press Ltd. Creative Commons CC BY-ND: This is an Open Access article distributed under the terms of the Creative Commons Attribution-NoDerivatives 4.0 International License (http://creativecommons.org/licenses/by-nd/4.0/).

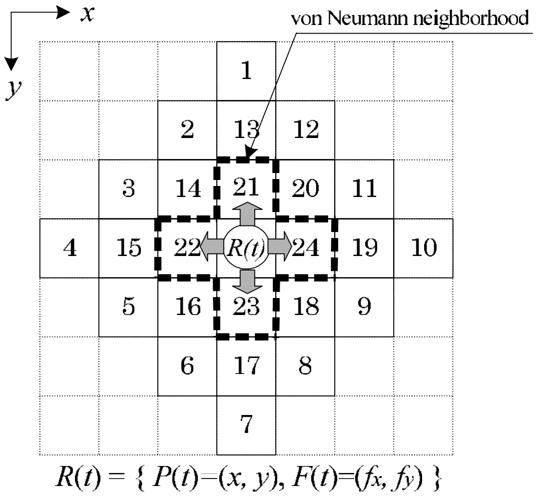

Fig. 2. Robot range of sight with three Manhattan distance.

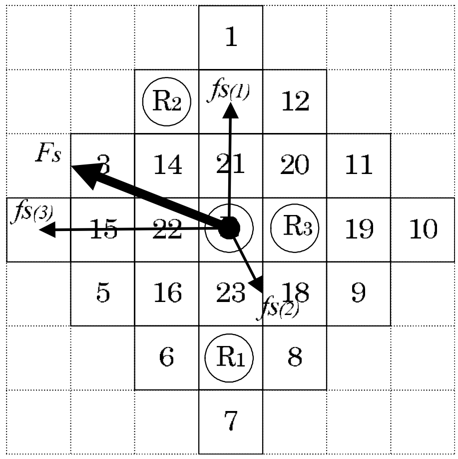

Fig. 3. Separation force example with three sheep robots.

3. Robot Models

3.1. Boids and Cellular Automata

To simulate the sheepdog problem, we use the Boids algorithm and cellular automata to model sheep and dog robot behaviors, proposing a simulation framework based on a Boids algorithm within 2D cellular automata.

The basic Boids flocking model consists of three types of simple steering forces (separation, cohesion, and alignment) describing how individual robots maneuver based on the location and speed of neighborhood robots:

1. *Separation*: The boid avoids collisions with neighborhood boids.

2. *Cohesion*: The boid moves toward neighborhood boids.

3. *Alignment*: The boid steers itself in the same direction as neighborhood boids.

To implement the sheepdog problem simulator, we propose the following framework for controlling the sheep and sheepdog behavior with cellular automata:

- *Robot activity*: Robots move within a 2D lattice of n × n grids in a toroidal world.

- *Robot range of sight*: Robot sensing detects the location of other robots, the robot type, and relative speed within a certain 24-cell range of sight (**Fig. 2**). The robot detects other sheep and dog robots within the range of sight.

- *Robot action rules*: A robot in the lattice field has two parameters (Eq. (1)). $R(t)$ represents status information of the robot at step time t. Parameter 1 is the *x*-*y* location in lattice as $P(t)$. Parameter 2 is the force that affects robot movement, $F(t)$.

- *Robot Movement*: The robot takes a step forward in the von Neumann neighborhood at step time $t+1$. The von Neumann neighborhood consists of four direct neighbors, i.e., directly to the right, directly to the left, directly above, and directly below the robot. The cell of next step $P(t+1)$ is decided by force vector $F(t)$ of the robot (Eq. (4)). Function $rand(n)$ returns a random number between 0 and n. If, for example, force vector $F(t)$ equals $(4,1)$, the robot moves one cell to the right with a probability of 80%, or one cell down with a probability of 20%.

Robot status is defined as in Eq. (1):

$$R(t) = \{P(t) = (x,y), F(t) = (f_x, f_y)\}. \quad \ldots \quad (1)$$

Force at step time $t+1$ is defined as in Eqs. (2) and (3):

$$F(t+1) = F_s(t) + F_d(t) + F(t) \quad \ldots\ldots \quad (2)$$

where $F_s(t)$ is the force affected by the same type of robot and $F_d(t)$ is the force affected by a different type of robot.

$$\begin{aligned} F_s(t) &= a_s F_{s_{sep}}(t) + b_s F_{s_{coh}}(t) + c_s F_{s_{ali}}(t) \\ F_d(t) &= a_d F_{d_{sep}}(t) + b_d F_{d_{coh}}(t) + c_d F_{d_{ali}}(t) \end{aligned} \quad . \quad (3)$$

where $F_{s_{sep}}$, $F_{s_{coh}}$, and $F_{s_{ali}}$ are separation, cohesion, and alignment forces using the same type of robot. Parameters a_s, b_s, and c_s are the weights of each force. $F_{d_{sep}}$, $F_{d_{coh}}$, and $F_{d_{ali}}$ are separation, cohesion, and alignment forces using different type of robots. a_d, b_d, and c_d are the weights of each force.

$$P(t+1) = \begin{cases} (x + sign(f_x), y) & when \quad d = 1 \\ (x, y + sign(f_y)) & when \quad d = 0 \end{cases}$$

$$\begin{aligned} rand(|f_x| + |f_y|) &\leq |f_x| \rightarrow d = 1 \\ rand(|f_x| + |f_y|) &> |f_x| \rightarrow d = 0. \end{aligned} \quad (4)$$

Separation force at step in time t is defined by

$$F_s(t) = \sum_i f_s(i)$$

where $f_s(i)$ is separation force elements affected by robots, and i is the index of robots within the range of sight (**Fig. 3**).

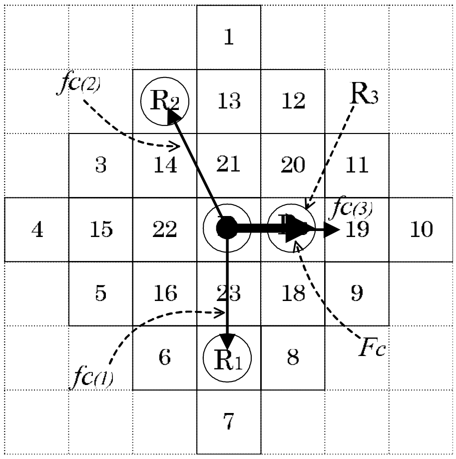

Fig. 4. Cohesion force example with three sheep robots.

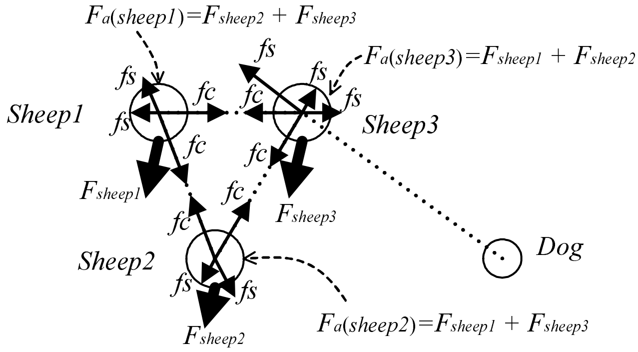

Fig. 5. Example of relationship among three sheep robots and one dog robot.

Cohesion force at step in time t is defined by

$$F_c(t) = \sum_i f_c(i)$$

where $f_c(i)$ is cohesion force elements affected by robots, and i is the index of robots within the range of sight (**Fig. 4**).

Alignment force at step in time t is defined by

$$F_a = \sum_i F(i)$$

where $F(i)$ is one of force elements owned by a neighborhood robot, and i is the index of robots within the range of sight. Force F_a acts on the robot to steer it towards the average heading of other robots within the range of sight.

3.2. Sheep Robot Model

To implement sheep robot behavior, we set the following action rules for sheep robot: First, sheep robots execute the simulated flocking. Second, sheep robots attempt to avoid colliding with dog robots. Sheep robot action rules based on $R(t)$ defined in the previous section, consist of the following forces:

1. *Separation*: The separation force $f_s(i)$ element is generated by a sheep robot or dog robot within the range of sight.
2. *Cohesion*: The cohesion force $f_c(i)$ element is generated by a sheep robot within the range of sight.
3. *Alignment*: The alignment force $F(i)$ element is generated by a sheep robot within the range of sight.

In the relationship between three sheep robots and one dog robot (**Fig. 5**), *Sheep1*, *Sheep2*, and *Sheep3* are within a mutual range of sight, and *Dog* is in the range of sight of *Sheep3*. Separation, cohesion, and alignment forces are shown as f_s , f_c, and F_a.

As cellular automata run, sheep robots flock with other neighborhood sheep robots while simultaneously attempting to avoid colliding with the dog robots.

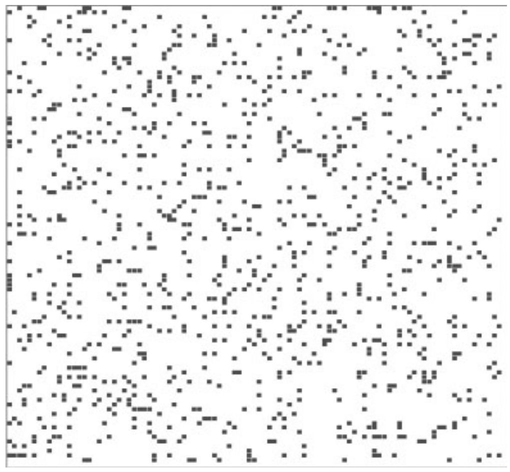

Fig. 6. Initial sheep robots pattern snapshot.

Figure 6 shows the snapshot of an initial sheep robots pattern in the sheepdog problem simulator. Grids are 100×100, and sheep robots number is 1000. Robots are placed randomly on the lattice.

Figure 7 shows the snapshot of cellular automata evolved by sheep robot action rules at step 500. Note the flocking sheep robots. After approximately 1000 steps, sheep robots are still observed flocking, while simultaneously tending to proceed in the same direction randomly chosen at each cellular automata execution.

3.3. Dog Robot Model

To implement dog robot behavior, we set the following action rules for dog robots: First, dog robots have no separation force in relation to sheep robots. Second, dog robots develop strong cohesion in relation to sheep robots. Third, dog robots develop strong alignment force in relation to sheep robots.

In short, dog robots chase sheep robots via cohesion and alignment forces, while dog robots simultaneously drawn together by cohesion force, simulating collaborative sheepdog.

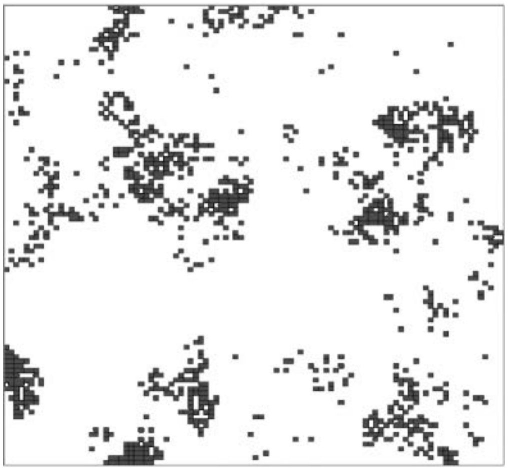

Fig. 7. Example of sheep robots flocking.

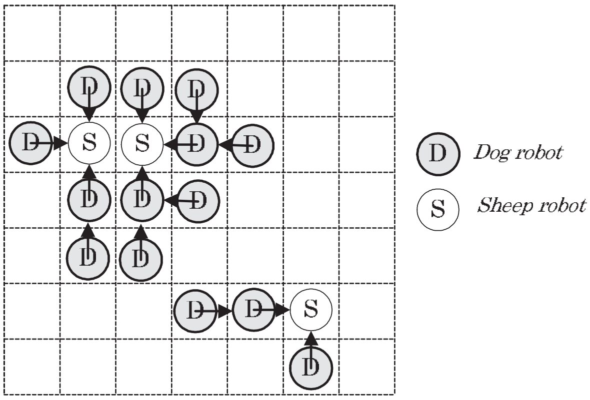

Fig. 8. Overview of dog robot chasing and herding sheep robots, and dog robots cohesion.

Figure 8 shows an overview of dog robot characteristics. The dog robots chase and capture sheep robots, in simultaneous cohesive action.

Dog robot action rules based on $R(t)$ defined in the previous section, consist of the following forces:

1. *Separation*: The separation force $f_s(i)$ element is generated by a dog robot within the range of sight.

2. *Cohesion*: The cohesion force $f_c(i)$ element is generated by a sheep or a dog robot within the range of sight.

3. *Alignment*: The alignment force $F(i)$ element is generated by a sheep or a dog robot with in the range of sight.

In the relationship between two dog robots and one sheep robot (**Fig. 9**), *Dog1*, *Dog2*, and *Sheep1* are within a mutual range of sight. Separation, cohesion, and alignment forces are shown as f_s , f_c, and F_a.

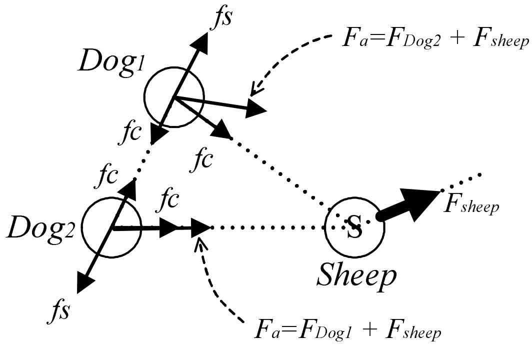

Fig. 9. Example of relationship among two dog robots and one sheep robot.

Table 1. Number of sheep robots for simulation.

Case	Number of sheep robots	Total number of robots
1	250	1250
2	500	1500
3	750	1750
4	1000	2000

4. Experiments

4.1. Sheepdog Problem Simulator

The simulator we developed in a Java environment to simulate the sheepdog problem consists of three components: The first, a cellular automata engine, contains sheep and dog behavior models. The second, a user interface, controls the number of sheep and dog robots, the field size, and other simulation parameters. The third, a utility module, imports and exports initial robot location patterns and implements the data logger for simulation results.

4.2. Robot Density Simulation

4.2.1. Robot Parameters

Experiment parameters are as follows: The field size is 100×100 grids. The number of dog robots is 1000. Four different numbers of sheep robots, 250, 500, 750, and 1000 were chosen to compare sheepdog simulation performance (**Table 1**).

Robot force weighting parameters in Eqs. (3) and (4) are as follows:

- **Sheep robots**: $a_s = 4$, $b_s = 1$, $c_s = 2$, $a_d = 10$, $b_d = 0$, and $c_d = 0$.

- **Dog robots**: $a_s = 1$, $b_s = 0$, $c_s = 1$, $a_d = 0$, $b_d = 40$, and $c_d = 5$.

The ratio of captured sheep robots is calculated by averaging the results of ten times of the simulations. The ratio

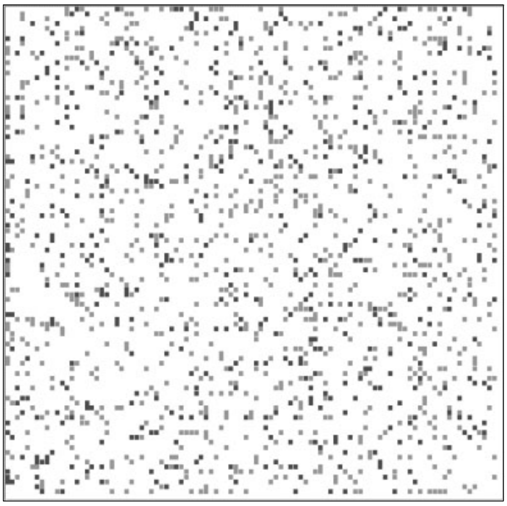

Fig. 10. Initial pattern snapshot: The number of sheep robots 500, the number of dog robots 1000, and the field size is 100×100 grids. Dark dots are sheep robots, and grey dots are dog robots.

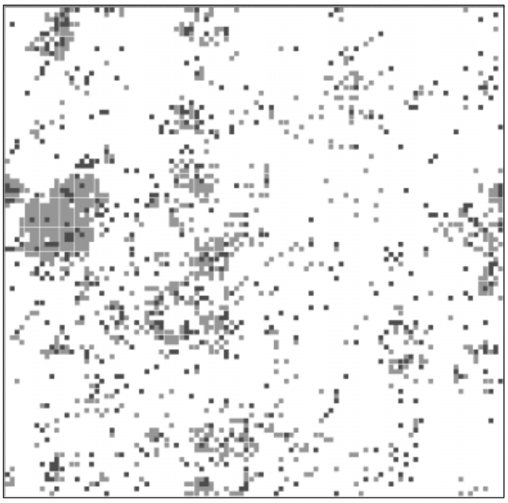

Fig. 11. Snapshot of sheepdog cellular automata field at step 2500.

is defined by the proportion between the number of captured sheep robots and the total number of sheep robots. The initial sheep and dog robots patterns are the same in each simulation (**Fig. 10**).

Figure 11 shows how sheepdog cellular automata have progressed at step 2500. Dog robots chase sheep robots, forming small clusters, and the number of captured sheep robots is low.

At step 25000, dog robots have captured sheep robots, forming a cluster (**Fig. 12**).

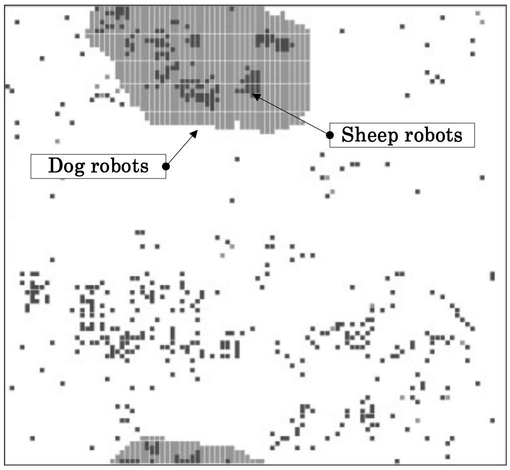

Fig. 12. Example: Dog robots cluster to herd sheep robots. Snapshot at step 25000.

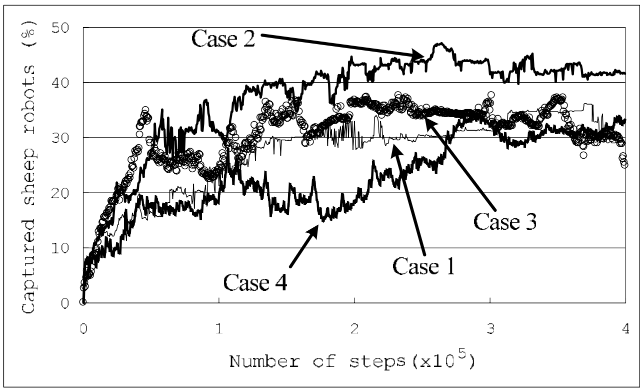

Fig. 13. Ratio of captured sheep robots: In case 1, sheep robots number 250, in case 2; 500, in case 3; 750, and in case 4; 1000.

4.2.2. Simulation Results

In simulation results (**Fig. 13**), the x axis is the number of steps over time and the y axis is the ratio of captured sheep robots. In case 1, sheep robots number is 250, in case 2; 500, in case 3; 750, and in case 4; 1000. Dog robots number is 1000. Captured sheep ratio $r(t)$ at step in time t is defined by

$$r(t) = \frac{N_c(t)}{N_{S_{total}}}$$

where $N_c(t)$ is the number of captured sheep robots at step in time t, and $N_{S_{total}}$ is the total number of sheep robots.

Note the following: Case 2 shows the best overall sheep capture ratio peaking at around 45%. In cases 1 and 3, the sheep capture ratio rose to around 35%. In cases 3 and 4, large dog robot clusters that appear, simultaneously collapses are observed, however. Case 4, with the worst overall sheep capture ratio, is plagued by frequent cluster collapse that lowers performance and involves violent ratio rises and falls.

In **Fig. 14**, a bar chart comparing the average sheep capture ratio to four cases of robot density shows that sheepdog performance peaks at a robot density of around 15%.

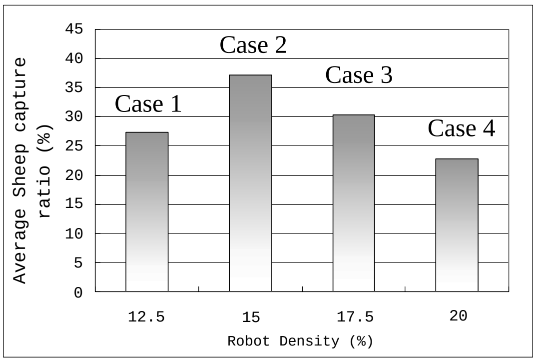

Fig. 14. Robot density and average sheep capture ratio.

Table 2. Comparison of robot density, sheep:dog ratio, and average number of robots within range of sight.

Case	Robot density d_r[%]	Sheep:dog ratio r_{sd} [%]	Average number of robots within range of sight n_{range}
1	12.5	25.0	3.125
2	15.0	50.0	3.750
3	17.5	75.0	4.375
4	20.0	100.0	5.000

Table 2 compares robot density, the sheep:dog ratio, and the average number of robots within the range of sight. Robot density of robots d_r is defined by the proportion of the total number of robots to the number of grids, which is 10000. Sheep:dog ratio r_{sd} is defined by the proportion of the number of sheep robots to the number of dog robots. The average number of robots within range of sight n_{range} indicates the ratio of robot congestion. In case 2, n_{range} is 3.75, with congestion acting to form a large dog robot cluster.

4.3. Robot Diversity Simulation

We now focus on diversity of dog robot parameter. The previous simulation used uniform characteristics for each sheep and dog robot. Here, we give different characteristics to each individual dog robot to compare performance between uniform and diverse dog groups. We also look at the relationship between sheepdog performance and the fractal dimension of cellular automata.

4.3.1. Robot Parameters

Number of dog robots is 1000, and number of sheep robots 500, in a setup of cases of dog robot parameters (**Fig. 15**) as follows:

- **Case 1**: $a_s = 1(\pm 1)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 1)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 10)$, and $c_d = 5(\pm 5)$.

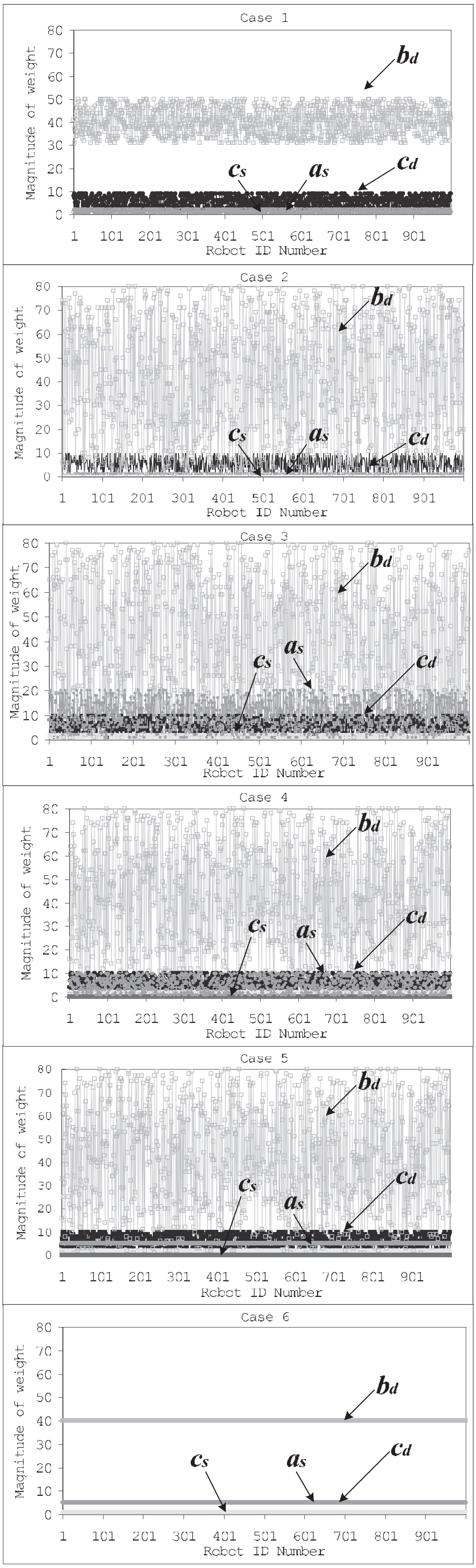

Fig. 15. Dog robot parameters.

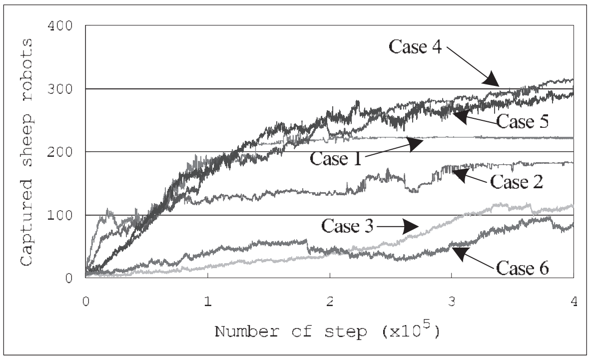

Fig. 16. Performances of different robot parameters.

- **Case 2**: $a_s = 1(\pm 1)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 1)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 40)$, and $c_d = 5(\pm 5)$.
- **Case 3**: $a_s = 10(\pm 10)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 1)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 40)$, and $c_d = 5(\pm 5)$.
- **Case 4**: $a_s = 5(\pm 5)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 1)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 40)$, and $c_d = 5(\pm 5)$.
- **Case 5**: $a_s = 5(\pm 0)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 1)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 40)$, and $c_d = 5(\pm 5)$.
- **Case 6**: $a_s = 5(\pm 0)$, $b_s = 0(\pm 0)$, $c_s = 1(\pm 0)$, $a_d = 0(\pm 0)$, $b_d = 40(\pm 0)$, and $c_d = 5(\pm 0)$.

Case 6 compares the uniform dog group to the diverse dog group in case 4. Case 4 defines each weight parameter as a random value from a uniform distribution, e.g., b_d is a random value between 1 and 80, whose median almost equals 40, i.e., the weight parameter from case 6. Simulation starts from the initial pattern in **Fig. 10**, and ends at step 400000.

4.3.2. Simulation Results

In simulation results (**Fig. 16**), the x axis is the number of steps over time and the y axis is the number of captured sheep robots calculated by averaging the results of ten simulations.

Note the following: Case 4 shows the best performance and case 6 shows the worst, with the big difference indicating that diversity in the dog robot group improves sheepdog performance over that in the uniform dog robot group.

We analyzed the number of captured sheep robots, the fractal dimension of cellular automata, and the sheepdog simulator snapshot from case 4 simulation (**Fig. 17**) and from case 6 simulation (**Fig. 18**), calculating fractal dimension by box-counting method.

At the start of simulation, the fractal dimension is approximately at 1.65, and decreases with the increasing number of captured sheep robots. The large cluster formed by dog robots capturing sheep robots decreases the fractal dimension, indicating a negative correlation between the number of captured sheep robots and the fractal dimension.

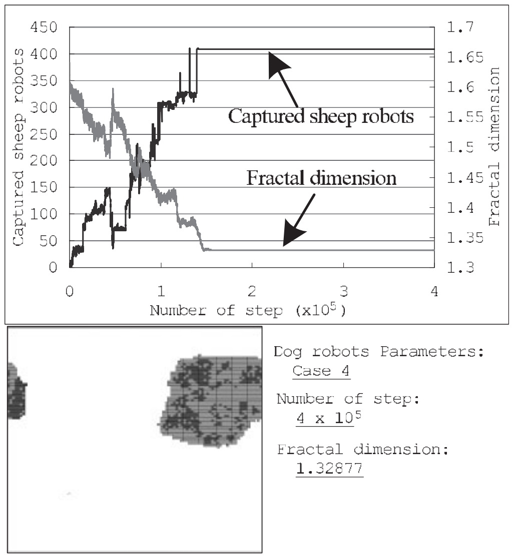

Fig. 17. Example of Case 4. Number of captured sheep robots, fractal dimension, and snapshot.

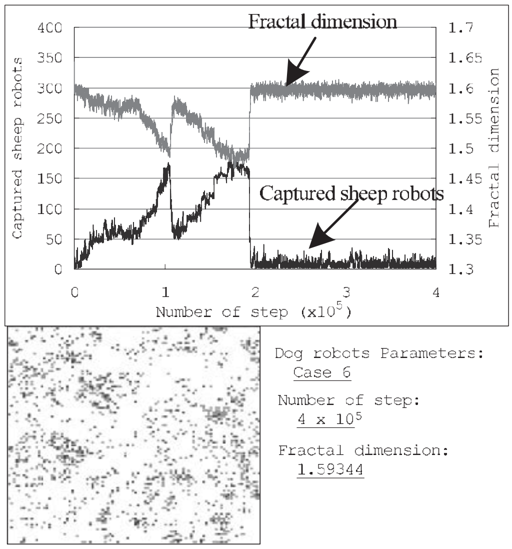

Fig. 18. Example of Case 6. The number of captured sheep robots, fractal dimension, and snapshot.

5. Conclusions

In the simulation framework we have proposed for the sheepdog problem, Boids model and cellular automata realize two different behavior models, i.e., sheep flocking and dog chasing and herding (capture).

Simulation results showed that sheepdog problem execution is affected by robot density, the sheep:dog ratio, and the number of robots within the robot range of sight.

Another decisive factor is robot diversity. The dog robot group with a diverse robot parameter shows better performance than that with a uniform robot parameter.

In future work, we plan to proceed with this research targeting optimization of robot weighting parameters through sharing distributed knowledge, introducing other factors into robot models such as movement cost, and combinations with other algorithms for pursuit problems.

References:

[1] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics," SIGGRAPH 1987 Conf. Proc., 21(4), pp. 25-34.

[2] W. J. Crowther, "Rule-based guidance for flight vehicle flocking," submitted to the Journal of Guidance, Dynamics and Control, Sep., 2002.

[3] U. Frisch, D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, and P. Rivet, "Lattice Gas Hydrodynamics in Two and Three Dimensions," Complex Systems, 1, pp. 649-707, 1987.

[4] G. W. Baxter and R. P. Behringer, "Cellular automata models of granular flow," Physical Review A, Vol.42, No.2, pp. 1017-1020, 1990.

[5] M. S. Alber, M. A. Kiskowski, J. A. Glazier, and Y. Jiang, "On Cellular Automaton Approaches to Modeling Biological Cells," Mathematical Systems Theory in Biology, Communication, IMA 134, Springer-Verlag, New York, pp. 1-40, 2002.

[6] H. M. Botee and E. Bonabeau, "Evolving And Colony Optimization," Adv. Complex Systems, 1, pp. 149-159, 1998.

[7] H. Kawamura, M. Yamamoto, K. Suzuki, and A. Ohuchi, "Multiple Ant Colonies Algorithm Based on Colony Level Interactions," IEICE Trans, Fundamentals, Vol.E83-A, No.2, pp. 371-379, Feb., 2000.

[8] T. Suzudo, "Spatial Pattern Formation in Asynchronous Cellular Automata with Mass Conservation," Physica A, Vol.343 C, pp. 185-200, 2004.

[9] K. Takahashi and M. Kakikura, "Research on cooperative capture by multiple mobile robots –a proposition of cooperative capture strategies in the pursuit problem–," Distributed Autonomous Robotic Systems, Vol.5, pp. 393-402, 2002.

[10] H. Yamaguchi, "A Cooperative Hunting Behavior by Mobile Robot Troops," IEEE Int. Conf. Robotics and Automation, p. 3204, 1998.

[11] N. Shimoyama, K. Sugawara, T. Mizuguchi, Y. Hayakawa, and M. Sano, "Collective Motion In a System of Motile Element," Physical Review Letters, Vol.76, No.20, pp. 3870-3873, 1996.

[12] G. Lee, Y. Komatsu, S. Sone, and N. Y. Chong, "Formation navigation of a team of mobile robots adapting to an environment," Int. Symposium on Robotics, Int. Federation of Robotics, 2005.

Name:
Yoshinobu Adachi

Affiliation:
Graduate School of Engineering, Tokyo Denki University

Address:
2-2, Nishiki-cho, Kanda, Chiyoda-ku, Tokyo 101-8457, Japan

Brief Biographical History:

1994 Joined SECOM Co., Ltd. Intelligent Systems Lab.
2001-2002 Visiting Scientist, VTT Technical Research Centre of Finland
2003 Joined Graduate School of Engineering, Tokyo Denki University

Main Works:

- "Simulation Language for Multiple Mobile Robots," Journal of Robotics and Mechatronics, Vol.9, No.5, 1997.

Membership in Academic Societies:

- The Robotics Society of Japan (RSJ)
- The Architectural Institute of Japan (AIJ)

Name:
Masayoshi Kakikura

Affiliation:
Professor, Department of Electronic Engineering, Faculty of Engineering, Tokyo Denki University

Address:
2-2, Nishiki-cho, Kanda, Chiyoda-ku, Tokyo 101-8457, Japan

Brief Biographical History:

1965 Joined Electro Technical Laboratory
1974-1975 Joined University of Edinburgh
1990-2006 Tokyo Denki University

Main Works:

- "A New Approach to Improve Absolute Positioning Accuracy of Robot Manipulators," Journal of Robotics Systems, Vol.4, No.1, pp. 145-156, 1987.
- "Setup Planning of Active Visual Sensing for Autonomous Robots," Journal of Robotics and Mechatronics, Vol.3, No.3, pp. 196-200, 1991.
- "Planning Twirling Rotation of an Object with a Multi-fingered Hand," Journal of Robotics Society of Japan, Vol.12, No.5, pp. 107-114, 1994.

Membership in Academic Societies:

- The Robotics Society of Japan (RSJ)
- The Society of Instrument and Control Engineeres (SICE)
- The Japan Society of Mechanical Engineers (JSME)