

Proceedings of
IES 2018

The 22nd Asia Pacific Symposium on Intelligent and Evolutionary Systems.



Sapporo, JAPAN

Dec.20-22, 2018

IES 2018 PROGRAM

Sapporo, Japan, December 20-22, 2018

Conference Day 1 - December 20, 2018 (Thu) Location: Small Auditorium	
12:30 – 16:30	IES 2018 Registration
13:00 – 13:10	Welcome speech by General chair
13:10 – 13:35	<p style="text-align: center;">Session Thu-1: Smart Systems Chair: <i>Prof. Hiroshi Sato</i></p> <p>A Study on Model-Based Development of Embedded System using Scilab/Scicos with TOPPERS/JSP <i>Masashi Sugimoto, Tomoki Shirakawa and Hitoshi Yoshimura</i></p>
13:35 – 14:00	<p>Point-cloud Based Geometric Modeling of Hokkaido-Ainu Motifs and Its Manufacturing Method <i>Tashi., Amm Sharif Ullah, Michiko Watanabe and Akihiko Kubo</i></p>
14:00 – 14:25	<p>Creating Digital Twin of Processed Surface using the Concept of Markov Chain for Industry 4.0 <i>Angkush Kumar Ghosh, Amm Sharif Ullah, Michiko Watanabe and Akihiko Kubo</i></p>
14:25 – 14:50	<p>El Farol Bar problem by Agreement algorithm based on trial and error behavior at the macro level <i>Nhuham Phung, Masao Kubo and Hiroshi Sato</i></p>
14:50 – 15:15	<p>Novelty Search-based Bat Algorithm: Adjusting Distance among Solutions for Multimodal Optimization <i>Takuya Iwase, Ryo Takano, Fumito Uwano, Hiroyuki Sato and Keiki Takadama</i></p>
15:15 – 15:30	<p style="text-align: center;">Coffee Break</p>
15:30 – 15:55	<p style="text-align: center;">Session Thu-2: Complex Systems Chair: <i>Prof. Keiki Takadama</i></p> <p>Analysis of Epidemic of Seasonal Influenza in Closed Space <i>Saori Iwanaga and Keigo Kawaguchi</i></p>
15:55 – 16:20	<p>The Influence of Social Media Writing on Online Search Behavior for Seasonal Events: The Sociophysics Approach <i>Nozomi Okano, Masaru Higashi and Akira Ishii</i></p>
16:20 – 16:45	<p>Emergence of Flocking Behaviors with Indirect Reward <i>Hiroshi Sato and Naoya Sekiguchi</i></p>
16:45 – 17:10	<p>When the Selfish Herd is Unsafe in the Middle <i>Wen-Chi Yang</i></p>
17:10 – 17:35	<p>Agent-Based Modeling and Complex Network Analysis for Clarifying Changes in the Japanese Textile and Apparel Industry B2B Networks <i>Yusaku Ogai, Yoshiyuki Matsumura, Yusuke Hoshino, Masanori Goka, Toshiyuki Yasuda and Kazuhiro Ohkura</i></p>
17:35 – 18:00	<p>Opinion Dynamics Theory for Analysis of Consensus Formation and Division of Opinion on the Internet <i>Akira Ishii and Yasuko Kawahata</i></p>

Conference Day 2 - December 21, 2018 (Fri) Location: Small Auditorium	
08:30 - 14:30	IES 2018 Registration
09:00 – 09:25	<p style="text-align: center;">Session Fri-1: Swarm Robotics Chair: <i>Prof. Keitaro Naruse</i></p> <p>Towards a Robotic Swarm using Deep Neuroevolution: An Experimental Study in Path Formation <i>Daichi Morimoto, Motoaki Hiraga and Kazuhiro Ohkura</i></p>
09:25 – 09:50	<p>Leading a swarm of mobile robots via a locally informed teleoperated robot <i>Toshiyuki Yasuda, Shunsuke Ota and Mitsuru Jindai</i></p>
09:50 – 10:15	<p>Effects of Body Size on Autonomous Specialization and Congestion of Robotic Swarms <i>Motoaki Hiraga and Kazuhiro Ohkura</i></p>
10:15 – 10:30	<i>Coffee Break</i>
10:30 – 10:55	<p style="text-align: center;">Session Fri-2: Machine Learning (1) Chair: <i>Prof. Keiji Suzuki</i></p> <p>The Novel Interactive Picture Book System: IOΞK <i>Saya Fujino and Naoki Mori</i></p>
10:55 – 11:20	<p>Prediction of Race Results and Generation of Predicted Articles in Keirin <i>Takumi Yoshida, Soichiro Yokoyama, Tomohisa Yamashita and Hidenori Kawamura</i></p>
11:20 – 11:45	<p>Vehicle Turn Signal Recognition using Internal features on Convolutional Neural Network <i>Keisuke Yoneda, Ryota Hagi and Naoki Suganuma</i></p>
11:45 – 12:10	<p>Combined Method of YOLO and IOU Tracker for Tracking and Identifying Chimpanzee <i>Yuichiro Ikeda, Hiroyuki Iizuka and Masahito Yamamoto</i></p>
12:10 – 13:30	<i>Lunch</i>
13:30 – 14:30	<p>Keynote Speech: System Intelligence For Service Robot Technology, <i>Prof. Hajime Asama, The University of Tokyo</i> Chair: <i>Prof. Takashi Kawakami</i></p>
14:30 – 14:45	<i>Break</i>
14:45 – 18:00	<p>Technical Tour : Okurayama Ski Jump Stadium, Sapporo Beer Museum</p>
18:30 – 20:30	<i>Banquet</i> Location: Keio Plaza Hotel Sapporo

Conference Day 3 - December 22, 2018 (Sat) Location: Small Auditorium	
08:00 - 12:00	IES 2018 Registration
08:30 – 08:55	<p>Session Sat -1: Swarm Intelligence Chair: Prof. Kazuhiro Ohkura</p> <p>A Novel Bare-bones Particle Swarm Optimization Algorithm with Co-evaluation <i>Jia Guo and Yuji Sato</i></p>
08:55 – 09:20	<p>Evolutionary Learning of Multiple TTSP Graph Structured Patterns from Positive and Negative TTSP Graph Data: Its Graph Pattern Display System and Performance Evaluation <i>Yuuki Yamagata, Fumiya Tokuhara, Yuito Inaba, Kouki Muratani, Tetsuhiro Miyahara, Yusuke Suzuki, Tomoyuki Uchida and Tetsuji Kuboyama</i></p>
09:20 – 09:45	<p>Modeling of driving force generated by rod wheel with single rod on loose soil <i>Taku Matsumoto, Keita Nakamura and Keitaro Naruse</i></p>
09:45 – 10:10	<p>The Substitute Worker Selection Method Using Agent-based Simulation <i>Kohei Hatamoto, Soichiro Yokoyama, Tomohisa Yamashita and Hidenori Kawamura</i></p>
10:10 – 10:25	<i>Coffee Break</i>
10:25 – 10:50	<p>Session Sat-2: Machine Learning (2) Chair: Prof. Tomohisa Yamashita</p> <p>Neural Network using Support Vector Machine and Genetic Algorithm <i>Kyoko Ogawa, Naoki Mori and Michifumi Yoshioka</i></p>
10:50 – 11:15	<p>Memory Augmented Neural Networks with Paired Controllers for One-shot Learning <i>Trung Kien Tran, Hiroshi Sato and Masao Kubo</i></p>
11:15 – 11:40	<p>Differentiation of communication signals to establish cooperation using Deep Q-Network <i>Hironobu Horiuchi, Hiroyuki Iizuka and Masahito Yamamoto</i></p>
11:45 – 12:10	<p>Deep Reinforcement Learning Method Enabling Stable Learning in Multi-Agent Environment <i>Kenji Hirata, Hiroyuki Iizuka and Masahito Yamamoto</i></p>
12:10 – 12:20	<i>Closing Talk</i>

A Study on Model-Based Development of Embedded System using Scilab/Scicos with TOPPERS/JSP

Masashi SUGIMOTO [†] Tomoki SHIRAKAWA [†] Hitoshi YOSHIMURA [‡]

[†] National Institute of Technology, Kagawa College, Mitoyo C., Japan

[‡] National Institute of Technology, Tomakomai College, Tomakomai C., Japan

sugimoto-m@es.kagawa-nct.ac.jp

Abstract: In recently, the technique of the model-based development has come to be used more widely on the site of the embedded system development. In the present study, we propose the technique for model-based development of the embedded system using Scilab/Scicos. As an instance of applying we tried the stabilizing control of the two-wheeled inverted pendulum, using LEGO Mindstorms NXT. The control program was generated from the designed feedback control system automatically as C program and applied as TOPPERS/JSP, and also LEGO Mindstorms NXT has been uploaded. As a result of the experiment, we obtained stable execution results. Therefore, the utility of the proposed technique was confirmed.

Keywords: Model-based Development, Code Generation, LEGO Mindstorms NXT, NXTway-GS, Scilab/Scicos, TOPPERS/JSP

1 Introduction

In case of the development for embedded systems, non-object oriented languages, such as C language or assembler language will be used in situation of development. However, in recent, object-oriented development using UML has also been carried out, moreover, even in the hardware design phase has been used. UML is a modelling language that has focused on analysing and designing for software development. Therefore, it is necessary to pay sufficient attention to whether the contents of the target block to analyse or designing software are provided by region or hardware region[1]. From this viewpoint, it is pointed out that UML cannot be applied in several aspects of the development of an embedded system[2]. For example, it is hard to describe for a control system of a function, such as a describe for concurrency control, a control timing, and a feedback loop for continuous system. To solve these problems, a research have been provided a method that applying the RT middleware technique. In this method, an RT middleware had been capturing Scilab/Scicos as a component. From this technique, this method generates some tasks from the Scicos block diagram, indirectly.

In this study, the designation and simulation of control systems of the embedded systems have been focused on. Moreover, as shown in fig. 1, “NXTway-GS” has been applied as a target robot of control. We have previously reported that focused on development using TOPPERS/OSEK, as known as TOPPERS/ATK[3, 4]. In this paper, we will try to extend the model-based development system support to TOPPERS/JSP.

From this standpoint, we will re-consider the block of a system that called an Auto-code generator.

This paper is organized as follows: In section 2, outline of model-based development, and previous study, will be stated. In section 3, a verification experiment configuration will be described. In section 4, the summary of this work is concluded.

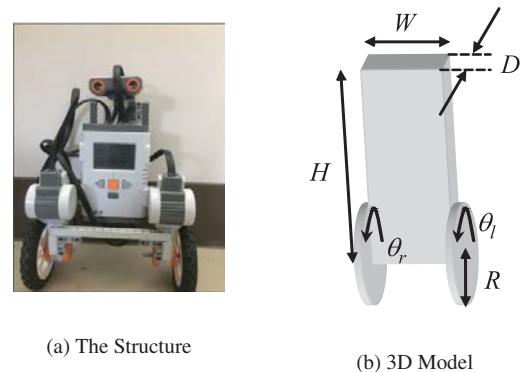


Fig. 1: The Structure of “NXTway-GS.”

2 System Configuration and Flow of Procedure

We aim to structure a system that can be directly generated a task or control API for TOPPERS/JSP from Scicos block

diagrams. For this solution, we will be able to complement a missing function of UML. Further, Scilab/Scicos has simulation function, so we can obtain simulation results, directly. Therefore, the proposed system has an environment that provides portability and efficiency development.

In previous research by Yamamoto[5], simulation is carried out by MATLAB/Simulink as a numerical calculation system. That is a highly reliable system with many use records, however, it is very expensive. On the other hand, Scilab/Scicos used in this study is an open source numerical calculation system developed by INRIA and ENPC, that is open source software. Moreover, an expansion package is also distributed free of charge. From this standpoint, Scilab/Scicos will be applied in this study.

Moreover, nowadays, this technology will be needed what the demands on the robotics, emmbedded systems, and IoT industry are expected to increase. In recently, TOPPERS project has been developed or released some platforms. However, these works are conform uITRON or not is depends on each work. Moreover, some works are conform VDX specify. If target processor-oriented or target task-oriented application will be developed, it is necessary to support for development that roots in RTOS beyond the differences of specifications such as uITRON or VDX specify. As this result, our proposed system will be worthly we considered.

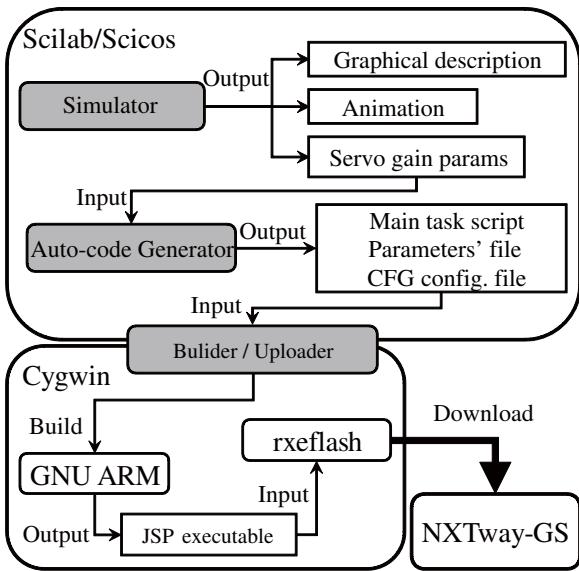


Fig. 2: Outline for the Proposal of Model-based Development System.

In our former study [3, 4], we had been considered functions (fig. 2) as below:

- (1) Simulator
- (2) Auto-code Generator
- (3) Builder / Uploader

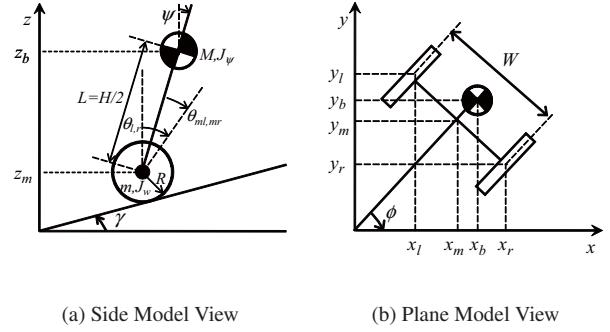


Fig. 3: Model of Side and Plane View of NXTway-GS [3].

However, we cannot apply the system that had been development, to TOPPERS/JSP directly. Therefore, we focused on generated files by (2) auto-code generator. In detail, we considered the necessary files to generate a TOPPERS/JSP executable. In former study [3, 4], we designed an OIL file for TOPPERS/OSEK, however, in TOPPERS/JSP, a CFG configuration file will be needed. Moreover, task notation and variable definition are different as TOPPRES/OSEK. As mentioned above, we have re-considered these problems and rebuild the systems.

3 Verification Experiment

3.1 The Experimental Outline

In this verification experiment, the posture of a two-wheeled inverted pendulum “NXTway-GS”, as an application that was stabilized using a computer simulation as a verification experiment. Moreover, an actual robot “NXTway-GS”, has been also applied postural control. In this experiment, the simulational function of Scicos will be used between designing a control model of NXTway-GS and LQR controller.

3.2 Configuration of Simulation 1 – for NXTway-GS Model

As shown in fig. 3, “NXTway-GS” can be described as a two-wheeled inverted pendulum model. The coordinate system used in Section 3.3 is described in fig. 3. Moreover, in fig. 3, ψ denotes the body pitch angle and $\theta_{l,r}$ denotes the wheel angle (l and r indicate left and right, respectively). Further, $\theta = 1/2 \cdot (\theta_l + \theta_r)$, and $\theta_{ml,mr}$ denotes the DC motor angle (l and r indicate left and right, respectively). The NXTway-GS’s physical parameters are listed in table 1.

3.3 Configuration of Simulation 2 – for NXTway-GS Modeling

NXTway-GS's motion equations can be derived, shown in fig. 3. If the direction of the model is the x -axis positive direction at $t = 0$, the equations of motion for each coordinate can be given as follows [3]:

$$\begin{aligned} & [(2m + M)R^2 + 2J_w + 2n^2 J_m] \ddot{\theta} \\ & + (MLR - 2n^2 J_m) \ddot{\psi} - Rg(M + 2m) \sin \gamma = F_\theta \quad (1) \\ & (MLR - 2n^2 J_m) \ddot{\theta} + (ML^2 + J_\psi + 2n^2 J_m) \ddot{\psi} - MgL\psi = F_\psi \quad (2) \\ & \left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2} (J_w + n^2 J_m) \right] \ddot{\phi} = F_\phi \quad (3) \end{aligned}$$

Here, \mathbf{x}_1 and \mathbf{x}_2 represent state variables. In addition, \mathbf{u} denotes input:

$$\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^\top \quad (4)$$

$$\mathbf{x}_2 = [\phi \ \dot{\phi}]^\top \quad (5)$$

$$\mathbf{u} = [v_l \ v_r]^\top \quad (6)$$

From above equations, state equations of NXTway-GS can be derived using eqs. (1), (2), and (3).

$$\frac{d}{dt} \mathbf{x}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u} + \mathbf{S} \quad (7)$$

$$\frac{d}{dt} \mathbf{x}_2 = \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u} \quad (8)$$

In this verification experiment, only a state variable \mathbf{x}_1 will be used. Because \mathbf{x}_1 contains the body pitch angles as variables ψ and $\dot{\psi}$, which are important for self-balancing. That's why the plane motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$) will not be considered in this experiment:

$$\frac{d}{dt} \mathbf{x}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u} \quad (9)$$

3.4 Configuration of Simulation 3 – Applying the LQR as A Controller

In this study, Linear-Quadratic Regulator (as known as LQR) is applied. The feedback gain \mathbf{k}_f so as to minimize the quadratic cost function J will be calculated by this LQR given as eq. (10).

$$J = \int_0^\infty [\mathbf{x}^\top(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^\top(t) \mathbf{R} \mathbf{u}(t)] dt \quad (10)$$

In this verification experiment, matrices \mathbf{Q} and \mathbf{R} are as following:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \quad (11)$$

$$\mathbf{R} = 1 \times 10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (12)$$

Table. 1: Physical Parameters of NXTway-GS

Symbol	Value	Unit	Physical property
g	9.81	[m/s ²]	Gravity acceleration
m	0.03	[kg]	Wheel weight [3]
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm ²]	Wheel inertia moment
M	0.635	[kg]	Body weight [3]
W	0.14	[m]	Body width
D	0.04	[m]	Body depth
H	0.144	[m]	Body height
L	$\frac{H}{2}$	[m]	Distance of center of mass from wheel axle
J_ψ	$\frac{ML^2}{3}$	[kgm ²]	Body pitch inertia moment
J_ϕ	$\frac{M(W^2+D^2)}{12}$	[kgm ²]	Body yaw inertia moment
J_m	1×10^{-5}	[kgm ²]	DC motor inertia moment [3, 4]
R_m	6.69	[\Omega]	DC motor resistance [3]
K_b	0.468	[V·s/rad.]	DC motor back EMF constant [3, 4]
K_t	0.317	[N·m/A]	DC motor torque constant [3, 4]
n	1	[1]	Gear ratio [3]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [3, 4]
f_w	0	[1]	Friction coefficient between wheel and floor [3, 4]

Mentioned above equation, \mathbf{k}_f , is a gain of optimal feedback, will be obtained by minimizing J . However, in the equation, below equations that includes Riccati equation will be solved:

$$\mathbf{A}^\top \mathbf{S} + \mathbf{S} \mathbf{A} - \mathbf{S} \mathbf{A} \mathbf{R}^{-1} (\mathbf{B}^\top \mathbf{S}) + \mathbf{Q} = 0 \quad (13)$$

$$\mathbf{x}^\top = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{C}_1(1,:) & 0 \end{bmatrix} \quad (14)$$

$$(15)$$

In this equation, $\mathbf{C}_1(1,:)$ denotes the 1st row of the matrix \mathbf{C}_1 . From these results, the feedback gain \mathbf{k}_f of the state-feedback stabilizer will be applied. However, in the verification experiment, the plane movement of the two-wheeled inverted pendulum will be not considered. Hence, $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$ were considered.

3.5 Simulation Conditions

In this verification simulation, the action signal with an unknown disturbance signal $\mathbf{d}(t)$ (fig. 4) will be mixed. In figure 4, we can describe the block diagram as fig. 5.

In this verification experiment, “NXTway-GS” model takes self-balancing. In addition, the properties of disturbance signal that we will provide as input and other conditions of a simulation are listed in table 2.

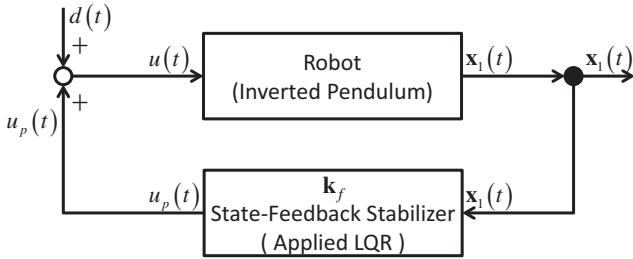


Fig. 4: Control Input Obtained by Mixing the Action and Disturbance Inputs.

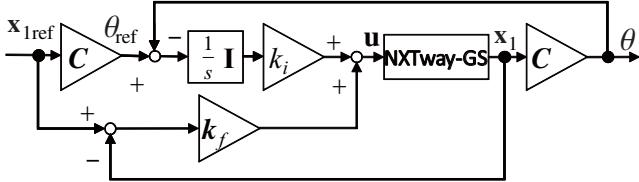


Fig. 5: NXTway-GS Servo Controller Block Diagram.

3.6 Results of Experiment

In this experiment, firstly, we have done the simulation using the servo controller that had been applied to the NXTway-GS model. In addition, the NXTway-GS model performs stationary balancing while 20 [sec.]. Figure 6 shows a simulation result of stationary balancing at initial value of body pitch angle equals to -6.0 [deg.]. Moreover, figure 7 show the experimental results using “actual” NXTway-GS. In this experiment, the NXTway-GS is executed by TOPPERS/JSP executable that had been generated by auto-code generator. In addition, the NXTway-GS performs that keeps stationary balancing for 20 [sec.].

3.7 Discussion on Simulated Results of the Proposed Method

Now we are trying to focus on simulation result. According to fig. 6, also, compensation result obtained using the proposed method (shown as the solid line) approach and oscillate to near zero, with time. Therefore, it can be said that the wheel is moving while trying to decrease the body pitch angle $\psi(t)$.

Next, we are trying to focus on experimental result as shown in fig 7. Here, focusing the control response point

Table. 2: Parameters for the Simulation of the Proposed Method

Symbol	Value	Unit	Physical property
ψ_0	-6.0	[deg.]	Initial value of body pitch angle
γ_0	0.0	[rad.]	Slope angle of movement direction
t_s	0.05	[s]	Sampling rate

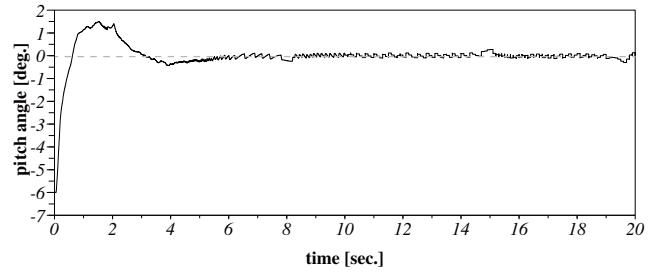


Fig. 6: Control Response of Body Pitch Angle ψ . (Simulational Result) (1)

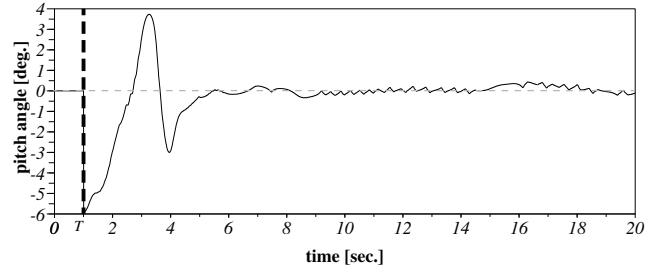


Fig. 7: Control Response of Body Pitch Angle ψ . (Actual Robot operated by TOPPERS/JSP)



Fig. 8: The scene of Stationary balancing control.

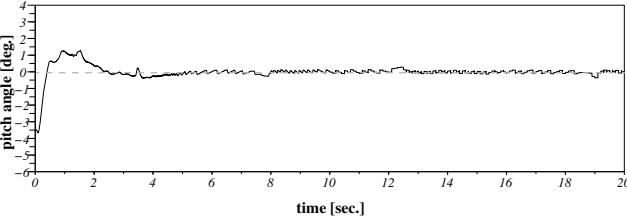


Fig. 9: Control Response of Body Pitch Angle ψ . (Simulational Result) (2)[4]

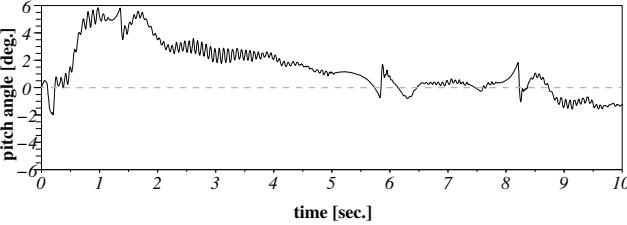


Fig. 10: Control Response of Body Pitch Angle ψ . (Actual Robot operated by TOPPERS/OSEK)[4]

is shown after $t = 1$ [sec.]. NXTway-GS does not run until 1 [sec.] passes because of gyro calibration. Therefore, we will only argue and focus on the part of the graph pertaining to state prediction part shown in T . According to fig. 7, compensation result obtained using the proposed method (shown as the solid line) approach and oscillate to near zero, with time. Therefore, it can be said that the wheel is moving while trying to decrease the body pitch angle $\psi(t)$.

In contrast, now we consider the control result using TOPPERS/OSEK. In this case, as similar as controlling by TOPPERS/JSP, fig. 7, compensation result obtained using the proposed method (shown as the solid line) approach and oscillate to near zero, with time. Therefore, it can be said that the wheel is moving while trying to decrease the body pitch angle $\psi(t)$.

From this viewpoint, it can be concluded that the simulation results will be retaining stable state, continuously. From this viewpoint, it can be said that this work is as controllable as former work. Therefore, it will be concluded that the results of the each verification experiment will be reasonable.

4 Conclusion

In this paper, the designation and simulation of control systems of the embedded systems have been focused on. We have previously reported that focused on development using TOPPERS/OSEK. In this paper, we have been extending the model-based development system support to TOPPERS/JSP. As a result, the structure has been building a system that can be directly generated a task or control API for TOPPERS/JSP from Scicos block diagrams. Further, Scicos has sim-

ulation function, so we can obtain simulation results, directly.

From each verification experiment, it can be concluded that the each result will be retaining stable state, continuously. Accordingly, the proposed method that is provided TOPPERS/JSP executable that includes simulation results and physical parameters can be controlled the actual NXTway-GS, stationary balancing as similar as simulational result, moreover, as similar as former works that applied by TOPPERS/OSEK, will be concluded.

References

- [1] T. Bahill and J. Daniels, “Using object-oriented and UML tools for hardware design: a case study,” *Systems Engineering*, Vol. 6 No. 3, pp.28-48, 2003.
- [2] N. Ando, “Introduction to RT-middleware for beginners : OpenRTM-aist-1.0 and how to use it,” *Journal of the Robotics Society of Japan*, Vol.28 No. 5, pp.550-555, 2010.
- [3] M. Sugimoto, H. Yoshimura, T. Abe, and I. Ohmura, “A study on model-based development of embedded system using Scilab/Scicos,” In proc. of the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp.343-344, 2010.
- [4] M. Sugimoto, H. Yoshimura, T. Abe, and I. Ohmura, “A study on model-based development of embedded system using Scilab/Scicos : development of auto-code generator,” In proc. of the JSME annual Conference on Robotics and Mechatronics 2010 (Robomec2010), 2A2-C10, pp1-4, 2010.
- [5] Y. Yamamoto, “NXTway-GS model-based design – control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT–,” *Cybernet Systems Co., Ltd.*, 2009.

Point-cloud Based Geometric Modeling of Hokkaido-Ainu Motifs and Its Manufacturing Method

Tashi † AMM Sharif Ullah † Michiko Watanabe † Akihiko Kubo †

† Kitami Institute of Technology, Koen-cho 165, Kitami, Japan
d187308030@std.kitami-it.ac.jp

Abstract: Hokkaido-Ainus uses different types of motifs for decorating their houses, clothing, ornaments, utensils, and spiritual goods. The motifs symbolize their identity as well as their sense of beauty. This article elucidates the geometric modeling of Hokkaido-Ainu motifs and its manufacturing method. Firstly, Hokkaido-Ainu motifs are identified and then classified them into groups. Next, the geometric shape of the motifs is represented in form of point-cloud which are used to create the solid CAD models with the aid of commercially available CAD packages. Finally, the solid CAD models are then realized by producing their replicas using a commercially available 3D printer. The finding of this study will help those who want to digitize cultural significant artifacts as well as manufacturing their replicas using the Additive Manufacturing process.

Keywords: Point-cloud, Geometric modeling, Additive Manufacturing, Hokkaido-Ainu, Motifs

1. Introduction

The Ainu or the Aynu are an indigenous people of Japan and Russia. The Japanese Ainu used to live in northern Japan, namely, Hokkaido and Aomori regions [1]. The Russian Ainu used to live in Sakhalin peninsula and the Kuril Islands. Based on their inhabitant, Ainu are called Hokkaido-Ainu, Tohoku-Ainu, Sakhalin-Ainu, and Chishima-Ainu [2].

In this paper, we focus on Hokkaido-Ainu and their motifs. Hokkaido-Ainu uses different types of motifs for decorating their houses, clothing, ornaments, utensils, and spiritual goods. The motifs symbolize their identity as well as their sense of beauty. There are many souvenirs crafted with Hokkaido-Ainu motifs available in some of the handicraft shops in Hokkaido. Thus, Hokkaido-Ainu motifs are important in terms of both cultural heritage and economic activities.

On the other hand, a great deal of craftsmanship is needed to manufacture the Hokkaido-Ainu motifs with the required precision and accuracy. This craftsmanship can be digitized using the Reverse Engineering (RE) based Additive Manufacturing (AM).

Recently, due to advances in RE technologies (such as laser and photogrammetric scanning) [3-6] and the advent of AM process (also known as 3D printing) [7-11], RE based AM has been used for digitizing, preserving, and restoring artifacts having cultural significance [12, 13]. For example, museums are using RE based AM for building a replica as well as building the missing parts of an importance relic [14, 15]. Like a 3D sculpture or architecture, a piece of 2D visual art such as a painting, photography, or drawing can be digitized in terms of a solid model using RE based AM. For instance, Fueferi et al. [16] have created a 2.5D tactical

model from a painting (i.e., 2D art), which helps individual having vision problems to experience the content of painting having historical significance. Even historical sites and artifacts are digitally documented using a 3D scanner [17]. Therefore, the combined use of 3D scanning and AM technologies (e.g., 3D printer) allows the replication of physical objects. However, the point-cloud obtained using the 3D scanner or image processing techniques requires a lot of computational efforts. Thus, the objective of this paper is to model Hokkaido-Ainu motifs using an analytically created point-cloud and its manufacturing method.

The rest of the paper is organized as follows. Section 2 describes the method used for the geometric modeling of Hokkaido-Ainu motifs and its manufacturing method. Section 3 describes the results and discusses its implications. Section 4 concludes this paper.

2. Method

This section describes the method used for the geometric modeling of Hokkaido-Ainu motifs and its manufacture method. The steps for the geometric modeling and manufacturing of Hokkaido-Ainu motifs is illustrated in Fig. 1. As shown in Fig. 1, the method comprises 6 steps. Step 1 describes the identification of Hokkaido-Ainu motifs. Step 2 describes the geometric modeling of Hokkaido-Ainu motifs. Step 3 describes the solid CAD modeling of Hokkaido-Ainu motifs. Step 4 describes triangulation modeling of the solid CAD model of Hokkaido-Ainu model. Step 5 describes the execution of the triangulated model of Hokkaido-Ainu motifs in AM devices (e.g. 3D printer). Finally, in Step 6, the physical model of Hokkaido-Ainu motifs is obtained by cleaning and finishing its surface. Step 3 to Step 6 is a manufacturing process. Each step is described detail in

the following subsections.

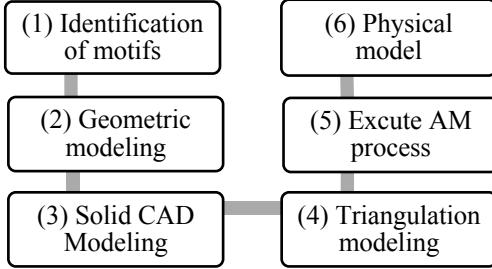


Fig. 1. Steps for the geometric modeling and manufacturing of Hokkaido-Ainu motifs.

2.1. Identification of Hokkaido-Ainu Motifs

As mentioned in the previous section, Hokkaido-Ainu uses different types of motifs for decorating their houses, clothing, ornaments, utensils, and spiritual goods. For example, Fig. 2 shows the tapestry of Hokkaido-Ainu displayed in the Sapporo station. It is clear from Fig. 2, that a single tapestry used different kinds of Hokkaido-Ainu motifs. Thus, identification of each motif is very crucial before going for geometric modeling and manufacturing.



Fig. 2. Tapestry of Hokkaido-Ainu.

The Sapporo city authority have listed 14 different types of Hokkaido motifs as shown in Table 1. As seen in Table 1, the Motif No. 1 (called Ayus in Ainu language) takes the shape of a thorn. The Motif No. 2 (Morew in Ainu language) takes the shape of a spiral. The Motif No. 3 (called Arus-morew in Ainu language) takes the shape of a spiral with small thorns. The Motif No. 4 (called Sikikew-nu-morew in Ainu language) takes the shape of a spiral with corners. The Motif No. 5 (called Sik in Ainu language) takes the shape of an eye. The Motif No. 6 (called Utasa in Ainu language) takes the shape of intersects each other. The Motif No. 7 (Uren-morew in Ainu language) takes the shape of two spirals. The Motif No. 8 (Ski-uren-morew in Ainu language) takes the shape of two spirals shape with an eye. The Motif No. 9 (called Morew-etok in Ainu language) takes the shape of a spiral plant. The Motif No. 10 (called Punkar in Ainu language) takes the shape of a

vane. The Motif No. 11 (called Apapo-piras (u) ke in Ainu language) takes shape of a flower. The Motif No. 12 (called Apapo-epuy in Ainu language) takes shape of flower bud. The motifs No. 13 and 14 shapes do not have Ainu names, but they look like a heart type shape and a fishing bell shape, respectively.

Morew, Sik, Utasa, and Ayus are frequently used motifs, and, thus, can be classified as the main motifs. Arus-morew, Sikike-nu-morew, Sikikew-nu-morew, and Uren-morew are motifs created by modifying the main motifs, and, thus, can be classified as synthetic motifs. The motifs called Morew-etok, Punkar, Apapo-piras (u) ke, and Apapo-epuy represent plants, and, thus, can be classified as the plant motifs. The other motifs (i.e., the motifs No. 13 and No. 14) can be classified as other motifs.

Table 1: Different types of Hokkaido-Ainu motifs [13, 18].

Number	Motif	Ainu Name
1	~	Ayus
2	◎	Morew
3	◎+	Arus-morew
4	◎*	Sikikew-nu-morew
5	★	Sik
6	++	Utasa
7	◎◎	Uren-morew
8	◎◎★	Ski-uren-morew
9	~	Morew-etok
10	~	Punkar
11	††	Apapo-piras(u) ke
12	△	Apapo-epuy
13	♡	—
14	◆	—

2.2. Geometric Modeling of Hokkaido-Ainu Motifs

Once completing the identification of the motifs as described in Section 2.2, then the motifs are mathematically defined. The main motifs and synthetic motifs are shown in Table 1 can be geometrically modeled using the mathematical equations. However, the plant motifs and other two motifs (i.e., the motifs No. 13 and No. 14) are challenging for geometric modeling using mathematical equations. An alternatively, one can use the algorithmic approach. In our previous work [13], we used both equation-based-approach and algorithmic approach. The equation-based-approach makes some complicity for the user because the user may not be familiar with the right set equations to be used. Therefore, in this paper, we used only the algorithmic

approach. The geometric shape of the motifs is modeled in terms of points called point-cloud because the point-cloud data are useful for the solid CAD modeling.

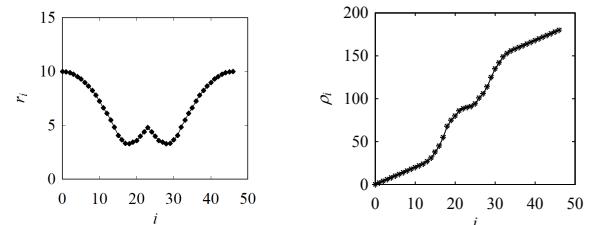
The proposed algorithm operates recursively by two user-defined parameters called distance and angle. The steps of the proposed algorithm for geometric modeling of motifs are shown below.

Point-cloud Creating Algorithm

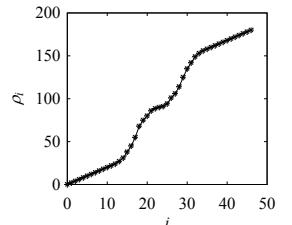
- 1 Define: Center Point $P_c = (P_{cx}, P_{cy}) \in \mathbb{R}^2$, Length $d > 0$, Initial Angle $\alpha \in \mathfrak{R}$
- 2 Define: Instantaneous Distances ($r_i \in \mathfrak{R} | i = 1, \dots, n$)
- 3 Define: Rotational Angles ($\rho_i \in \mathfrak{R} | i = 1, \dots, n$)
- 4 Calculate: $P_0 = (P_{0x}, P_{0y})$ so that $P_{0x} = P_{cx} + d \cos \alpha$ and $P_{0y} = P_{cy} + d \sin \alpha$
- 5 Iterate:
 - For $i = 1, \dots, n$
 - Rotate P_0 by an angle ρ_i around P_c in the counter-clockwise direction to create $P_i = (P_{ix}, P_{iy})$ so that
 - $$P_{ix} = P_{cx} + (P_{0x} - P_{cx}) \cos \rho_i - (P_{0y} - P_{cy}) \sin \rho_i$$
 and
 - $$P_{iy} = P_{cy} + (P_{0x} - P_{cx}) \sin \rho_i - (P_{0y} - P_{cy}) \cos \rho_i$$
 - Extend P_i to P_{ei} that is point on the line P_cP_i at a distance r_i from P_c
 - $$P_{eix} = P_{cx} + (P_{ix} - P_{cx}) \frac{r_i}{d}$$
 and
 - $$P_{eiy} = P_{cy} + (P_{iy} - P_{cy}) \frac{r_i}{d}$$
 - End for
- 6 Output: Point-Cloud, $PC = \{P_{ei} | i = 1, \dots, n\}$

The proposed algorithm consists of four steps. Step 1 is the input step and the following entities are defined: Center Point $P_c = (P_{cx}, P_{cy}) \in \mathbb{R}^2$, Length $d > 0$, Initial Angle $\alpha \in \mathfrak{R}$, Instantaneous Distances ($r_i \in \mathfrak{R} | i = 1, \dots, n$), and Rotational Angles ($\rho_i \in \mathfrak{R} | i = 1, \dots, n$). Step 2 is the calculation step and an Initial Point $P_0 = (P_{0x}, P_{0y})$ is calculated in this step, where P_0 is a point at a distance d and the line P_cP_0 makes an angle α in the counter-clockwise direction from the x -axis. Step 3 is the iteration step and in this step, the points denoted as $P_i = (P_{ix}, P_{iy})$ are created by rotating P_0 at an angle ρ_i in the counter-clockwise direction from the x -axis. Afterward, the points $P_i = (P_{ix}, P_{iy})$ are placed at a distance r_i from P_c resulting $P_{ei} = (P_{ix}, P_{iy}), i = 1, \dots, n$. Step 4 is the output step and in this step the points $P_{ei} = (P_{ix}, P_{iy}), i = 1, \dots, n$ are collected for creating the desired Point-Cloud, i.e., $PC = \{P_{ei} | i = 1, \dots, n\}$.

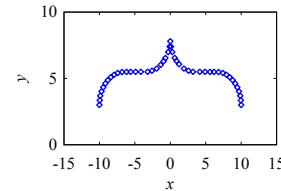
Fig. 3 shows the point-cloud of Ayus created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 3$, and where r_i and ρ_i are systematically adjusted for $i = 1, 2, \dots, 46$. Fig. 4 shows the point-cloud of Morew created by the proposed algorithm, for $\alpha = 270^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 20$, and where r_i is systematically adjusted and ρ_i is



(a) Input: r_i



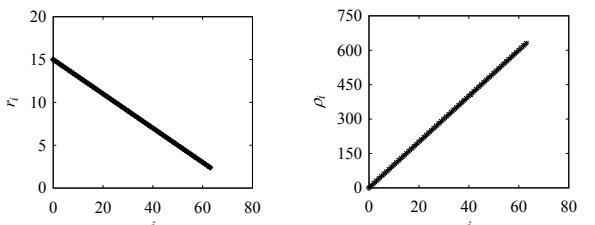
(b) Input: ρ_i



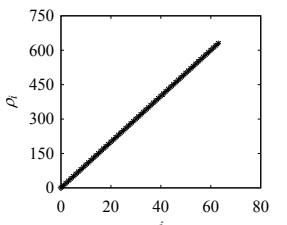
(c) Output: Point-cloud

(d) Solid CAD Model

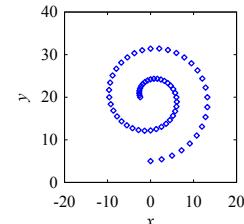
Fig. 3. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Ayus.



(a) Input: r_i



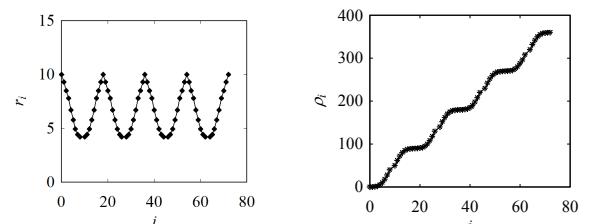
(b) Input: ρ_i



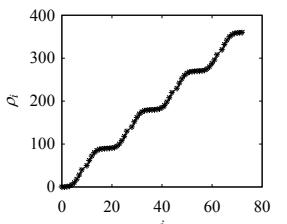
(c) Output: Point-cloud

(d) Solid CAD Model

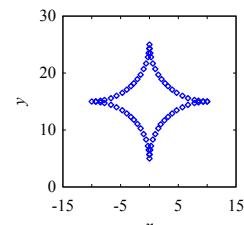
Fig. 4. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Morew.



(a) Input: r_i



(b) Input: ρ_i



(c) Output: Point-cloud

(d) Solid CAD Model

Fig. 5. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Sik.

linearly increased from 0° to 630° , for $i = 1, 2, \dots, 63$. The

same point-cloud can be used for Arus-morew and Sikikew-nu-morew with slight modification.

Fig. 5 shows the point-cloud of Sik created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 15$, and where r_i and ρ_i are systematically adjusted for $i = 1, \dots, 72$. Fig. 6 shows the point-cloud of Utasa created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 15$, where r_i is decreased from 10 to 0 keeping $\rho_i = 0^\circ$, for $i = 1, 2, \dots, 10$, and repeated the same procedure for $\rho_i = 90^\circ$, $\rho_i = 180^\circ$, and $\rho_i = 270^\circ$.

Fig. 7 shows the point-cloud of Uren-morew created by the proposed algorithm. As shown in Fig. 7, it is created by using two Morews. For first Morew, $\alpha = 270^\circ$, $d = 10$, $P_{cx} = -6.6$, $P_{cy} = 15$, where r_i is systematically adjusted and ρ_i is linearly increased from 0° to 630° , for $i = 1, 2, \dots, 63$, and for second Morew, $\alpha = 240^\circ$, $d = 10$, $P_{cx} = 6.6$, $P_{cy} = 15$, where r_i is systematically adjusted and ρ_i is linearly increased from 840° to 1470° , for $i = 1, 2, \dots, 64$. Fig. 8 shows the point-cloud of Sik-uren-morew created by the proposed algorithm. As motif name indicates that it can be created by combining Sik and Uren-morew as shown in Fig. 5 and Fig. 7, respectively.

Fig. 9 shows the point-cloud of Morew-etok, for $\alpha = 320^\circ$, $d = 10$, $P_{cx} = 3$, $P_{cy} = 15$, and where r_i and ρ_i are systematically adjusted for $i = 1, 2, \dots, 48$. Fig. 10 shows the point-cloud of Punkar created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 5$, $P_{cy} = 20$, and where r_i and ρ_i are systematically adjusted for $i = 1, 2, \dots, 30$. Fig. 11 shows the point-cloud of Apapo created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 20$, and where r_i and ρ_i are systematically adjusted for $i = 1, 2, \dots, 52$.

Fig. 12 shows the point-cloud of Apapo-epuy created by the proposed algorithm, for $\alpha = 0^\circ$, $d = 10$, $P_{cx} = 0$, $P_{cy} = 20$, and where r_i and ρ_i are systematically adjusted for $i = 1, 2, \dots, 67$.

2.3. Solid CAD Modeling

Now the point-clouds created in Section 2.2 are transferred to commercially available CAD packages for solid CAD modeling. The results of the CAD modeling are shown in Fig. 3(c), 4(c), 5(c), 6(c), 7(c), 8(c), 9(c), 10(c), 11(c) and 12(c).

2.4. Triangulation Modeling

Once the solid CAD modeling is done, the triangulation modeling is performed simply by saving the solid CAD model into an STL data using CAD packages. The results of the CAD modeling are shown in Fig. 3(d), 4(d), 5(d), 6(d), 7(d), 8(d), 9(d), 10(d), 11(d) and 12(d).

2.5. Execution of Additive Manufacturing Process

The STL data generated from the triangulation modeling is transferred to the AM devices (e.g., 3D printer) and then execute the 3D printer to build the physical model.

2.6. Clean and Finish Physical Model

Finally, the physical model is removed from the 3D printer and then clean the surface. Sometimes the surface finishing like polishing and painting may require

depending upon the user choice.

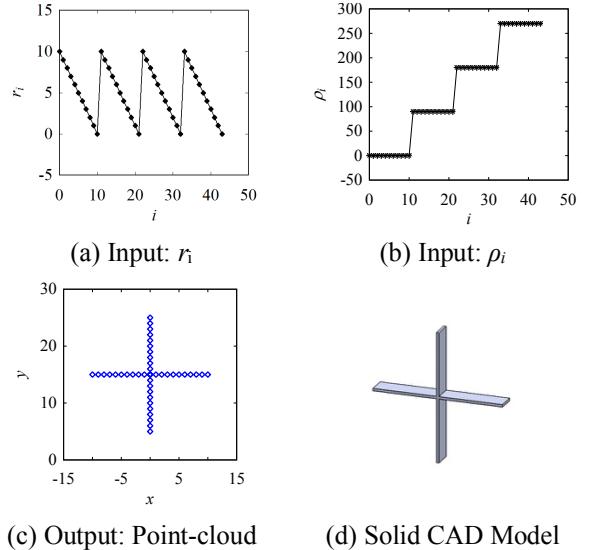


Fig. 6. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Utasa.

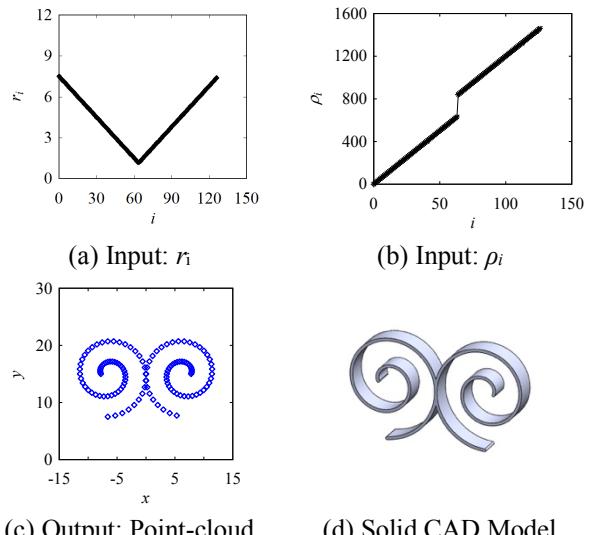


Fig. 7. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Uren-morew.

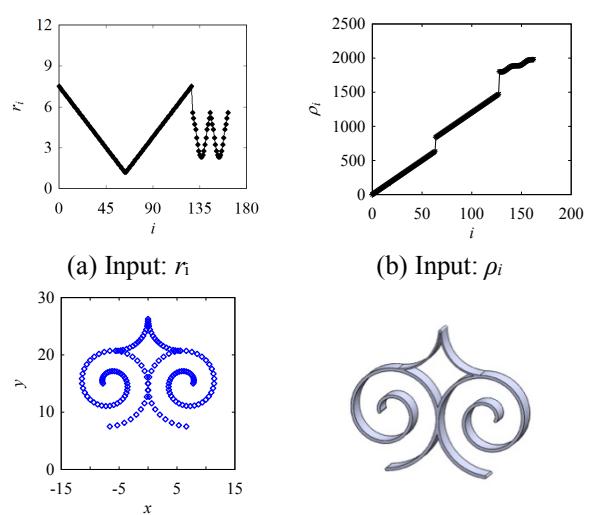


Fig. 8. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Sik-uren-morew.

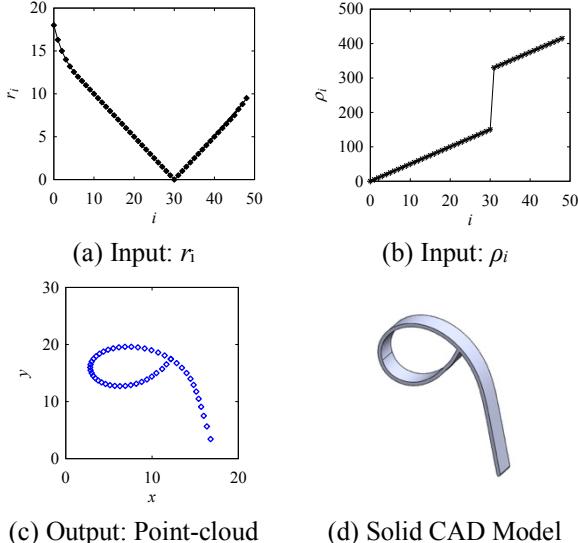


Fig. 9. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Morew-etok.

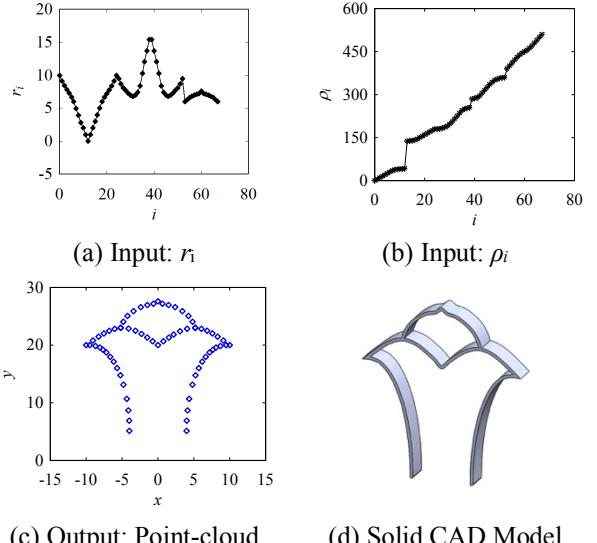


Fig. 12. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Apapo-epuy.

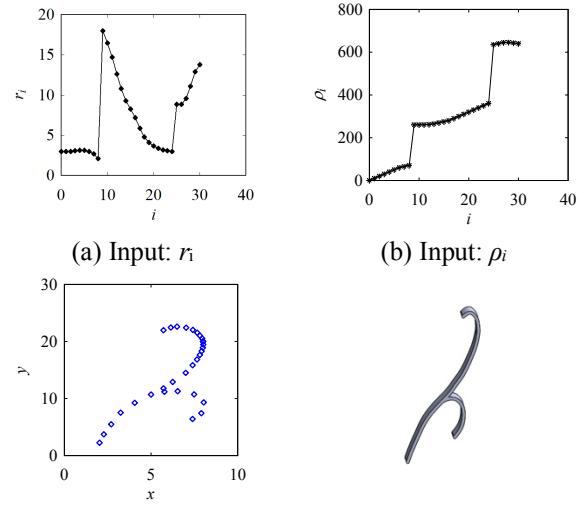


Fig. 10. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Punkar.

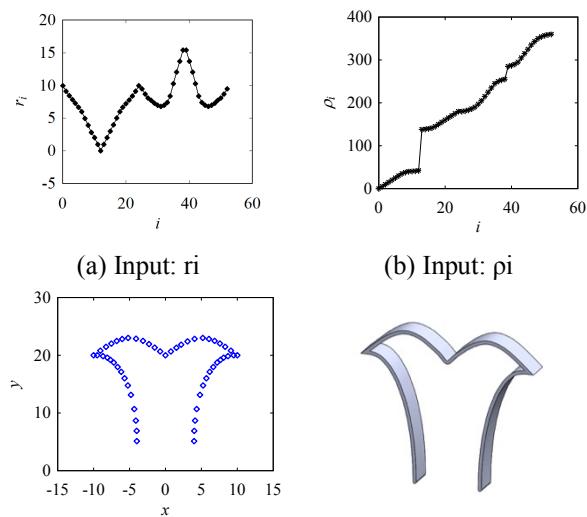


Fig. 11. Geometric and Solid CAD modeling of Hokkaido-Ainu motif called Apapo.

3. Results and Discussions

Based on Hokkaido-Ainu motifs described in Section 2, the two physical models, namely, Model-1 and Model-2 are manufactured using the AM process.

The Model-1 is created based on Hokkaido-Ainu motif called Sik-ureu-morew as shown in Fig. 13(a). As seen in Fig. 13(a), the Model-1 consists of four Sik-uren-morews. First, a single Sik-ureu-morew is created using the proposed algorithm. Then it is rotated three times. The complete point-cloud is shown in Fig. 13(b). The point-cloud of Model-1 is then transferred to commercially available 3D CAD packages for CAD modeling. The result of the solid CAD modeling is shown in Fig. 13(c). Once the CAD modeling is done, the solid CAD model is saved as an STL data file that can be used for building a physical model by AM devices (e.g., 3D printer) as shown Fig. 13(d). In this case, an ordinary 3D printer is used (the details are not shown to avoid commerciality). The physical model built by the 3D printer is shown in Fig. 13(e).

The Model-2 is created based on Hokkaido-Ainu motif shown in Fig. 14(a). As seen Fig. 14(a), the Model-2 consists of Morew, Sik, Ayus, ellipses, curves, and lines. The point-cloud of motifs are created using the proposed algorithm. The complete point-cloud is shown in Fig. 14(b). The solid CAD modeling, triangulation modeling, and AM process of Model-2 are performed similarly to the Model-1.

It is seen from the Model-1 and Model-2, one can combined main motifs, synthetic motifs, or plant motifs to create other significant artifacts based on Hokkaido-Ainu motifs. Hokkaido-Ainu motifs look complex at first glance, however, by analyzing the geometric characteristics of each motif, it is found that all the motifs are symmetrical. The point-cloud data created for the motifs are not limited to a 3D printer or AM process, it can be used integrated with other software and devices (e.g., 2D printer) for creating decorative objects or souvenirs. Thus, the proposed method for geometric modeling and manufacturing can be used in the wide

range of applications other than manufacturing the artifacts having culturally significant objects.

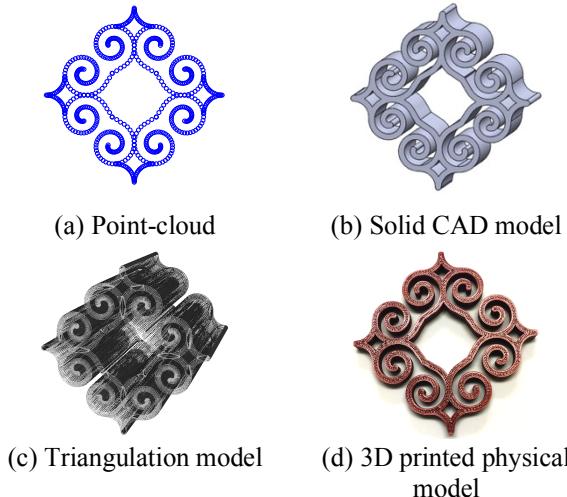


Fig. 13. Model-1 based on Hokkaido-Ainu motifs called Sik-uren-morew.

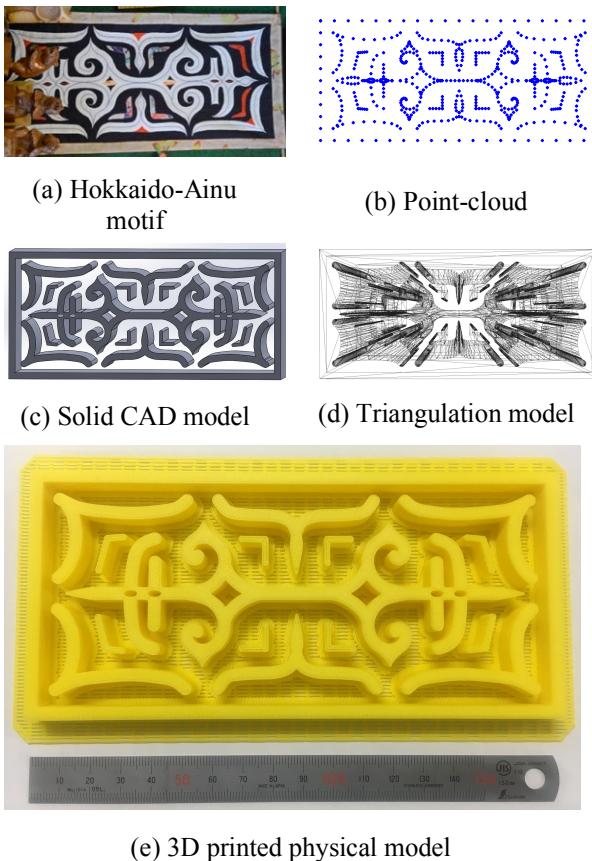


Fig. 14. Model-2 based on Hokkaido-Ainu motifs.

4. Conclusion

The paper presented the geometric modeling of Hokkaido-Ainu motifs and its manufacturing method. Hokkaido-Ainu motifs are identified and classified into four motifs, namely, main motifs, synthetic motifs, plant motifs, and other motifs. An algorithmic approach has been implemented for the geometric modeling of Hokkaido-Ainu motifs. The algorithm recursively creates points using the user-defined radius of curvature and angle of rotation. The ten basic Hokkaido-Ainu

motifs have been digitized using the proposed algorithm. The solid models of the motifs have been produced using a commercially available CAD package. The basic motifs have been combined using the standard geometric modeling operations (e.g., rotation and displacement) to create more complex motifs which are often seen on the houses, clothing, ornaments, utensils, and spiritual goods of Hokkaido-Ainus. For Instance, in one of the complex motifs, the motifs called Sik and Uren-Morew have been used. In another complex motif, the motifs called Sik, Uren-Morew, and Ayusi have been used. The CAD models and the replicas of the respective complex motifs have also been produced, which exhibits the effectiveness of the proposed method. The models developed in this paper are based on 2D point-cloud to 2.5D model and the next phase of the study will investigate the modeling of 2D point-cloud data to a 3D model.

5. References

- [1] S. C. H. Cheung, "Ainu culture in transition," *Futures*, vol. 35, no. 9, pp. 951-959, 2003/11/01/2003. DOI: [https://doi.org/10.1016/S0016-3287\(03\)00051-X](https://doi.org/10.1016/S0016-3287(03)00051-X).
- [2] H. Ōnishi, "The Formation of the Ainu Cultural Landscape: Landscape Shift in a Hunter-Gatherer Society in the Northern Part of the Japanese Archipelago," *Journal of World Prehistory*, vol. 27, no. 3, pp. 277-293, December 01 2014. DOI: 10.1007/s10963-014-9080-2.
- [3] T. Várády, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-Aided Design*, vol. 29, no. 4, pp. 255-268, 1997. DOI: [https://doi.org/10.1016/S0010-4485\(96\)00054-1](https://doi.org/10.1016/S0010-4485(96)00054-1).
- [4] D. W. James, F. Belblidia, J. E. Eckermann, and J. Sienz, "An innovative photogrammetry color segmentation based technique as an alternative approach to 3D scanning for reverse engineering design," *Computer-Aided Design and Applications*, vol. 14, no. 1, pp. 1-16, 2017. DOI: <https://doi.org/10.1080/16864360.2016.1199751>.
- [5] N. Krznar, A. Pilipović, and M. Šcerer, "Additive Manufacturing of Fixture for Automated 3D Scanning – Case Study," *Procedia Engineering*, vol. 149, no. Supplement C, pp. 197-202, 2016. DOI: <https://doi.org/10.1016/j.proeng.2016.06.656>.
- [6] M. Paulic, T. Irgolic, J. Balic, F. Cus, A. Cupar, T. Brajlih, and I. Drstvensek, "Reverse Engineering of Parts with Optical Scanning and Additive Manufacturing," *Procedia Engineering*, vol. 69, no. Supplement C, pp. 795-803, 2014. DOI: <https://doi.org/10.1016/j.proeng.2014.03.056>.
- [7] I. Gibson, D. Rosen, and B. Stucker, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, 2nd ed., Springer-Verlag New York, 2015. DOI: <https://doi.org/10.1007/978-1-4939-2113-3>.
- [8] M. K. Thompson, G. Moroni, T. Vaneker, G. Fadel, R. I. Campbell, I. Gibson, A. Bernard, J. Schulz, P. Graf, B. Ahuja, and F. Martina, "Design for

- Additive Manufacturing: Trends, opportunities, considerations, and constraints," *CIRP Annals*, vol. 65, no. 2, pp. 737-760, 2016. DOI: <https://doi.org/10.1016/j.cirp.2016.05.004>.
- [9] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. L. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, "The status, challenges, and future of additive manufacturing in engineering," *Computer-Aided Design*, vol. 69, pp. 65-89, 2015. DOI: <https://doi.org/10.1016/j.cad.2015.04.001>.
- [10] R. Nooran, *3D printing: technology, applications, and selection*, 1st ed., Boca Raton, FL, CRC Press, Taylor & Francis Group, 2018.
- [11] B. Camille, "What are you printing? Ambivalent emancipation by 3D printing," *Rapid Prototyping Journal*, vol. 21, no. 5, pp. 572-581, 2015. DOI: <https://doi.org/10.1108/RPJ-09-2014-0128>.
- [12] R. Scopigno, P. Cignoni, N. Pietroni, M. Callieri, and M. Dellepiane, "Digital Fabrication Techniques for Cultural Heritage: A Survey," *Computer Graphics Forum*, vol. 36, no. 1, pp. 6-21, 2017. DOI: <https://doi.org/10.1111/cgf.12781>.
- [13] Tashi, A. S. Ullah, M. Watanabe, and A. Kubo, "Analytical Point-Cloud Based Geometric Modeling for Additive Manufacturing and Its Application to Cultural Heritage Preservation," *Applied Sciences*, vol. 8, no. 5, pp. 656(1)-656(18), 2018. DOI: <https://doi.org/10.3390/app8050656>.
- [14] M. Hess and S. Robson, "Re-engineering Watt: A case study and best practice recommendations for 3D colour laser scans and 3D printing in museum artefact documentation," in *Lacona IX - Lasers in conservation*, British Museum, London, 2013, pp. 154-162.
- [15] R. A. Neumüller M., Rist F., Kern C., "3D Printing for Cultural Heritage: Preservation, Accessibility, Research and Education, Ioannides M., Quak E., (Eds.)," *3D Research Challenges in Cultural Heritage*, vol. 8355, no. Lecture Notes in Computer Science., pp. 119-134, 2014. DOI: https://doi.org/10.1007/978-3-662-44630-0_9.
- [16] R. Furferi, L. Governi, Y. Volpe, L. Puggelli, N. Vanni, and M. Carfagni, "From 2D to 2.5D i.e. from painting to tactile model," *Graphical Models*, vol. 76, no. 6, pp. 706-723, 2014/11/01/ 2014. DOI: <https://doi.org/10.1016/j.gmod.2014.10.001>.
- [17] K. Hasegawa, L. Li, N. Okamoto, S. Yanai, H. Yamaguchi, A. Okamoto, and S. Tanaka, "Application of Stochastic Point-Based Rendering to Laser-Scanned Point Clouds of Various Cultural Heritage Objects," *IJAT*, vol. 12, no. 3, pp. 348-355, 2018.
- [18] NA, (2016). Introduction of Traditional Crafts: Ainushiriki. Available online: <http://www.city.sapporo.jp/shimin/pirka-kotan/jp/kogei/ainu-siriki/> (Accessed on January 15, 2018).

Creating Digital Twin of Processed Surface using the Concept of Markov Chain for Industry 4.0

Angkush Kumar Ghosh [†] AMM Sharif Ullah [‡] Michiko Watanabe [‡] Akihiko Kubo [‡]

[†] Kitami Institute of Technology/Graduate School of Engineering, Kitami, Japan

[‡] Kitami Institute of Technology/Faculty of Engineering, Kitami, Japan

m1752108013@std.kitami-it.ac.jp

Abstract: Digital Twins (DTs) are the computable virtual abstractions of real-world objects, processes, and phenomena. They are needed to fulfill the functional requirements of Industry 4.0 i.e., understanding (why it is happening), prediction (what will happen), and adaptation (decision-making for choosing the right course of action). However, constructing phenomena twins is a cumbersome task because most of the manufacturing phenomena (e.g., surface roughness, cutting force, tool wear, and alike) are stochastically nonlinear in nature. This study shows that the phenomena twins of processed surfaces (surfaces created by machining, grinding, and alike) consist of knowledge extraction and simulation component. In this study, for extracting the knowledge, the concept of Markov chain is utilized to encapsulate the dynamics underlying the processed surface heights by using some discrete latent states and their transition probabilities. For constructing the simulation component, a Monte Carlo simulation-based procedure is developed that simulates the surface heights stochastically based on the extracted knowledge. The effectiveness of the proposed methodology is also shown by conducting a case study wherein the surface heights due to multi-pass grinding operation are recreated. The quantitative comparisons between the real and simulated surface heights are also presented in detail. The constructed Digital Twin of the processed surface is also represented using the semantic web, as the manufacturing sector will soon face an era of Web 3.0/4.0 where manufacturing knowledge will be represented and shared using semantic web (or web of concept maps). As a result, the outcomes of this study also provide some insights into knowledge representation for developing the artificially intelligent manufacturing systems for the next generation.

Keywords: Industry 4.0, Digital Twin, Processed Surface, Surface Roughness, Markov Chain, Transition Probability, Monte Carlo Simulation, Semantic Web, Concept Map

1. Introduction

The manufacturing sector has been reinventing itself throughout the time, as schematically illustrated in Fig. 1. As seen in Fig. 1, in Industry 1.0, the manufacturing sector witnessed mechanization by utilizing the steam engine-based devices. In the following stage (i.e., Industry 2.0), the productivity was enhanced by introducing mass production assembly lines. In Industry 3.0, the manufacturing sector witnessed automation of manufacturing tasks by introducing the numerically controlled devices. Nowadays, Industry 3.0 has been transforming into an era called Industry 4.0. It involves the Internet more explicitly than before to fulfill the following three functional requirements: 1) understanding (why it is happening), 2) prediction (what will happen), and 3) adaptation (decision-making for choosing the right course of action) [1], [2]. To do so, the Internet and web technologies have been contributing a lot to evolve the manufacturing sector in rapid shape [3]. From the viewpoint of Industry 4.0, the manufacturing enablers (e.g., devices, systems, machine tools, operators, and alike) that are needed to perform the manufacturing activities will be brought under the umbrella of Internet and web so that the enablers can exchange the manufacturing contents (e.g., knowledge, information, and so on) whenever required to achieve the above-mentioned functional requirements [4].

Now it is evident that to achieve the functional

requirements of Industry 4.0 or to bring all the enablers under the umbrella of Internet and web means to build a virtual-world or a cyber-world from which the physical-world or real-world of manufacturing can be steered [1]. It is worth mentioning that a set of systems, populated by the contents called Digital Twins (DTs), are needed as the contents in that cyber-world for achieving the above-mentioned functional requirements of Industry 4.0 [5], [6].

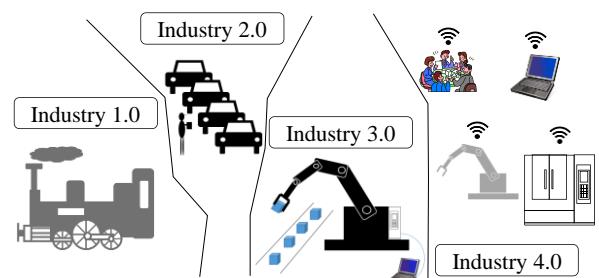


Fig. 1 Evolution of manufacturing sector.

The DTs are the computable virtual abstractions of the real-world objects, processes, and phenomena. As seen in Fig. 2, whenever a manufacturing activity is performed by the manufacturing enablers, it contains certain objects or shapes, processes, process sequences, and thereby results phenomena (e.g., tool wear, tool temperature, surface roughness, cutting force, and alike). Thus, from the viewpoint of the contents, there are three types of DTs,

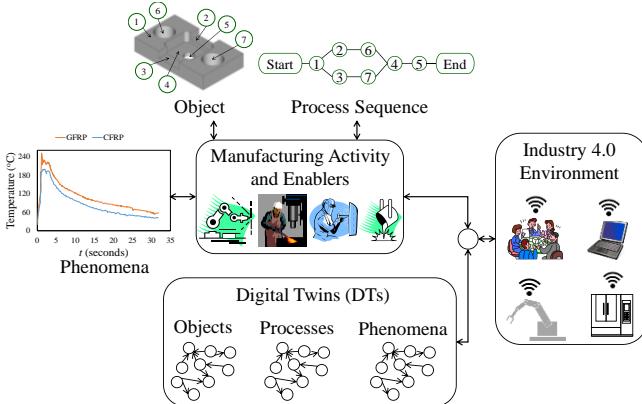


Fig. 2 The context of Digital Twins (DTs).

namely, object twin, process twin, and phenomenon twin. These twins are integrated into the environment of Industry 4.0, as shown in Fig. 2. The information of the manufacturing enablers (i.e., description of the activities, devices, and systems) and the knowledge of manufacturing processes are needed to construct the DTs [7], [8]. At the same time, the information received from the environment of Industry 4.0 (e.g., sensor signals) can be used to construct the DTs.

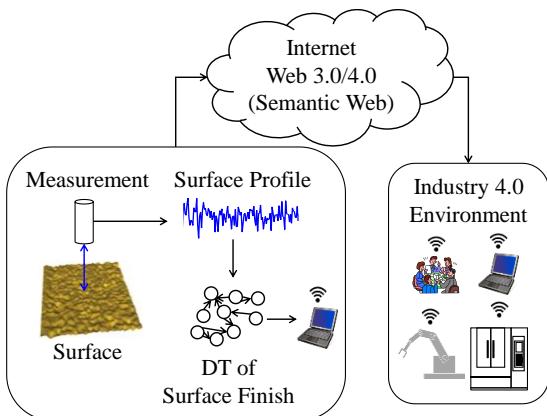


Fig. 3 The context of this study.

However, this study addresses the Digital Twin (DT) of a processed surface i.e., a phenomenon twin. As we know, when a manufacturing process (e.g., grinding, turning, drilling, milling, polishing, laser machining and alike) is performed, the surface roughness is used as a performance measure of the process. As a result, quantifications of the characteristics of the processed surface are essential for all manufacturing processes. This means that when a manufacturing process continues in the environment of Industry 4.0, the system that monitors, characterizes, and provides the processed surface related decisions must be equipped with the relevant DT. Since most of the manufacturing processes are highly complex and exhibit nonlinear phenomena, it is a cumbersome task to create the DT of a processed surface by using an analytical approach. As an alternative, one can construct the DT of a processed surface by using the numerical data of the surface heights. This scenario is shown in Fig. 3. The height information of a processed surface exhibits the underlying dynamics. The DT captures the dynamics (i.e., knowledge extraction from the surface heights) and reproduces the surface heights (i.e., simulation of surface heights) whenever necessary within the context of Industry 4.0. Based on this contemplation, this study utilizes the concept of Markov chain [9] – [11] in

constructing the DT.

In addition, after the knowledge extraction from and simulation of the processed surface, the constructed DT needs to be represented (i.e., knowledge representation) using the semantic web for the next generation manufacturing [3], [4]. This scenario is also shown in Fig. 3. Based on this contemplation, this study also deals with the issue of semantic web representation of the constructed DT.

The remainder of this article is organized as follows: Section 2 provides a literature review on Industry 4.0 and DTs. Section 3 describes the methodology to construct the DT of a processed surface by utilizing the concept of Markov chain. Section 4 provides a case study showing how to implement the proposed methodology. This section also discusses the results obtained and describes the semantic web representation of the constructed DT. Section 5 provides the concluding remarks of this study.

2. Literature Review

Numerous authors have defined the concept of Industry 4.0 in different ways. However, it can be described best by its features, namely, 1) Interconnection and Interoperability, 2) Virtualization, 3) Autonomy and Decentralization, 4) Real-time Capability, 5) Service Orientation, and 6) Modularity. Interconnection and Interoperability mean seamless connectivity and data communication among all manufacturing enablers (e.g., machines, humans, sensor, actuators, and alike) [2], [12], [13]. Virtualization means monitoring the manufacturing activities through modeling, simulation, and data analytics [13], [14]. Autonomy and Decentralization mean decentralized and autonomous decision-making capabilities of the manufacturing enablers [12], [15]. Real-time Capability means real-time data processing using data analytics [12], [16], [17]. Service Orientation means integration and sharing of services offered by the stakeholders in the whole value chain of manufacturing (e.g., design, production, marketing, provision of after-sales service, and so on) [7], [18]. Modularity means flexibility to deal with the changes (e.g., change in process planning, change in product design, and alike) by analysis and decision-making [7], [18]. The above-mentioned features of Industry 4.0 encourages the fact that the physical-world and cyber-world underlying the manufacturing sector must be merged. This evolves the concept of Cyber-Physical Systems (CPS) [1], [13], [19]. In CPS, the physical-world refers to the physical objects that are needed to perform the manufacturing activities (e.g., machines, tools, sensors, physical networks among computing devices, actuators, robots, computers, and alike), whereas the cyber-world refers to the computational entities that are needed to get the best from the physical objects (e.g., historical data, real-time data, information, software, data analytics, knowledge-based systems, algorithms, decision-making systems, and alike) [16], [20]. It is worth mentioning that CPS has evolved some technologies, namely, Radio Frequency Identification (RFID) [20], Internet of Things (IoT) [16], [21], Internet of Services (IoS) [16], Internet of People (IoP) [22], Internet of Content and Knowledge (IoCK) [22] and so on. Some of these technologies are inclined to the physical-world and some others are inclined to the cyber-world. Nevertheless, they help achieve a seamless merger between the physical and cyber-worlds, as schematically illustrated in Fig. 4. As

mentioned above (also can be seen in Fig. 4), the cyber-world consists of historical data, real-time data, information, software, data analytics, knowledge-based systems, algorithms, decision-making systems, and alike. Apart from these entities, the entities called DTs are the useful constituents of the cyber-world of CPS [7], [12]. The relative position of the DTs is schematically shown in Fig. 4.

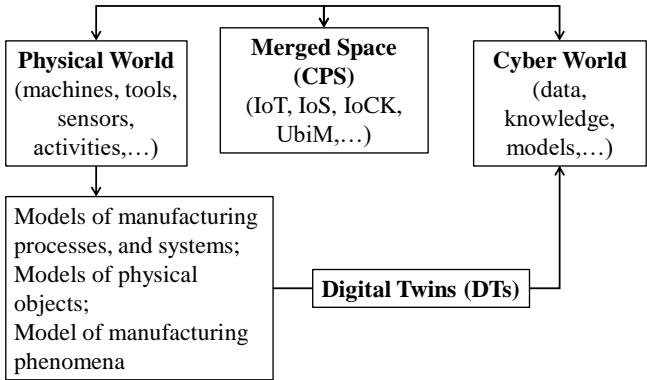


Fig. 4 The relative position of DTs.

Here, the DTs refer to the computable virtual abstractions of the physical objects, processes, and phenomena [5], [6]. They can simulate the physical-world in the cyber-world and thereby help make the right decisions for determining the course of action to ensure the desired performance of manufacturing activities [23], [24]. This means that without the help of the DTs, the functional requirements of Industry 4.0 (i.e., understanding, prediction, and adaptation) cannot be achieved. Therefore, the cyber-world must be populated with the DTs along with other contents (e.g., historical data, real-time data, and alike). Nevertheless, the DTs can be classified into three broad categories, namely, object twins, process twins, and phenomena twins. Object twins (e.g., machine tool, conveyors, product, and alike) have already been used in the systems of Industry 4.0 [6], [7]. Similarly, the process twins have been used in production planning (e.g., layout configuration, labor allocation, production capacity, product servitization, and process planning) [8], [23], [25]. On the other hand, the phenomena twins (e.g., cutting force, surface finish, tool wear, and alike) have not yet been studied in detail [4] compared to the object and process twins. For this reason, this study addresses the issue of constructing phenomena twins.

Apart from the importance of constructing the DTs, the issue of semantic web representation (i.e., knowledge representation) of the constructed DTs is required to be studied because semantic web will be used to functionalize the connections of CPS in near future [26]. Semantic web emphasizes on the meaning of the contents exchanged through the Internet and will play a key role in achieving the web-embedded intelligent devices in the years to come. For this reason, the constructed DTs must be systemized using the semantic web technology. In this case, the concept mapping can be an option for the semantic web representation of the manufacturing knowledge [3], [4].

3. Methodology

This section deals with the methodology for creating the DT of surface finish. Here, surface finish is primarily

defined by using the heights of a processed surface. The methodology is schematically illustrated in Fig. 5. As seen in Fig. 5, there are three segments denoted as “1”, “2”, and “3”. The segments “1” and “3” deal with the issues of the physical-world, whereas the other segment “2” deals with the issues of the cyber-world. In the physical-world (denoted as “1”), the grinding experiments take place where a grinding wheel removes material from a workpiece surface under the given grinding conditions (e.g., grinding wheel velocity, work surface velocity, depth of cut, feed rate, and so on). When the experiment is completed, the surface heights of the workpiece surface denoted as z_{ij} can be measured by using a non-contact surface measuring instrument or any other means. This is one of the results. The set consisting of all experimental data of surface heights is denoted as $SD = \{z(o)_{ij} \mid o = 1, \dots, O, i = 0, 1, \dots, n, j = 1, \dots, m\}$ that stores the historical data in the cyber-world (denoted as “2”). Suppose that there is a computational arrangement that learns from the SD and simulates the surface heights denoted as zs_{ij} for a given set of grinding conditions in the cyber-world (denoted as “2”). In addition, consider that another grinding operation is being carried out in the physical-world (denoted as “3” as schematically illustrated in Fig. 5) and the surface finish is being measured in the form of surface heights denoted as zm_{ij} by a monitoring system. The monitored surface heights (zm_{ij}) can be compared with the simulated ones (zs_{ij}) for making a decision, i.e., whether to continue the operation or change the grinding conditions.

The above explanation refers to the fact that the computational arrangement must extract the knowledge from the historical data of surface heights, represent the knowledge for reuse, and simulate the surface heights whenever necessary. Therefore, this computational arrangement is the DT of the processed surface.

The general structure of the DT of a processed surface consists of three modules, namely, 1) Knowledge Extraction Module, 2) Knowledge Representation Module, and 3) Simulation Module. The Knowledge Extraction Module can use rule-, dynamical system-, or feature-based approach. The goal is to capture the dynamics underlying the surface heights stored in the SD. The Knowledge Representation Module can use either the conventional web-based approach or semantic web-based approach to represent the extracted knowledge. Finally, the Simulation Module can use either the Monte Carlo and/or deterministic simulation-based approach to simulate the surface heights zs_{ij} whenever needed. In the rule-based approach, the dynamics underlying the surface heights can be captured by using a set of *if...then...* rules [27]. The rules can be computed in the Simulation Module by using a Monte Carlo simulation approach. A suitable web technology can represent the rules as well as the simulation tool, as described in [27], [28]. In the dynamical system-based approach, the dynamics underlying the surface heights can be captured by using a dynamical system, e.g., a dynamical system based on Q-sequence as described in [29]. The dynamical system can be represented by suitable web technology if needed. The Simulation Module can be developed by integrating the Monte Carlo and deterministic simulation-based approaches, as described in [29]. On the other hand, in the feature-based approach, the dynamics underlying the surface heights can be captured by using a set of features, namely, trend, irregularity, burst, and cycle, as described in [30]. The features can be represented by suitable web technology if

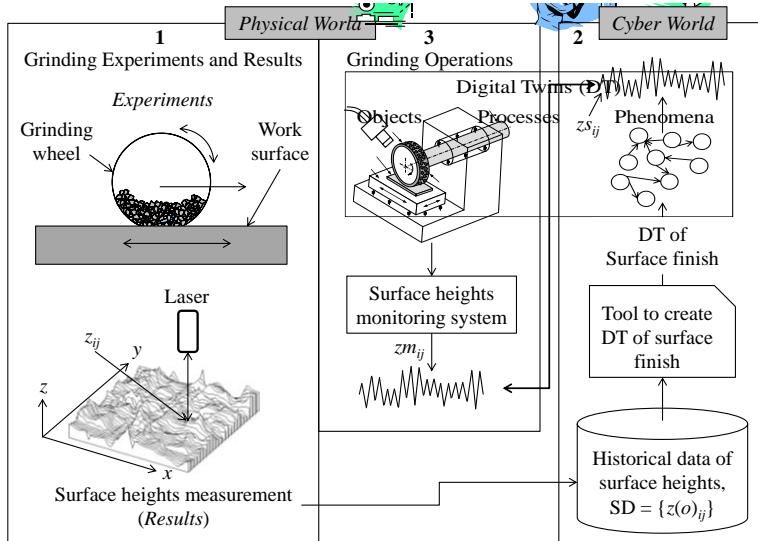


Fig. 5 DT of surface finish and its implication.

needed. The Simulation Module can be developed by integrating the Monte Carlo and deterministic simulation-based approach, as described in [4], [30].

However, in this study, a different approach is proposed that is based on the Markov chain for capturing the underlying dynamics of the surface heights. The general framework of the proposed procedure is schematically illustrated in Fig. 6.

As seen in Fig. 6, let $z_{ij} \in \mathbb{R}$, $i = 0, 1, \dots, N$, $\exists j \in \mathbb{N}$ be the surface heights of the roughness or primary profile of a processed surface. Let $z_{nij} \in [0, 1]$ be the normalized value of z_{ij} . Therefore, the following expression holds:

$$z_{nij} = \frac{z_{ij} - \min_{i=0,\dots,N}(z_{ij})}{\max_{i=0,\dots,N}(z_{ij}) - \min_{i=0,\dots,N}(z_{ij})} \quad (1)$$

Let A , B , C , D , and E be five mutually exclusive intervals in $[0, 1]$ so that $A < B < C < D < E$ and $A \cup B \cup C \cup D \cup E = [0, 1]$. The most straightforward definitions of the intervals are as follows: $A = [0, 0.2)$, $B = [0.2, 0.4)$, $C = [0.4, 0.6)$, $D = [0.6, 0.8)$, and $E = [0.8, 1]$ as schematically illustrated in Fig. 6. These intervals are the states of z_{nij} . This means that if $z_{nij} \in A$, then the state of z_{nij} is A . If $z_{nij} \in B$, then the state of z_{nij} is B . If $z_{nij} \in C$, then the state of z_{nij} is C . If $z_{nij} \in D$, then the state of z_{nij} is D . Finally, if $z_{nij} \in E$, then the state of z_{nij} is E .

Consider a return map consisting of the points $(z_{nij}, z_{n,i+1,j})$, $i = 0, \dots, N-1$. The reason for considering the return map is that it helps understand the underlying dynamics of the surface heights. The dynamics can be represented by the probabilities of the transitions from all possible states (i.e., latent states of Markov chain). This scenario is shown in Fig. 6. Each number represents the probability of the Markov process changing from one state to another state, with the direction indicated by the arrow. For example, if the surface height is in state A , then the probability it changes to states B , C , D , and E is 0.2, 0.2, 0.1, and 0.2 respectively, while the probability it remains in state A is 0.3 as shown in Fig. 6. Similar contemplation holds for other states also. Therefore, the states have transition probabilities that can be denoted as $Pr(X_{i+1,j} | Y_{ij})$ where $\forall X_{i+1,j} \in \{A_{i+1,j}, B_{i+1,j}, C_{i+1,j}, D_{i+1,j}, E_{i+1,j}\} \in [0, 1]$, $\forall Y_{ij} \in \{A_i, B_i, C_i, D_i, E_i\}$, $A_{i+1,j} = A_i = A$, $B_{i+1,j} = B_i = B$, $C_{i+1,j} = C_i = C$, $D_{i+1,j} = D_i = D$, and $E_{i+1,j} = E_i = E$ as shown in Fig. 6. These transition probabilities create an entity defined as

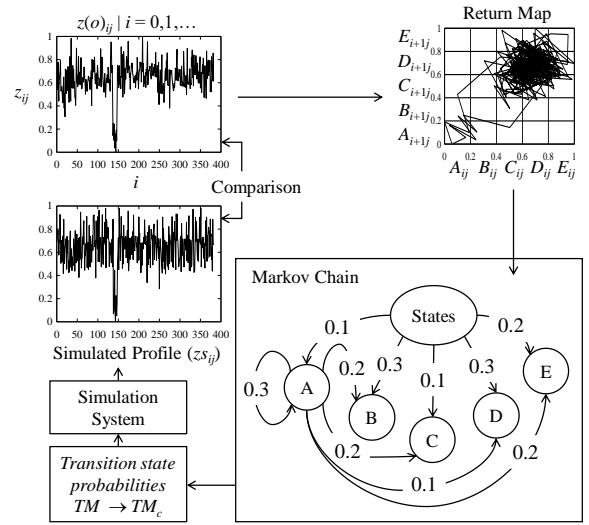


Fig. 6 Markov Chain-based procedure of DT.

Transition Probability Matrix (TM) as follows:

$$TM = \begin{bmatrix} Pr(A_{i+1,j}|A_{ij}) & \dots & Pr(E_{i+1,j}|A_{ij}) \\ \vdots & \ddots & \vdots \\ Pr(A_{i+1,j}|E_{ij}) & \dots & Pr(E_{i+1,j}|E_{ij}) \end{bmatrix} \quad (2)$$

Each transition state has a cumulative probability denoted as $Prc(...)$, resulting in a matrix defined as Transition Cumulative Probability Matrix (TM_c) as follows:

$$TM_c = \begin{bmatrix} Prc(A_{i+1,j}|A_{ij}) & \dots & Prc(E_{i+1,j}|A_{ij}) \\ \vdots & \ddots & \vdots \\ Prc(A_{i+1,j}|E_{ij}) & \dots & Prc(E_{i+1,j}|E_{ij}) \end{bmatrix} \quad (3)$$

In Eq. (3), $Prc(A_{i+1,j}|A_{ij}) = Pr(A_{i+1,j}|A_{ij})$, $Prc(B_{i+1,j}|A_{ij}) = Pr(B_{i+1,j}|A_{ij}) + Prc(A_{i+1,j}|A_{ij})$, $Prc(C_{i+1,j}|A_{ij}) = Pr(C_{i+1,j}|A_{ij}) + Prc(B_{i+1,j}|A_{ij})$, $Prc(D_{i+1,j}|A_{ij}) = Pr(D_{i+1,j}|A_{ij}) + Prc(C_{i+1,j}|A_{ij})$, $Prc(E_{i+1,j}|A_{ij}) = Pr(E_{i+1,j}|A_{ij}) + Prc(D_{i+1,j}|A_{ij}) = 1$. For other transition states, the similar formulation holds. Now, one can create the following intervals for the sake of simulation: $a(A_{ij}) = [0, Prc(A_{i+1,j}|A_{ij}))$, $b(A_{ij}) = [Prc(A_{i+1,j}|A_{ij}), Prc(B_{i+1,j}|A_{ij}))$, $c(A_{ij}) = [Prc(B_{i+1,j}|A_{ij}), Prc(C_{i+1,j}|A_{ij}))$, $d(A_{ij}) = [Prc(C_{i+1,j}|A_{ij}), Prc(D_{i+1,j}|A_{ij}))$, and $e(A_{ij}) = [Prc(D_{i+1,j}|A_{ij}), Prc(E_{i+1,j}|A_{ij}))$.

For the sake of simulation, a normal distribution-based approach can be employed as described below. Let $Nor_X(\mu_X, \sigma_X)$ be a normally distributed variable with mean μ_X and standard deviation σ_X where $X \in \{A, B, C, D, E\}$. Thus, one can randomly simulate the surface heights $zsij$ according to the Surface Height Simulation Algorithm (SHSA) as shown in Table 1.

4. Case Study

The previous section describes the methodology for creating the DT of a processed surface. In this section, the proposed methodology is applied to create the DT of a processed surface due to grinding. The experimental data of the surface heights denoted as $SD = \{z(o)ij | o = 1, \dots, O, i = 0, 1, \dots, n, j = 1, \dots, m\}$ are collected from the work described in [31]. The sets of historical data of surface heights denoted as Case 1, ..., Case 4, as seen in Figs. 7-10, are the surface heights of the same work surface area after the first, second, third, and fourth pass of the grinding wheel, respectively. The underlying grinding conditions (grinding wheel velocity,

work surface velocity, depth of cut, and feed rate) are described in [31].

Table 1 Surface Height Simulation Algorithm (SHSA)

1	<i>Randomly assign</i>	$zs_{i=0j} \leftarrow [0,1]$
2	<i>Iterate</i>	For $i = 0, \dots, N$
3	<i>Determine the state of zs_{ij}</i>	$zs_{ij} \in A \rightarrow X_{ij} = A_{ij};$ $zs_{ij} \in B \rightarrow X_{ij} = B_{ij};$ $zs_{ij} \in C \rightarrow X_{ij} = C_{ij};$ $zs_{ij} \in D \rightarrow X_{ij} = D_{ij};$ $zs_{ij} \in E \rightarrow X_{ij} = E_{ij}$
4	<i>Generate random number</i>	$r_{ij} \in [0,1]$
5	<i>Simulate the next state zs_{i+1j}</i>	$r_{ij} \in a(X_{ij}) \rightarrow X_{i+1j} = A_{i+1j}, zs_{i+1j} \leftarrow Nor_A(\mu_A, \sigma_A);$ $r_{ij} \in b(X_{ij}) \rightarrow X_{i+1j} = B_{i+1j}, zs_{i+1j} \leftarrow Nor_B(\mu_B, \sigma_B);$ $r_{ij} \in c(X_{ij}) \rightarrow X_{i+1j} = C_{i+1j}, zs_{i+1j} \leftarrow Nor_C(\mu_C, \sigma_C);$ $r_{ij} \in d(X_{ij}) \rightarrow X_{i+1j} = D_{i+1j}, zs_{i+1j} \leftarrow Nor_D(\mu_D, \sigma_D);$ $r_{ij} \in e(X_{ij}) \rightarrow X_{i+1j} = E_{i+1j}, zs_{i+1j} \leftarrow Nor_E(\mu_E, \sigma_E)$

For example, consider the historical data of the surface heights corresponding to Case 1. The surface heights are normalized to produce $zn_{ij} \in [0,1]$ in accordance with Eq. (1). The return map consisting of the points $(zn_{ij}, zn_{i+1j}), i = 0, \dots, N-1$, is also constructed. The transition states are as follows: $A = [0,0.2], B = [0.2,0.4], C = [0.4,0.6], D = [0.6,0.8]$, and $E = [0.8,1]$, and $A \cup B \cup C \cup D \cup E = [0,1]$. In accordance with Eq. (2), the Transition Probability Matrix (TM) underlying Case 1 is given by the Eq. (4).

$$TM_{(Case\ 1)} = \begin{bmatrix} 0 & 0.5 & 0.25 & 0.25 & 0 \\ 0 & 0.192 & 0.577 & 0.212 & 0.019 \\ 0.004 & 0.142 & 0.696 & 0.130 & 0.028 \\ 0.043 & 0.058 & 0.594 & 0.275 & 0.030 \\ 0 & 0.091 & 0.273 & 0.545 & 0.091 \end{bmatrix} \quad (4)$$

In accordance with Eq. (3), the Transition Cumulative Probability Matrix (TM_c) corresponding to Case 1 is given by Eq. (5). Similar calculation holds for other cases as well.

$$TM_c_{(Case\ 1)} = \begin{bmatrix} 0 & 0.5 & 0.75 & 1 & 1 \\ 0 & 0.192 & 0.769 & 0.989 & 1 \\ 0.004 & 0.146 & 0.842 & 0.972 & 1 \\ 0.043 & 0.101 & 0.695 & 0.970 & 1 \\ 0 & 0.091 & 0.364 & 0.909 & 1 \end{bmatrix} \quad (5)$$

For the simulation, the values of the mean of the respective states are set as follows: $\mu_A = 0.1, \mu_B = 0.3, \mu_C = 0.5, \mu_D = 0.7$, and $\mu_E = 0.9$. The values of the standard deviation are set as follows: $\sigma_A = \sigma_B = \sigma_C = \sigma_D = \sigma_E = 0.05$. Using the results of TM_c , the intervals of cumulative probabilities denoted as $a(Y_{ij}), b(Y_{ij}), c(Y_{ij}), d(Y_{ij}),$ and $e(Y_{ij})$ ($\forall Y \in \{A, B, C, D, E\}$) are calculated. For example, for Case 1, when the transitions from A_{ij} to other possible states are considered, the cumulative probabilities are as follows: $a(A_{ij}) = [0,0.5], b(A_{ij}) = [0.5, 0.75], c(A_{ij}) = [0.75, 1], d(A_{ij}) = 1$, and $e(A_{ij}) = 1$. Needless to say, it corresponds to the first row in Eq. (5).

The above values are used to simulate the surface

heights by using the procedure shown in the 4-th and 5-th steps in SHSA. Figures 7-10 show the results of the simulation in terms of time series and return maps corresponding to Cases 1,...,4 respectively. For some cases, the simulated surface heights resemble the real surface heights (e.g., Fig. 10). For some other cases, the simulated surface heights do not entirely resemble the real surface heights. This means that the presented DT is applicable when the grinding operation has stabilized after multiple passes. Having said that, it does not necessarily mean that the DT produces quantitatively similar surface heights. For ensuring the quantitative similarity between the real and simulated surface heights, it is important to compare them by using a standard surface roughness parameter, e.g., R_a . As such, the values of R_a are calculated for the real and simulated surface heights. The results are shown in Fig. 11.

The simulations have been carried out for 100 times for each case. The straight line having the unit slope (shown by the blue color) is the ideal line that shows that the simulated R_a is equal to the real R_a . As seen in Fig. 11, for all cases, more than 50% of the simulated results are greater than that of the real ones. This means that the DT simulates a rougher surface than the real one. Therefore, it is necessary to incorporate other computing means so that the simulated surface heights become as smoother as the real one. On the other hand, the return maps can be compared quantitatively by using a mathematical entity called possibility distribution [32], which is a probability-distribution-neutral representation of uncertainty. The results corresponding to each case are shown in Fig. 12. As seen in Fig. 12, all the possibility distributions of the simulated surface heights resemble the possibility distributions of the real surface heights. This means that the return maps are quantitatively the same or the presented DT of a processed surface can produce reliable results.

Nevertheless, Figure 13 shows the screen-shot of the Excel® spreadsheet-based computing tool that implements the simulation process. As seen in Fig. 13, the user needs to input the number of data points of real surface heights, the data of real surface heights ($z(i)$), define the number of latent states, and standard deviation (σ). The computing tool does the rest. It provides the time series and return maps corresponding to the real and simulated surface heights, as well as, the respective possibility distributions for the sake of comparison.

However, Fig. 14 shows the screen-shot of semantic web representation (i.e. a concept map or Cmap) of the DT of a processed surface which is created using CmapTools®. One can access this concept map from the following URL and integrate it to the systems of Industry 4.0: https://cmapscloud.ihmc/DT_of_processed_surface. It is worth mentioning that Fig. 14 contains the following propositions: 1) grinding operation requires grinding wheel, workpiece, grinding machine, grinding conditions, surface metrology instrument; 2) surface metrology instrument measures surface heights to represent surface roughness; 3) surface heights are normalized to construct digital twin (DT) of surface roughness using Markov chain; 4) Markov chain consists of user defined latent states for example A, B, C, D, E that have state probabilities; 5) transition of surface heights from one of the latent states to A, B, C, D, E have transition probabilities; 6) Markov chain is used to construct the computing tool to model and simulate surface heights. The concepts (i.e., grinding operation, grinding wheel, workpiece, grinding machine, grinding conditions, surface

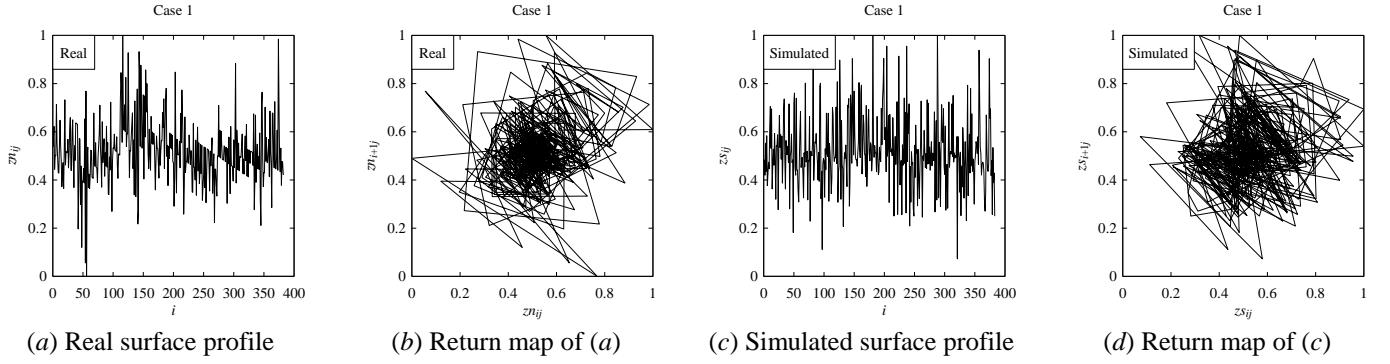


Fig. 7 Results corresponding to Case 1 [31].

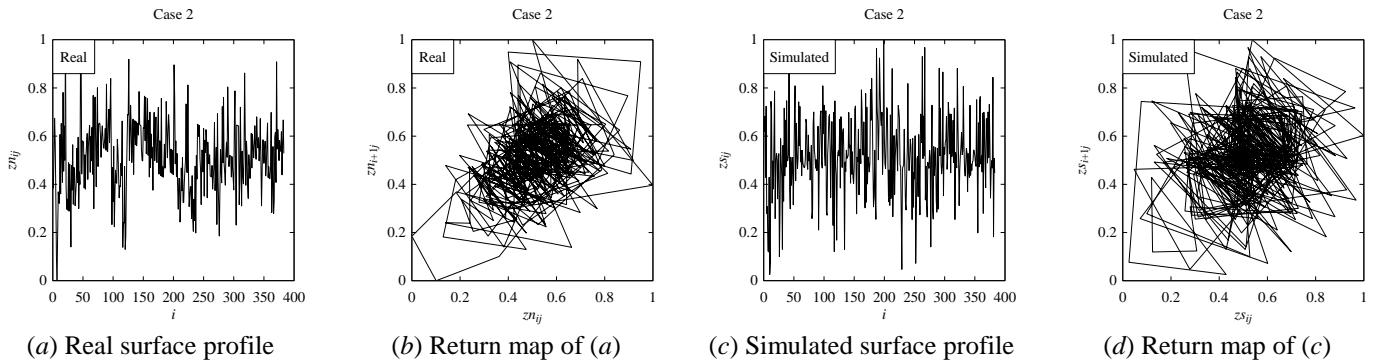


Fig. 8 Results corresponding to Case 2 [31].

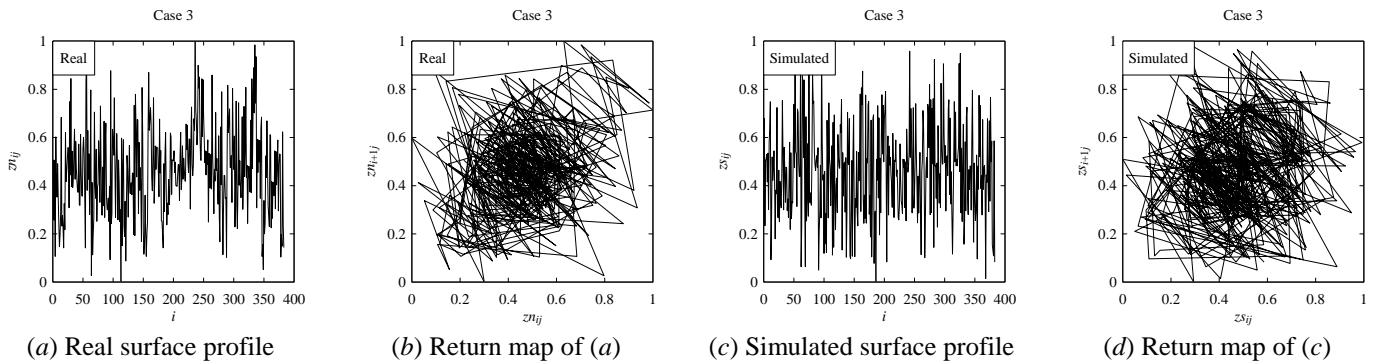


Fig. 9 Results corresponding to Case 3 [31].

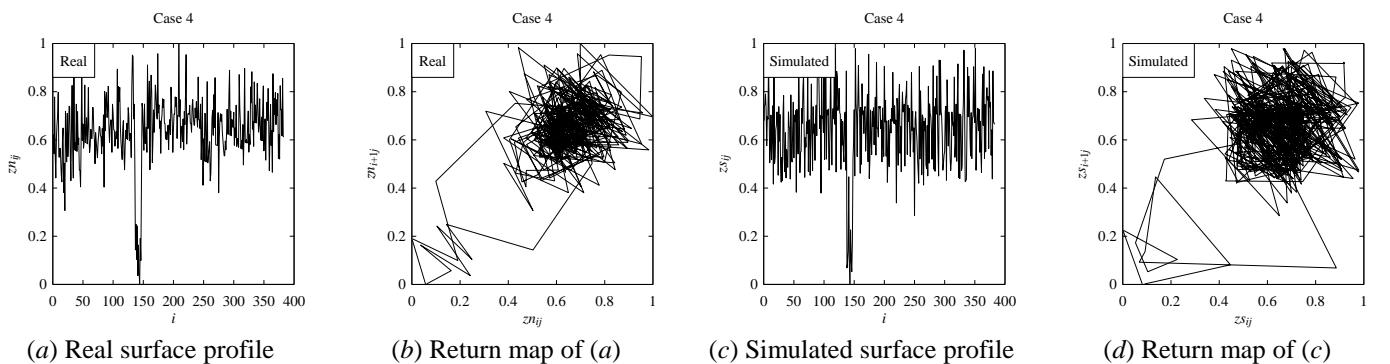


Fig. 10 Results corresponding to Case 4 [31].

metrology instrument, surface heights, digital twin (DT), computing tool, surface roughness, Markov chain, user defined latent states, transition, state probabilities, and transition probabilities) in this concept map can be interlinked with each other as well as other concept maps for better understanding if required. For example, with the concept of “grinding conditions” one can link another concept map that illustrates what are grinding conditions and how they affect the grinding operation and so on. The concept of “computing tool” can be linked to a source from

where the user can download the computing tool (e.g., the computing tool shown in Fig. 13). It can also be linked to another concept map that illustrates how to use the computing tool and so on.

5. Concluding Remarks

In this study, Markov chain-based knowledge extraction procedure and a Monte Carlo-based simulation system are

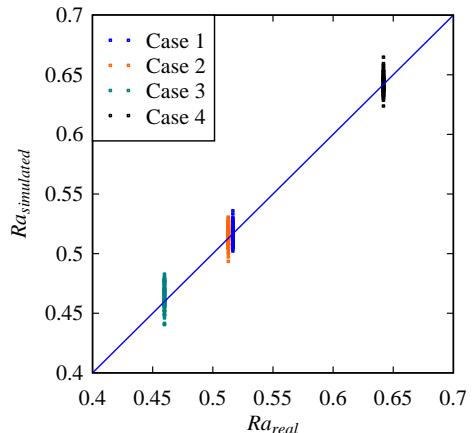


Fig. 11 Comparison between the real and simulated surface roughness.

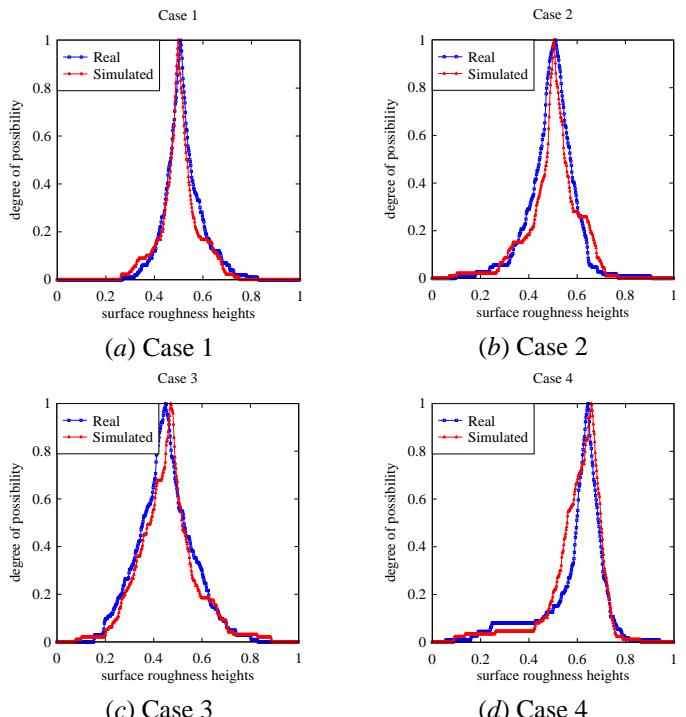


Fig. 12 Possibility distributions of the real and simulated surface heights.

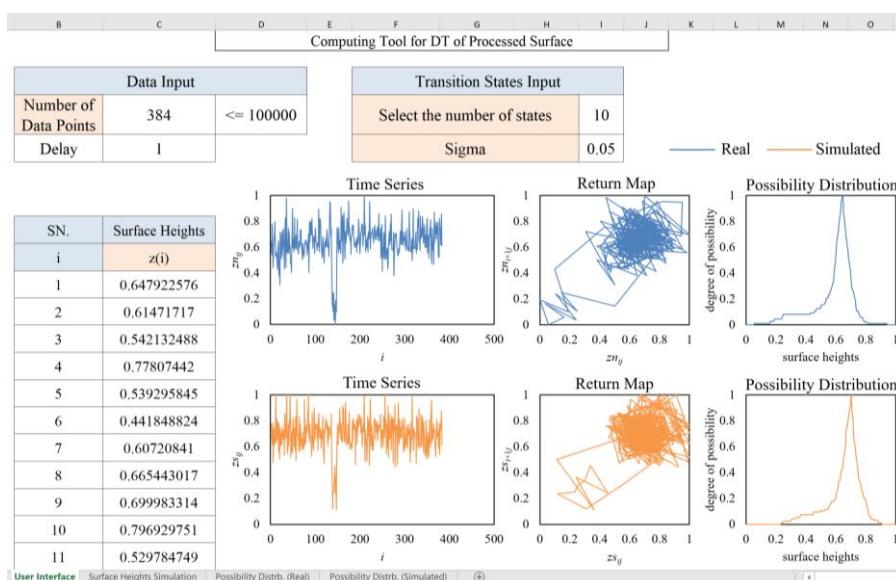


Fig. 13 Computing tool for the DT of processed surface.

developed to create the DT of a processed surface. The presented procedure has been applied to create a DT of multi-pass grinding operation. In addition, the semantic web representation of the proposed methodology through Cmap is also carried out. Nevertheless, the DT creates a rougher surface compared to the real ones. Further investigation is needed to solve this problem. In addition, the simulation system will be incorporated with the semantic web shown in Fig. 14 for utilizing the constructed DT in the systems relevant to Industry 4.0 (e.g., cyber-physical system).

References

- [1] H. Kagermann, W. Wahlster, and J. Helbig,

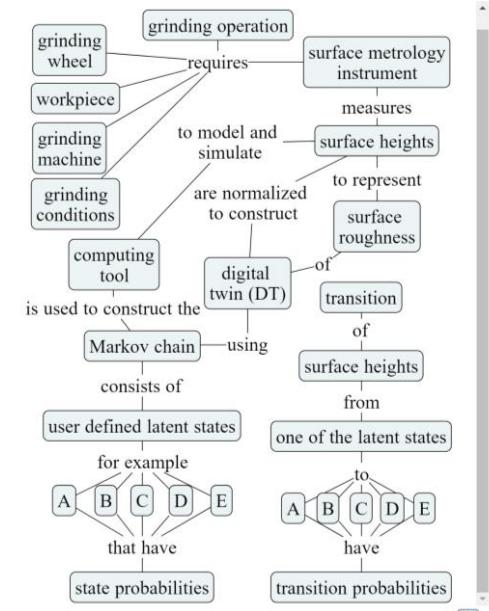


Fig. 14 Cmap of the constructed DT.

“Recommendations for implementing the strategic initiative INDUSTRIE 4.0”, in Final Report of the Industrie 4.0 Working Group, acatech (National Academy of Science and Engineering), 2013.

- [2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0”, Bus. Inf. Syst. Eng., Vol. 6, No. 4, pp. 239 – 242, 2014.
- [3] A.M.M. Sharif Ullah, N. Arai, M. Watanabe, “Concept map and internet-aided manufacturing”, Procedia CIRP, Vol. 12, pp. 378 – 383, 2013.
- [4] X. Wang, A. S. Ullah, and A. Kubo, “Modeling and Simulation of Nonlinear Phenomena for Internet-aided Manufacturing”, proceedings of the Asia Pacific

- [5] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems", in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham: Springer International Publishing, pp. 85 – 113, 2017.
- [6] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data", *Int. J. Adv. Manuf. Technol.*, Vol. 94, No. 9–12, pp. 3563 – 3576, 2018.
- [7] S. I. Shafiq, C. Sanin, E. Szczerbicki, and C. Toro, "Virtual engineering object / virtual engineering process: A specialized form of cyber physical system for industrie 4.0", *Procedia Comput. Sci.*, Vol. 60, No. 1, pp. 1146 – 1155, 2015.
- [8] F. Tao, M. Zhang, J. Cheng, and Q. Qi, "Digital twin workshop: a new paradigm for future workshop", *Comput. Integr. Manuf. Syst.*, Vol. 23, No. 1, pp. 1 – 9, 2017.
- [9] A. M. Fraser, "Hidden Markov models and dynamical systems", Philadelphia: SIAM, 2008.
- [10] A. Petropoulos, S. P. Chatzis, and S. Xanthopoulos, "A hidden markov model with dependence jumps for predictive modeling of multidimensional time-series", *Information Sciences*, Vol. 412 – 413, pp. 50 – 66, 2017.
- [11] Z. Li, H. Fang, M. Huang, Y. Wei, and L. Zhang, "Data-driven bearing fault identification using improved hidden Markov model and self-organizing map", *Computers and Industrial Engineering*, Vol. 116, pp. 37 – 46, 2018.
- [12] A. Kusiak, "Smart manufacturing", *Int. J. Prod. Res.*, Vol. 56, No. 1–2, pp. 508 – 517, 2018.
- [13] Y. Cheng, Y. Zhang, P. Ji, W. Xu, Z. Zhou, and F. Tao, "Cyber-physical integration for moving digital factories forward towards smart manufacturing: a survey", *Int. J. Adv. Manuf. Technol.*, Vol. 97, No. 1 – 4, pp. 1209 – 1221, 2018.
- [14] P. O'Donovan, K. Leahy, K. Bruton, and D. T. J. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities", *J. Big Data*, Vol. 2, No. 1, pp. 1 – 26, 2015.
- [15] J. Lee, B. Bagheri, and H.-A. Kao, "Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics", proceedings of the Int. Conference on Industrial Informatics (INDIN), 2014.
- [16] X. Yao, J. Zhou, Y. Lin, Y. Li, H. Yu, and Y. Liu, "Smart manufacturing based on cyber-physical systems and beyond", *J. Intell. Manuf.*, pp. 1 – 13, 2017.
- [17] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues", *J. Ind. Inf. Integr.*, Vol. 6, pp. 1 – 10, 2017.
- [18] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review, 2015.
- [19] K.-D. Thoben, S. Wiesner, and T. Wuest, "'Industrie 4.0' and Smart Manufacturing – A Review of Research Issues and Application Examples", *Int. J. Autom. Technol.*, Vol. 11, No. 1, pp. 4 – 19, 2017.
- [20] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing", *CIRP Ann.*, Vol. 65, No. 2, pp. 621 – 641, 2016.
- [21] Y. Lu and J. Cecil, "An Internet of Things (IoT)-based collaborative framework for advanced manufacturing", *Int. J. Adv. Manuf. Technol.*, Vol. 84, No. 5–8, pp. 1141 – 1152, 2015.
- [22] X. Yao and Y. Lin, "Emerging manufacturing paradigm shifts for the incoming industrial revolution", *Int. J. Adv. Manuf. Technol.*, Vol. 85, No. 5–8, pp. 1665 - 1676, 2016.
- [23] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the DT for design and production engineering", *CIRP Ann.*, Vol. 66, No. 1, pp. 141 – 144, 2017.
- [24] S. Boschert and R. Rosen, "Digital Twin—The Simulation Aspect", in *Mechatronic Futures*, P. Hohenberger and D. Bradley, Eds. Cham: Springer International Publishing, pp. 59 - 74, 2016.
- [25] P. Centobelli, R. Cerchione, and T. Murino, "Layout and Material Flow Optimization in Digital Factory", *Int. J. Simul. Model.*, Vol. 15, No. 2, pp. 223 – 235, 2016.
- [26] L. Ramos, "Semantic Web for manufacturing, trends and open issues: Toward a state of art", *Computers & Industrial Engineering*, Vol. 90, No. 1, pp. 444 – 460, 2015.
- [27] A. M. M. S. Ullah and K. H. Harib, "Knowledge extraction from time series and its application to surface roughness simulation", *Inf. Knowl. Syst. Manag.*, Vol. 5, No. 2, pp. 117 – 134, 2006.
- [28] A. M. M. Sharif and K. H., "Zadehian Paradigms for Knowledge Extraction in Intelligent Manufacturing", in *Manufacturing the Future*, Pro Literatur Verlag, Germany / ARS, Austria, No. July, pp. 291 – 308, 2006.
- [29] A. M. M. Ullah, "Surface Roughness Modeling Using Q-Sequence", *Math. Comput. Appl.*, Vol. 22, No. 2, pp. 33 – 44, 2017.
- [30] A. M. M. Sharif Ullah, J. Tamaki, and A. Kubo, "Modeling and simulation of 3D surface finish of grinding", *Adv. Mater. Res.*, Vols. 126-128, pp. 672 – 677, 2010.
- [31] A. M. M. S. Ullah, A. Caggiano, A. Kubo, and M. A. K. Chowdhury, "Elucidating grinding mechanism by theoretical and experimental investigations", *Materials*, Vol. 11, No. 2, pp. 274 – 292, 2018.
- [32] A. M. M. Sharif Ullah and M. Shamsuzzaman, "Fuzzy Monte Carlo Simulation using point-cloud-based probability–possibility transformation", *Simulation*, Vol. 89, No. 7, pp. 860 – 875, 2013.

El Farol Bar problem by Agreement algorithm based on trial and error behavior at the macro level

Nhuai Phung Masao Kubo Hiroshi Sato

Department of Computer Science
National Defense Academy of Japan
ed17006@nda.ac.jp

Abstract: In this paper, we apply BRT algorithm, which is a value-free method to the best-of- n problem (called BSTn problems), to El Farol Bar problem in order to verify its effectiveness. The BSTn problem is one of the collective decision-making problems whereby a group of distributed agents is required to select the best group state among a set of n options, and the algorithms that can solve more complicated problems ($n \gg 2$) have been required in recent years. The BRT algorithm is a BSTn algorithm that can be used in a dynamic environment even if the quality of the option is not well-known and is known as high scalability for the number of agents and number of options n . In this paper, we focus on the advantage of BRT that can deal with many options and show that the group's state can be optimized at a high level. In the previous algorithms of $n = 2$, in order to accurately determine the state of the group consisting of multicomponent, a separate complicated mechanism using this in multiple stages was necessary. The El Farol Bar problem is a kind of complimentary game used in distributed AI and Game theory in which a numerical optimality is required to optimize the number of people going to the bar. We extended the BRT algorithm in order to solve this problem and verified its effectiveness using computer experiments.

Keywords: The best-of- n problem, El Farol Bar problem, Swarm robotics, Multi-agent, Value-free

1. Introduction

In recent years, as robots and software agents have become popular, the research on behavior of a group consisted of a large number of individuals in collective decision-making has been actively studied. Optimality of the system achieved by individual decision-making is not only an important issue for engineering applications but also for understanding our own society more deeply. Therefore, it has been actively focused in various fields of including Game theory, multi-agent system, and distributed AI. However, in recent years it is necessary to understand the behavior of the decision-making in more complicated situations and environment, and models more sophisticated are required.

One of this issue is the best-of- n problem in which a group of people has to select the best option among a set of n alternatives without a leader [1]. Nearly all previous studies focused on binary decision-making scenarios [2, 3, 4, 7]. Thus, it was difficult to analyze the case of $n \gg 2$. On the other hand, if we can deal with this case ($n \gg 2$), it can be expected that the adaptive ability of the group will be improved dramatically. In the previous models focused on binary decision-making scenario, the group could only handle 2 modes: *Everyone agrees* and *Everyone opposes*, but it could also handle another mode as a delicate state of a group

such as 90% *agrees* or 91% *agrees*. Figuratively speaking, a restaurant which has only 2 menus will not accept almost of all requests from customers. However, if a restaurant has many menus it can satisfy all customers demands. A large number of alternatives can change its performance qualitatively.

Against this background, in recent years approaches called *value-sensitive* and *value-free* have been proposed. The value-sensitive approach models collective decision-making by nest-site selection process by honeybee swarms [9, 10, 11, 12], in which the quality of options is represented by the number of individuals supported. It is experimentally clarified that if the group size is sufficiently large for n , microscopic decision-making, for example, cross-inhibition [9], can enable a subpopulation that selected the best option among multiple ($n > 2$) options to become a majority eventually [11, 12].

Meanwhile, the value-free approach has the feature that a group can achieve its macrostate with a good value by discovering the quality value of options simultaneously even if the value is unknown in advance. For example, Iwanaga et al.[4] proposed an individual decision-making inspired by T.Shering's critical mass [13], and they also clarified that the group quickly converges to the state where everyone selects one of the 2 options. At the time of convergence, the group

knows the value of the selected option quality for the first time. Then, if the value is high enough, the selected option will be the final result of the collective decision-making, otherwise, individuals can quit selecting this option and start selecting another option in order to achieve a better macrostate. We have proposed a collective decision-making model (BRT model, Bias and Rising Threshold model) in which by introducing the *give up* term to individual's decision-making process even if there are a lot of options ($n \gg 2$), the suitable macrostate will be achieved [5, 6]. However, these previous studies dealt with the problem that the number of options n is relatively small, and its superiority as described above has not been clarified.

In this paper, we apply the BRT model to the El Farol Bar problem [14] and clarify its effectiveness. As will be described in detail later, the El Farol bar problem is a distributed decision-making problem to make a number of visitors suitable for the bar's seating capacity, roughly speaking. We will show that a group can make a suitable decision for the capacity of a number of strategies based on the ratio between the number of people going to the bar and staying at home.

This paper is organized as follows. Firstly, we will explain the BRT model. Next, we will describe the El Farol bar problem and explain how to apply the BRT model to this problem. Finally, we indicate the effectiveness of the proposed method using computer experiments.

2. Related Works

2.1 Value-free approach: Iwanaga and Namatame model [4]

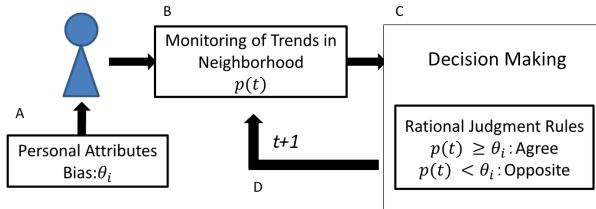


Fig. 1: The role of social skin in individual decision making [4].

In this paper, we use the BRT model inspired by the characteristics of Iwanaga and Namatame model, that is the group rapidly becomes bipolar. Firstly, we explain Iwanaga and Namatame model.

The research by Iwanaga and Namatame [4] got inspiration from Schelling's critical mass research [13]. As is shown in Fig.1, they stated that individual decision-making does not only depend on personal philosophy and personal preferences, but also on the atmosphere of the whole group ($p(t)$). Each individual shall take the position of agreeing or

disagreeing with an opinion and has a personal attribute bias value θ_i (Fig.1A), which is used in this decision-making process (Fig.1C). Each individual observes the ratio of agreed members to the opinion $p(t)$ (Fig.1B).

$$\begin{cases} p(t) \geq \theta_i : \text{Agree} \\ p(t) < \theta_i : \text{Oppose} \end{cases} \quad (1)$$

As shown in Equation 1, an individual i agrees with an option if the percentage of agents who agree with this option $p(t)$ (the agreement ratio) is higher than the bias value θ_i at time t . After all individuals made their decisions, the agreement ratio $p(t+1)$ at time $t+1$ changes (Fig.1D), so the individual makes its decision again (Fig.1C).

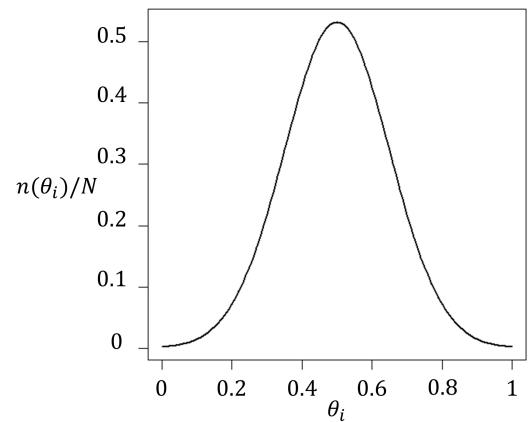


Fig. 2: Bias θ distribution example [4].

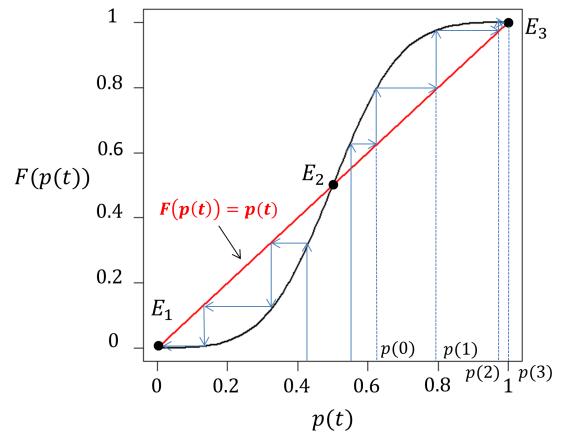


Fig. 3: Cumulative distribution function and Equilibrium point of collective consensus decision (E_1 , E_3 are stable points, E_2 is unstable point [4]).

Iwanaga and Namatame (Iwanaga and Namatame 2002) proposed a method to analyze the dynamics of the agreement ratio $p(t)$ in this decision-making process. If the distribution

of θ is determined as shown in Fig.2, the cumulative distribution function $F(\theta)$ forms an S-shaped curve as shown in Fig.3. If the agent makes a decision according to Equation 1, the ratio $p(t)$ at the time t follows the following equation:

$$F(\theta) = \sum_{\theta_i \leq \theta} n(\theta_i) \quad (2)$$

$$p(t+1) = F(p(t)) \quad (3)$$

$$p^* = F(p^*) \quad (4)$$

Thus, the solutions of the above equation are called fixed points. There are two types of fixed points: a stable point and 2 unstable points. If the bias value θ_i is distributed in a bell shape, as shown in Fig.2, one unstable point (E_2) and two stable points (E_1, E_3) are generated in the cumulative distribution function, and all members smoothly move their opinions to ultimately agree or disagree (Fig.3). In the other words, the behavior of a group can be classified into the following three patterns.

- The case of $p(t) = 0.5$ corresponds to E_2 in Fig.3. State of the swarm does not change because $F(p(t)) = p(t)$. However, if $p(t)$ changes for some reason, the state of the swarm will transit to one of the following two states.
- The case of $p(t) > 0.5$ ($E_2 < p(t) \leq 1$). In this case, since $p(t+1) = F(p(t)) > p(t)$, the state of swarm transits rapidly to the state ($p(t) = 1$) that all agents agree with the option. In the Fig.3, because of $p(0) < p(1) < p(2) < p(3) \approx 1$, the agreement state is reached with approximately 3 units of time from the almost equilibrium state.
- The case of $p(t) < 0.5$ ($0 \leq p(t) < E_2$). Since $p(t+1) = F(p(t)) < p(t)$, the state of swarm transits rapidly to the state ($p(t) = 0$) that all agents disagree with the option.

If the distribution shape of the bias θ_i is same, i.e., the bias values θ_i of all individuals are generated from a same distribution, the number of iterations until convergence does not depend on the number of individuals. Thus, it is possible to expect that the convergence reaches promptly even with a large number of individuals.

However, this model is not expanded to n -alternatives and a failure recovery dynamics is not discussed.

2.2 El Farol Bar Problem [14]

El Farol bar problem is a complementary problem in Game theory and it has become a popular framework of complex systems. The problem was created in 1994 by W. Brian Arthur based on a bar in Santa Fe, New Mexico. It describes the situation where finite population of people has to choose whether to go or not to the El Farol Bar, which is enjoyable

only if it is not too crowded. So much so, in fact, that the preferences of the population can be described as follows:

- If less than 60% of the population go to the bar, they will have a better time than they stayed at home.
- If more than 60% of the population go to the bar, they will have a worse time than they stayed at home.

Everyone has to decide at the same time whether they will go to the bar or not. After deciding to go, they can know how many others go with them. Roughly speaking, this bar is a nice bar so everyone wants to go, but if the number of visitors is not at around 60% of the population, they can not enjoy the bar. Therefore, it is difficult to form a society where many people are satisfied by a group of the same sense of value, and it is taken up as a basis for the study of complex society.

3. Proposed Method

3.1 Mechanism of the best strategy-discovering ability

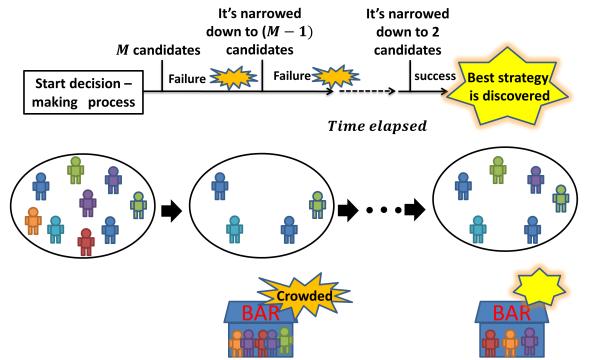


Fig. 4: Mechanism of the suitable strategy-discovering ability.

Fig.4 shows the collective decision-making mechanism of our BRT model. Firstly, one strategy is collectively selected from M candidates and the number of unselected candidates is narrowed down until the suitable strategy is selected. After reaching an agreement, if the group senses that the selected strategy is not the best one (Failure-discovering), all agents will keep making their decisions. After a lapse of time, they will change their strategies and start a new decision-making by themselves. By repeating this process, we can expect to realize the ability to discover the suitable strategy among many candidates.

3.2 BRT model [5]

Based on the above, in the following we propose a model which enable a group of people to collectively agree with

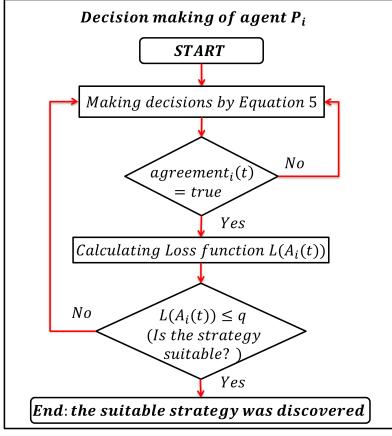


Fig. 5: Decision-making of agent P_i based on BRT model.

a strategy and discover suitable strategy despite of a large number of strategies ($M > 2$), named the BRT model.

Here, we assumed that there is a group of N people $G = \{P_i : i = 1, \dots, N\}$. $A = \{a_j : j = 1, \dots, M\}$ is a set of strategies with $M \geq 2$, which is the number of elements of this set (i.e., the number of strategies). $A_i(t) \in A$ is the strategy of agent P_i at time t . The agent P_i has a bias θ_i ($0 < \theta_i < 1$). $n(a_j)$ is the number of agents selecting the strategy a_j . $A_i(t) \in A$ is the strategy of agent P_i at time t .

Fig.5 shows the decision-making of agent P_i based on BRT model. The agent P_i decides its strategy $A_i(t+1)$ at time t as follows. If

$$n(A_i(t))/N \geq \theta_i + \tau \cdot c_i(t) \cdot (t - t_{i,\text{last}}) \quad (5)$$

is satisfied, $A_i(t+1) = A_i(t)$. Otherwise, a strategy other than $A_i(t)$ is stochastically chosen and becomes $A_i(t+1) \in A \setminus A_i$. Here, we define τ is a constant representing the prediction rate of the increase in number of supporters.

In addition, $t_{i,\text{last}}$ is the time at which the agent P_i last changed its strategy, and $(t - t_{i,\text{last}})$ is the time span that the same strategy continues to be selected.

$$t_{i,\text{last}}(t) = \begin{cases} t & A_i(t) \neq A_i(t-1) \\ t_{i,\text{last}}(t-1) & \text{otherwise} \end{cases} \quad (6)$$

$c_i(t)$ is a function that is equal to 0 or 1 depending on the follows:

$$c_i(t) = c(t) = \begin{cases} 0 & \forall i, \text{agreement}(t) = \text{true} \wedge E(A_i(t)) \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

where $\text{agreement}_i(t)$ is defined as follows:

$$\text{agreement}_i(t) = \text{agreement}(t) = \begin{cases} \text{true} & \forall i, \exists j, A_i(t) = a_j \\ \text{false} & \text{otherwise} \end{cases} \quad (8)$$

The function $E(A(t))$ is a function to judge whether the group is in a good or bad state and is set for each problem. The

$E(A(t))$ function for the El Farol problem will be described later.

If the ratio of agents who select the same strategy as theirs is lower than $\theta_i + \tau \cdot c_i(t) \cdot (t - t_{i,\text{last}})$, the agent selects stochastically a new strategy from other strategies. The second term on the right side of Equation 5 is a term with increasing value over time.

In the previous works [5], BRT model shows a high scalability for M and N but there are no discussions on how to apply this model to an actual problem.

3.3 The distribution of bias θ_i

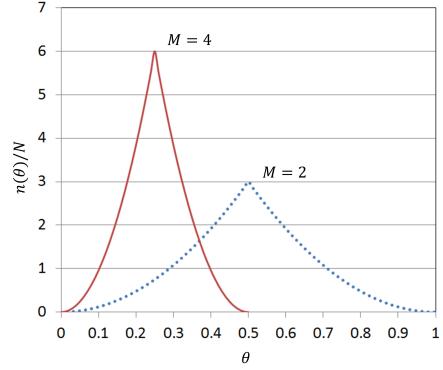


Fig. 6: Bias θ distribution examples.

We propose a bias generating method by quadratic function in this paper. We set θ_i as follows:

$$n(\theta_i) = \begin{cases} 0 & \theta_i \leq 0 \\ k_1 N \theta_i^2 & 0 < \theta_i \leq \frac{1}{M} \\ k_1 N (\theta_i - \frac{2}{M})^2 & \frac{1}{M} < \theta_i < \frac{2}{M} \\ 0 & \frac{2}{M} \leq \theta_i \end{cases} \quad (9)$$

where N is the number of agents, $n(\theta_i)$ represents the number of agents taking θ_i , $\mu = 1/M$ is the expectation of the distribution, and $k_1 = \frac{3M^3}{2}$ is a normalization term. Thus, we derive following equation:

$$\frac{\int_0^1 n(\theta_i) d\theta_i}{N} = 1 \quad (10)$$

Fig.6 illustrates some examples of the distributions of θ determined by Equation 9. As can be seen, the shape of the bias distribution is a peak determined by the number of strategies M only.

3.4 El Farol Bar Problem by BRT model

$A = \{a_j : j = 1, \dots, M\}$ is a set of M strategies. Strategy a_j consists of individual roles: $a_j = \{r_{ij} : i = 1, \dots, N\}$, where r_{ij} is role of agent P_i who selects the strategy a_j . According to the El Farol Bar problem, the agent P_i has 2 roles: $r_{ij} = \{B, H\}$. If $r_{ij} = B$, the agent P_i will go the bar; if $r_{ij} = H$, he

will stay at home (i.e., not going to the bar). Thus, when an agent decides to select a strategy, this means he is required to follow his role included this strategy.

Here, we consider the following El Farol Bar problem. The agent P_i feels good or bad based on how is the bar crowded. n_{el} is the optimal number of visitors which enable them to enjoy the bar most. At time t , the closer the number of people who decided to go to the bar is to the value of n_{el} , the better the state of the group becomes with less unsatisfaction. If more than n_{el} people decide to go to the bar, they will all feel bad because the bar was crowded. If less than n_{el} people decide to go to the bar, people who decide not to go will regret it and feel bad. In both cases, how bad do people feel depends on how crowded the bar is (or how empty the bar is). The more the bar gets crowded (or the more the bar gets empty), the more it increases. Therefore, in this paper we evaluate how bad a person feel using the following simple value function, called *Loss* function:

$$L(A_i(t)) = \frac{|n_b(t)/N - n_{el}/N|}{N} \quad (11)$$

where $n_b(t)$ is the number of people deciding to go to the bar at time t in strategy $A_i(t)$.

Each agent makes its decision according to the BRT model and selects a strategy among a set of M strategies. If everyone selects a same strategy ($agreement(t) = true$), the group state will become agreement state and the information about going to the bar or not described by this strategy of each agent will be sent to the others. Then, the strategy is evaluated as follows:

- if $L(A_i(t)) \leq q$, $A_i(t)$ is considered as a *suitable* strategy a_g (defined later) and the decision-making ends with success.
- if $L(A_i(t)) > q$, $A_i(t)$ is evaluated that it is not the suitable strategy, i.e., the suitable strategy has not been discovered (failure-discovering) and the decision-making is continued (failure-recovering).

Quorum value $q > 0$ is a constant representing the compromise value of the group. In real world, not every society seeks a perfect state. If a state is good to a certain extent, it will be maintained. By adopting a quorum value q , improvement in the decision-making of the group can be expected. The following equations show how we evaluate a strategy in agreement state using q and *Loss* function $E(A_i(t))$:

$$E(A_i(t)) = \begin{cases} true & L(A_i(t)) \leq q \\ false & otherwise \end{cases} \quad (12)$$

We define that the strategies which enable most people to enjoy the bar are suitable strategies, named $A_{goal} = \{a_g \in A, E(a_g) = true\}$. In addition, we also define a strategy that

fits perfectly the optimal number of visitors n_{el} is *best strategy*, called a_{best} . Thus, there may be several suitable strategies, but there is only one best strategy.

$$\forall g, L(a_{best}) \leq L(a_g) \quad (13)$$

4. The Experimental Results

4.1 Settings of the Strategies

Here, based on the El Farol Bar problem, we set the strategies to satisfy that there is always one a_{best} , in which, is among M of strategies. Thus, we set that each strategy a_j ($j \in \{1, \dots, M\}$) has its own the number of agents $n_{b,j}$ who decide to go to the bar as follows:

$$n_{b,j} = \frac{j \cdot N}{M}, j \in \{1, \dots, M\} \quad (14)$$

An example of the settings is shown in Table 1.

Table. 1: An example of strategies setting with $M = 5, N = 10$.

	a_1	a_2	a_3	a_4	a_5
P_1	B	B	B	B	B
P_2	B	B	B	B	B
P_3	H	B	B	B	B
P_4	H	B	B	B	B
P_5	H	H	B	B	B
P_6	H	H	B	B	B
P_7	H	H	H	B	B
P_8	H	H	H	B	B
P_9	H	H	H	H	B
P_{10}	H	H	H	H	B
n_y/N	0.2	0.4	0.6	0.8	1

In addition, we assume that all agents select a random strategy in initial time. We used these settings for all of the following experiments.

4.2 Example of Collective Dynamics of BRT Model

Here, by computer simulation, we show the typical dynamics of the proposed model in the El Farol Bar problem. We set $N = 300, M = 5, \tau = 0.003, n_{el} = 0.6 \cdot N$ and $q = 0.03$. According to Equation 14, a_3 is the best strategy (a_3 is also a suitable strategy). Fig.7 and Fig.8 show the transition example in the case of $M = 5$. In Fig.7, the horizontal axis is time, and the transition of the number agents selecting each is shown by the vertical axis. The result shows that the agreement states are achieved 3 times and at the 3rd time, the suitable strategy is discovered as strategy a_3 .

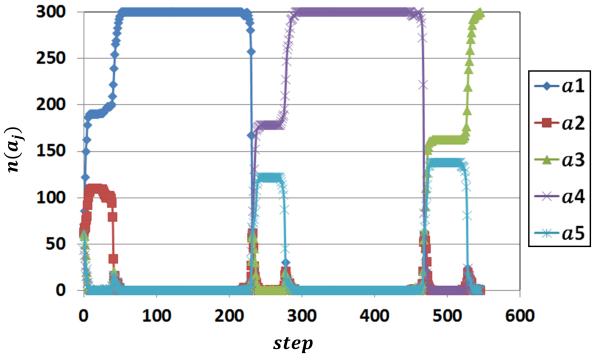


Fig. 7: The transition of number agents selecting each strategy. $M = 5$, $N = 300$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$ and $q = 0.03$.

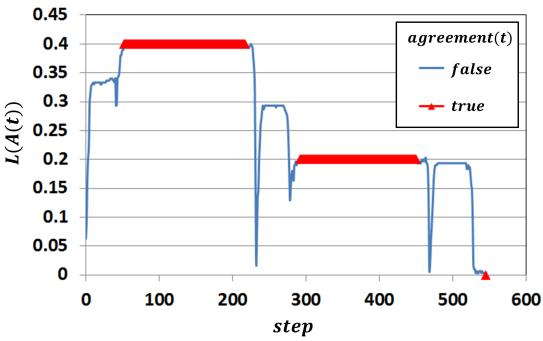


Fig. 8: The transition of the proportion of agents selecting to go to the bar. $M = 5$, $N = 300$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$ and $q = 0.03$.

Fig. 8 shows a transition of *Loss* function $L(A(t))$. The blue line shows the value of *Loss* function, that the group doesn't know, when the agreement state has not been achieved ($agreement(t) = false$). At this time, since all agents change their selected strategy almost every time step, the value of $L(A(t))$ fluctuates greatly. The red Δ mark shows a transition of *Loss* function value when the agreement state has been achieved. At this time, the strategy is evaluated by $L(A(t))$ to discover the suitable strategy. After 2 times failure-discovering, strategy a_3 is discovered as the suitable strategy at about step 540.

From above, it was found that the group can realize the ability to discover for the suitable strategy at the macro level by the proposed method.

4.3 Scalability in group size

In order to discover the suitable strategy, the group have to make an collective decision on a strategy. In the example of the previous subsection, the time required for an agreement, which is the time span from the initial time to the end of the agreement, seems to be a constant value that not depend on

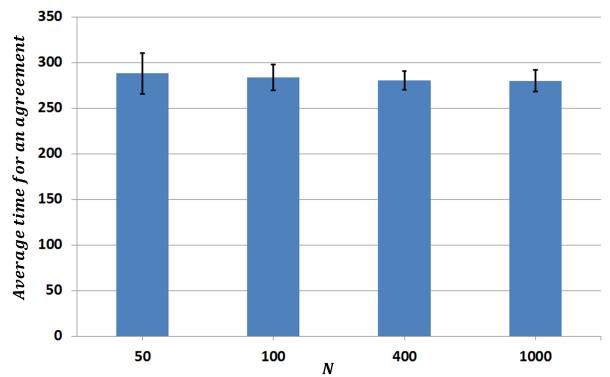


Fig. 9: The average time for an agreement across different group size N . $M = 10$, $\tau = 0.003$, 10000 trials.

the group size N . Here, we do an experiment to verify this prediction. We set $M = 10$, $\tau = 0.003$. At the initial time, assume that each individual selects its strategy randomly. We measured the time required for an agreement from the initial time in the cases of $N = 50, 100, 400, 1000$. Fig. 9 shows the results of performing 10000 trials. The horizontal axis represents the group size N , the vertical axis represents the average time for an agreement. As can be seen, the time required for an agreement is almost a constant value regardless of the value of N .

From the above, we found that the proposed method allows the group make a collective decision in which the time required for an agreement does not depend on the group size N .

4.4 The ability of the suitable strategy-discovering and its features

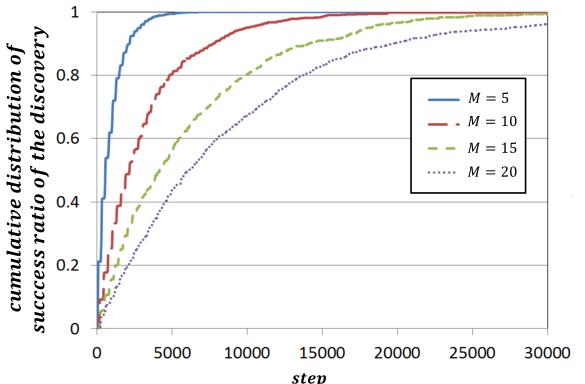


Fig. 10: Cumulative distribution of success ratios of the discovery across different M . $N = 300$, $\tau = 0.003$, $q = 0.03$, $n_{el} = 0.6 \cdot N$.

Here, we show that a group can discover the suit-

able strategy among multiple strategies using the proposed method by computer experiments. The experimental setups are $N = 300$, $\tau = 0.003$, $q = 0.03$, $n_{el} = 0.6 \cdot N$. With each value of M , the number of cases where all of the agents selected a_x was counted for all 1000 trials, and the probability of discovery was determined. The experimental results are shown in Fig.10. We define a successful decision as the suitable strategy a_g was collectively discovered, and success ratio $R_{success}$ is the proportion of the successful decisions in 1000 trials.

$$R_{success} = \frac{n_{success}}{n_{trials}} \quad (15)$$

where $n_{success}$ is the number of success decisions and n_{trials} is the number of trials.

Fig.10 shows cumulative distribution of success ratios of the discovery across different M . The horizontal axis represents the number of strategies time required for the discovery, the vertical axis represents the cumulative distribution of the success ratios. As can be seen, in all of the cases of different M , the cumulative distribution of the success ratios were very high value ($> 96\%$). Thus, we found that the proposed method could enable the group to discover the suitable strategy even when $M \gg 2$.

From the above, the proposed method has a high scalability for the number of strategies M .

4.5 Influence of Quorum q value

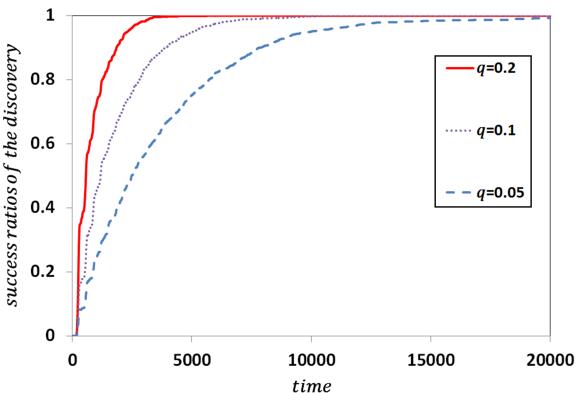


Fig. 11: Cumulative distribution of success ratios of the discovery across different q value. $M = 20$, $N = 100$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$.

Next, we consider how q affects the decision-making. q is the threshold of the suitability of the bar defined Equation 12. The experimental setups are $M = 20$, $N = 100$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$ and $q = 0.05, 0.1, 0.2$. The smaller q is, the more difficult to feel good. With each value of q , we measured the time taken to discover the suitable strategies, and counted the number of trials where the suitable strategies were discovered. Fig.11 shows the cumulative distribution of

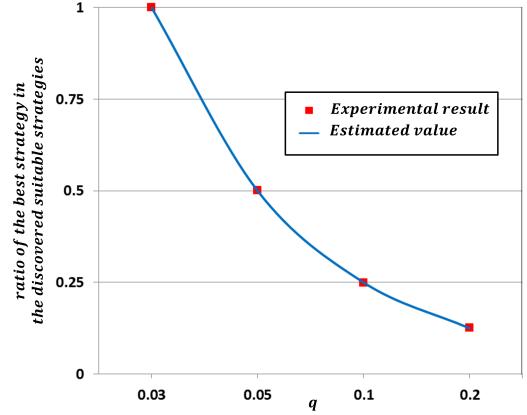


Fig. 12: The relationship between q and the ratio of the number of trials where a_{best} was discovered among the number of trials where the suitable strategy was discovered. $M = 20$, $N = 100$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$, $span = 50000$, 10000 trials.

the ratios of the discovery for each of 1000 trials. As a result, the group was able to discover the suitable strategies at any value of q .

In addition, the larger q value is set, higher success ratios were shown. The reason is, when q value is very small, the discovered strategy is the best strategy a_{best} with high probability. And when q value is set to higher value, there are more strategies $A(t)$ that satisfy $L(A(t)) < q$ as suitable strategies a_g . In the other works, the number of suitable strategies that can be discovered among M strategies depends on q value. In order to consider how q affects the relationship between the best strategy and the suitable strategies, we set $q = 0.03, 0.05, 0.1, 0.2$, counted the number of trials where the discovered strategy was the best strategy a_{best} , and measured the time taken to discover the suitable strategies in 10000 trials. The experimental setups are $M = 20$, $N = 100$, $\tau = 0.003$, $n_{el} = 0.6 \cdot N$, $span = 50000$. The results were shown in Fig.12 and Fig.13.

In this experiment, based on the Equation 14 the difference of 2 strategies in ratio of people who will go to bar is $\frac{n_{b,j+1}}{N} - \frac{n_{b,j}}{N} = \frac{1}{M} = 0.05$. Thus, when $q = 0.03$, only the best strategy a_{best} will be discovered because $L(A(t)) > q$ with any strategies except for a_{best} . When $q = 0.05$, the best strategy and 1 strategy will be discovered as the suitable strategy. Likewise, in the cases of $q = 0.1$ and $q = 0.2$, the number of discoverable strategies can be estimated to be 4 and 8, respectively. Thus, the ratios of the number of trials where a_{best} will be discovered in the cases of $q = 0.03, 0.05, 0.1, 0.2$ are $1, 1/2, 1/4$ and $1/8$, respectively.

Fig.12 shows relationship between q and the ratio of the number of trials where a_{best} was discovered among the number of trials where the suitable strategy was discovered. The

horizontal axis represents q , the vertical axis represents the ratio. \square mark shows the experimental result and blue line shows the above estimated values are shown. As can be seen, the obtained experimental result is the same as the estimated value.

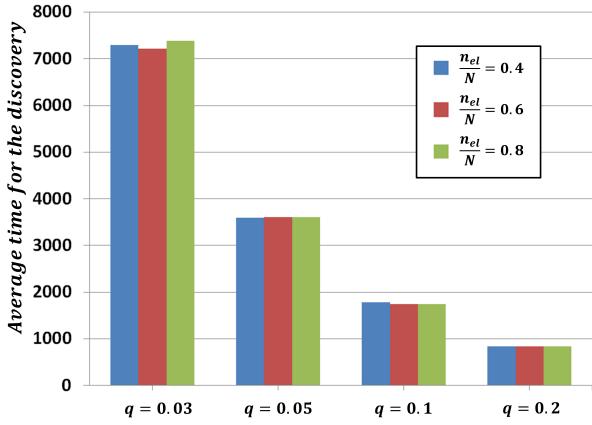


Fig. 13: The relationship between q value and time required for the discovery across different optimal value n_{el} . $M = 20, N = 100, \tau = 0.003, span = 50000, 10000$ trials.

Fig.13 shows the relationship between q value and average time required for the discovery across different optimal value n_{el} . Even if n_{el} was set to a different value, it was found that almost the same results were obtained. Additionally, the average time to discovery suitable strategies tends to be shortened by q , and its ratio seems to be about the same as the discovery rate shown in Fig.12. This experiment suggests that the time required for the discovery is almost proportional to the reciprocal of the number of the stable strategies, and if the number is the same, it does not change even if the ideal visitor number n_{el} of the bar changes.

From the above, we found that the group is able to discover the suitable strategies at any value of q . In addition, it is possible to discover the suitable strategy in a shorter time by adjusting q .

5. Conclusion

In this paper, we proposed a value-free agreement algorithm which enables groups to discover the suitable strategies among multiple alternatives by trial and error. We applied the proposed method to the El Farol bar problem by introducing *role* into strategy and a quorum evaluation based on the combination of roles. The experimental results verified that the proposed could solve the El Farol bar problem with a high scalability for the number of strategies M and group size N .

References

- [1] Gabriele Valentini, Eliseo Ferrante and Marco Dorigo, “The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives”, *Frontiers in Robotics and AI*, Vol.4, Article 9, pp.1–18, 2017.
- [2] Parker C.A.C., and Zhang H., “Cooperative decision-making in decentralized multiple-robot systems: the best-of-n problem”, *IEEE/ASME Transactions on Mechatronics*, Vol.14, Issue 2, 2009.
- [3] Wessnitzer J., Melhuish C., “Collective Decision-Making and Behaviour Transitions in Distributed Ad Hoc Wireless Networks of Mobile Robots: Target-Hunting”, In: Banzhaf W., Ziegler J., Christaller T., Dittrich P., Kim J.T. (eds) *Advances in Artificial Life. ECAL 2003. Lecture Notes in Computer Science*, Vol.2801, pp. 893–902, 2003
- [4] Iwanaga Saori, and Akira Namatame, “The complexity of collective decision”, *Nonlinear Dynamics, Psychology, and Life Sciences*, Vol.6, Issue 2, pp.137–158, 2002.
- [5] Nhu Hai Phung, Masao Kubo, Hiroshi Sato, Saori Iwanaga, Akira Namatame, “Agreement algorithm with trial and error method at macro level”, *Swarm2017*, pp.82–85, 2017.
- [6] Masao Kubo, Nuhai Phung, Hiroshi Sato, “Efficient collective search by agents that remember failures”, *Acrob 2018*, 2018.
- [7] Gabriele Valentini, Heiko Hamann and Marco Dorigo, “Self-Organized Collective Decision Making: The Weighted Voter Model”, *AAMAS ’14 Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 45–52, 2014.
- [8] A. Scheidler, A. Brutschy, E. Ferrante, M. Dorigo, “The k-Unanimity Rule for Self-Organized Decision Making in Swarms of Robots”, *IEEE Transactions on Cybernetics*, Vol.46, Issue 4, pp. 1175–1188, 2016.
- [9] Thomas D. Seeley, P. Kirk Visscher, Thomas Schlegel, Patrick M. Hogan, Nigel R. Franks, James A. R. Marshall, “Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms”, *Science*, Vol.335, Issue 6064, pp.108–111, 2012.
- [10] Darren Pais, Patrick M. Hogan, Thomas Schlegel, Nigel R. Franks, Naomi E. Leonard, James A. R. Marshall, “A Mechanism for Value-Sensitive Decision-Making”, *PloS One*, 2013, doi: 10.1371/journal.pone.0073216.
- [11] Andreagiovanni Reina, Gabriele Valentini, Cristian Fernandez-Oto, Marco Dorigo, Vito Trianni, “A Design Pattern for Decentralised Decision-Making”, *PloS One*, 2015, doi: 10.1371/journal.pone.0140950.
- [12] Andreagiovanni Reina, James A. R. Marshall, Vito Trianni and Thomas Bose, “Model of the best-of-N nest-site selection process in honeybees”, *Phys. Rev. E* 95, 2017, doi: 10.1103/PhysRevE.95.052411.
- [13] Schelling, T.S. Norton & Company, “Micromotives and Macrobehavior”, *Norton & Company*, 1978.
- [14] Arthur W.B., “Inductive Reasoning and Bounded Rationality”, *American Economic Review: Papers and Proceeding*, Vol.84, No.2, pp.404–411, 1994.

Novelty Search-based Bat Algorithm: Adjusting Distance among Solutions for Multimodal Optimization

Takuya Iwase [†] Ryo Takano [†] Fumito Uwano [†] Hiroyuki Sato [†] Keiki Takadama [†]

[†] The University of Electro-Communications, Tokyo, Japan

tanu_iwa@cas.lab.uec.ac.jp, takano@cas.lab.uec.ac.jp, uwano@cas.lab.uec.ac.jp, h.sato@uec.ac.jp, keiki@hc.uec.ac.jp

Abstract: This paper proposes the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible in multimodal optimization. In detail, this paper focuses on bat algorithm (BA) which copes with the trade-off between exploration and exploitation in the process of the solution search and extends it by introducing novelty search for keeping the distance among solutions. Through simulations of the comparisons between NSBA and BA in the test-bed multimodal functions, the following implications have been revealed: (1) NSBA finds more number of local optima than BA in both Griewank and Rastrigin Functions; (2) the number of local optima in NSBA increases as the number of populations increases, while that in BA does not change even though the number of populations increases in both functions.

Keywords: swarm intelligence, bat algorithm, novelty search, multimodal function

1. Introduction

Most of metaheuristic algorithms for optimization problems are based on biological evolution as nature-inspired system. For example, Particle Swarm Optimization (PSO) [1] modeled as fish swarm searches solutions by considering both the local best solution of their own fishes and the global best solution among all fishes. Firefly Algorithm (FA) [2] searches solutions by moving to a brighter firefly (*i.e.*, better solution). As other algorithm, Bat Algorithm (BA) [3] searches solutions according to the characteristic of echolocation which promotes bats to start to find food or prey (*i.e.*, solutions) widely and narrows down the target food or prey by changing their loudness and pulse emission rate. In detail, all bats continuously search solutions by selecting the better solution than the current one, and reduces the number of times of selecting the better solution, which can be found by one of the following three searches: (i) a movement toward the target *i.e.*, the bat which finds the best solution; (ii) a local movement around the target as a local search; and (iii) a random movement as a *global* search. However, all of these algorithms (*i.e.*, PSO, FA and BA) are not appropriate for dealing with real-world problems which need to find many local optima in search space. This is because the above algorithms are designed to find one single global optimum.

To overcome this problem, this paper focuses on BA and extends it to propose the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible.

In this research, BA is employed because of the following reasons: (1) BA based on the three search mechanisms from (i) to (iii) as described above, while PSO and FA is mainly based on one search mechanism (which is respectively based on the local and best solutions in PSO and the movement to better solutions in FA); and (2) this difference suggests that one of search mechanisms in BA can be modified for finding multiple local optima with keeping the two search mechanisms as original, while the search mechanism in PSO and FA are hard to be modified for such a purpose due to a lack of the original search mechanism. What should be noted here is that BA still has the functions of exploitation and exploration in the solution search process as a local search by the mechanism (ii) and a global search by the mechanism (iii).

This paper is organized as follows. After this section, the mechanisms of BA and the proposed BA are explained in Sections 2 and 3. Section 4 describes the multimodal functions as the test-bed problem in the experiment. Section 5 shows the results and Section 6 discusses them. Finally, our conclusion is given in Section 7.

2. Bat Algorithm

As described in Section 1, BA is a metaheuristic algorithm based on the bat behavior according to its loudness and pulse emission rate of the reflect wave, which control the balance between a local and global search. When a bat finds the better solution than the current one, the loudness A_i and the pulse

rate r_i gradually decreases and increases, respectively. To find better solution, the bat has the following three solution search mechanisms: (i) the bat i flies to the target (*i.e.*, the bat which finds the best solution) with the velocity controlled by frequency f_i ; (ii) the bat i flies around the target as a local search; and (iii) the bat i flies randomly in search space as a global search. Let us explain these search mechanisms. First, in the search mechanism (i), all bats change their locations x_i with their velocities v_i toward the global best solution. For this calculation, the frequency f_i , velocity v_i , and location x_i , of the bat i are calculated as follows:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_* - x_i^{t-1}) * f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

In detail, the new solution x_i is updated by adding the new the velocity v_i which is derived from the previous velocity v_i^{t-1} , the distance between the global best location and the previous location $x_* - x_i^{t-1}$, and frequency f_i which range is $[f_{min}, f_{max}]$ where $f_{min} = 0$ and $f_{max} = 1$. β is uniform random value from 0 to 1. Next, in the solution search mechanism (ii), the new solution x_{loc} is generated around the global best solution as follows:

$$x_{loc} = x_* + \epsilon A^t, \quad (4)$$

where ϵ is uniform random value between $[0, 1]$. In Eq.(6), A^t is the averaged loudness of all bats. Finally, in the search mechanism (iii), x_{rnd} is generated randomly in search space as follows:

$$x_{rnd} = x_{lb} + (x_{ub} - x_{lb}) * rand(1, D) \quad (5)$$

where x_{ub} and x_{lb} describe the upper and lower bounds of the search space, and $rand(1, D)$ is the D dimensional uniform random value between $[0, 1]$.

When a bat finds the better solution than the current one, the loudness A_i and pulse emission rate r_i are updated as follows:

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_i^0 [1 - exp(-\gamma t)] \quad (7)$$

Note that the loudness A_i^0 is initialized as $A_i^0 = 1$ and the pulse rate is initialized as a uniform random value r^0 between $[0, 1]$ or a number closed around zero. The parameters α and γ are the symbolized damping coefficient. The pseudo code of BA is given in the Algorithm 1 and its brief summary is described below.

- STEP1: Population initialization of bats (line 1 to 3)
The population of bats $x_i(i = 1, 2, \dots, N)$, the loudness A_i^0 , the pulse rate r_i^0 are initialized as the initial values. The frequency f_i is initialized by Eq.(1).

- STEP2: New solution updates (line 6)
The new solutions x_i is calculated by Eqs. (2)(3).
- STEP3: New solution generation around global best solution x_* (line 7 to 9)
A new solution x_{loc} is generated around x_* by Eq. (4) when the pulse emission rate r_i is lower than a random value.
- STEP4: Random new solution generation (line 10)
A new solution x_{rnd} is generated randomly by Eq. (5).
- STEP5: Solutions update(line 11 to 14)
When $rand < A_i$, the best solution is selected from x_i , x_{loc} , and x_{rnd} by Eqs.(6),(7)
- STEP6: Return to STEP2

Algorithm 1 Bat Algorithm

Require: Objective Function $F(x)$

- 1: Initialize Population $x_i(i = 1, 2, \dots, N)$ and v_i
 - 2: Define frequency f_i at location x_i [eq.(1)]
 - 3: Initialize pulse rates r_i , and loudness A_i
 - 4: **while** ($t <$ Max number of iterations) **do**
 - 5: **for** $i=1$ to N **do**
 - 6: Generate a new solution x_i and velocity v_i [eqs.(2) to (3)]
 - 7: **if** ($rand > r_i$) **then**
 - 8: Generate a new solution x_{loc} around a global best solution x_i [eq.(4)]
 - 9: **end if**
 - 10: Generate a new solution x_{rnd} randomly
 - 11: **if** ($rand < A_i \& \min(F(x_i), F(x_{loc}), F(x_{rnd})) < F(x_{i*})$) **then**
 - 12: Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
 - 13: **end if**
 - 14: Evaluate all bats and select a best solution x_* in the current solutions
 - 15: **end for**
 - 16: **end while**
-

3. Proposed Algorithm

3.1 Novelty Search

Novelty search [4] developed in the context of computation aims to search new solutions which have not yet searched. For this purpose, novelty search measures the distance among the candidate solutions to evaluate the dense of them and then generates new solutions into sparse area by considering the distance among candidate solutions. The sparseness of the solutions is calculated as follows:

$$\rho(x) = \frac{1}{k} \sum_{i=0}^k dist(x, \mu_i), \quad (8)$$

where $\rho(x)$ is the sparseness at the location x , k is the number of solutions around x in the k -nearest neighbors, and $dist(x \text{ and } \mu_i)$ is the distance between x and μ_i which is the i -th nearest neighbor of x . The $\rho(x)$ value indicates the averaged distance between x and solutions around x . Fig. 1 shows an example of the case where $k = 3$ (*i.e.*, the number of solutions around the target solution). where the yellow star indicates the target solution and the red circles indicates the other solutions. In this figure, the yellow solution moves to the sparse area where is far from other three red solutions.

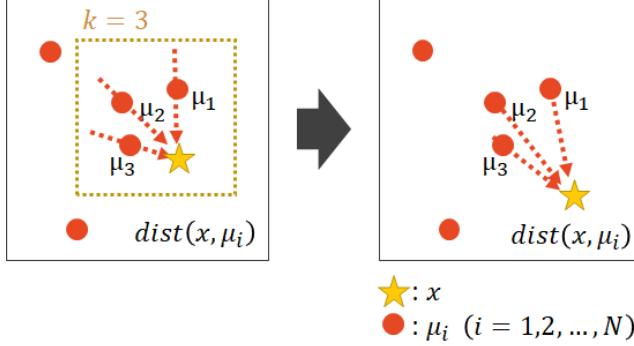


Fig. 1: distributed a solution to sparse area

3.2 Novelty Search-based Bat Algorithm

Novelty Search-based Bat Algorithm (NSBA) is extended from BA by introducing the concept of novelty search, however, we change the value of sparseness equation mentioned above eq.(8) to vector equation. The essential difference between NSBA and BA is related to the search mechanism (i). Concretely, all solutions are updated by adding their velocity calculated by the following Eqs. (9) and (10), which corresponds to Eq. (2) in BA.

$$d_i^{t-1} = \frac{1}{N} \sum_{j=1}^N \frac{(x_{i*} - x_j^{t-1})}{|x_{i*} - x_j^{t-1}|^2} \quad (9)$$

$$v_i^t = v_i^{t-1} + d_i^{t-1} * f_i \quad (10)$$

where N is population size, x_{i*} is the current personal best solution, and x_i^{t-1} is previous solution. All bats with velocity v_i^t and location x_i^t are updated same as (10) and (3) of BA. Used distance function in Novelty search describes scalar equation. However in this proposes, we alter scalar to vector equation for determining search direction. Next, in the search mechanism (ii), x_{loc} is generated around the personal best solution x_{i*} as follows:

$$x_{loc} = x_{i*} + \epsilon A^t, \quad (11)$$

where ϵ is uniform random value between $[0, 1]$.

3.3 Algorithm of NSBA

The pseudo code of NSBA is given in the Algorithm 2, and its brief summary is described below. Except for the step 2, the other steps are the same as BA.

- STEP1: Population initialization of bats (line 1 to 3)
This step is the same as BA.
- STEP2: New solution update (line 6)
The new solutions x_i^t are calculated by Eqs. (10)(3) with (9).
- STEP3: New solution generation around solutions x_i (line 7 to 9)
A new solution x_{loc} is generated around x_{i*} by Eq. (11)
- STEP4: Generate a new solution randomly (line 10)
This step is the same as BA.
- STEP5: Solutions update (line 11 to 15)
This step is the same as BA.
- STEP6: Return to STEP2

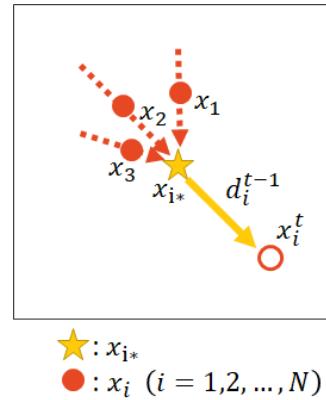


Fig. 2: Bat motion of NSBA

Algorithm 2 Novelty Search-based Bat Algorithm

Require: Objective Function $F(x)$

```

1: Initialize Population  $x_i (i = 1, 2, \dots, N)$  and  $v_i$ 
2: Define frequency  $f_i$  at location  $x_i$  [eq.(1)]
3: Initialize pulse rates  $r_i$ , and loudness  $A_i$ 
4: while ( $t < \text{Max number of iterations}$ ) do
5:   for  $i=1$  to  $N$  do
6:     Generate a new solution  $x_i$  and update velocity  $v_i$ 
      [eqs.(3)(9)(10)]
7:     if ( $rand > r_i$ ) then
8:       Generate a new solution  $x_{loc}$  around the solution
          $x_i$  [eq.(11)]
9:     end if
10:    Generate a new solution  $x_{rnd}$  randomly (or without
         $x_{rnd}$ )
11:    if ( $rand < A_i$  &  $\min(F(x_i), F(x_{new}), F(x_{rnd})) <$ 
         $F(x_{i*})$ ) then
12:      Accept the new solution, and update pulse rate  $r_i$ 
        & loudness  $A_i$  [eqs. (6)(7)]
13:    end if
14:  end for
15:  Evaluate the all bats and select a best solution  $x_{i*}$  in
    the current solutions
16: end while

```

4. Experiment

To validate the effectiveness of NSBA, this paper employs the following well-known multimodal functions which have similar fitness landscape: Griewank function [5] and Rastrigin function [6].

4.1 Benchmark Test Functions

Table. 1 summarizes the features of the benchmark multimodal functions. In detail, the first, second, third, and fourth lines indicate the search space, the fitness value of the global optimum, the number of global and local optima, respectively. Figs. 3(a) & 4(a) show the fitness landscape of both functions: 3(b) & 4(b) show the contour plot of both functions from the two-dimensional viewpoint. These figures indicate a change of the fitness value with the color density, where the horizontal and vertical axes indicate x_1 and x_2 , respectively. As a color becomes darker, the fitness value becomes smaller. This paper employs Griewank and Rastrigin functions because both functions are similar fitness landscape but they are some differences of the number of the local optima, search space and fitness value.

F_1 : Griewank Function

This function is described as follows as shown in Fig. 3(a).

$$F(x) = \sum_{i=1}^D \frac{x_i}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (12)$$

Table. 1: Measurement of Benchmark Test Functions

Function	F_1	F_2
Search Space	$-10 \leq x_i \leq 10$	$-5 \leq x_i \leq 5$
$F(x_*)$	0	0
Num of global optima	1	1
Num of local optima	16	120

where D is the number of dimension and the global optimum is $f(x_*) = 0$, at $x_* = [0, 0]$. In the case of $D = 2$ this function has 17 local optima $f(x_{i*}) \approx 0$ at $\pm x \approx [6.2800, 8.8769], [3.1400, 4.4385], [0, 8.8769], [6.2800, 0], [9.4200, 4.4385]$ in the range between $-10 \leq x_i \leq 10$ with $i = 1, 2$.

F_2 : Rastrigin Function

This function is described as follows as shown in Fig. 4(a).

$$F(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)] \quad (13)$$

where D is the number of dimension and the global optimum is $f(x_*) = 0$ at $x = [0, 0]$. In the case of $D = 2$, this function has 121 local optima in the search space at $\pm x_i = [0, \dots, 11, 0, \dots, 11]$ in the range between $-5 \leq x_i \leq 5$ with $i = 1, 2$.

4.2 Evaluation Criteria

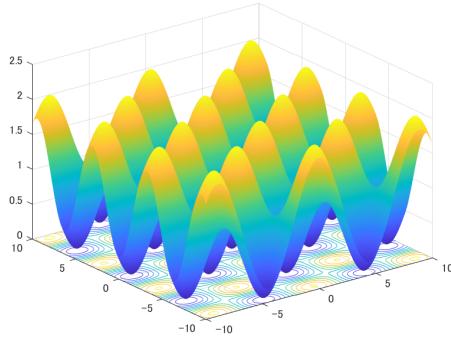
This experiment employs Peak Ratio(PR) [7] as the evaluation criterion in the CEC (IEEE Congress on Evolutionary Computation) 2013 competition [8]. The PR value measures the ratio of the found global and local optima in the total number of true peaks and it is calculated as follows:

$$PR = \frac{\sum_{run=1}^{MR} FPs}{TP * MR} \quad (14)$$

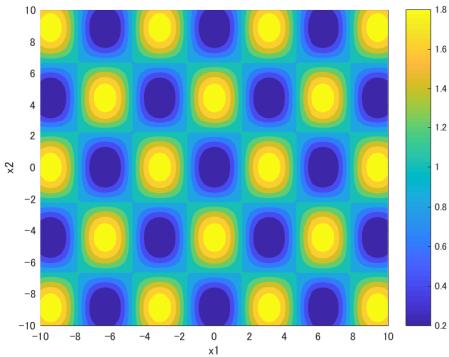
where MR indicates the maximum run, FPs indicates indicates the number of peaks found by the optimization algorithm. TP indicates the true number of peaks of the function. We define that the peak is found when the Euclid distance between the true peak and the solution calculated by the optimization algorithm is less than the threshold $\varepsilon = 0.1$.

4.3 Experimental Parameters

All experiments employs the parameters as follows: frequency $f_{max} = 1$, $f_{min} = 0$, loudness $A^0 = 1$, parse rate $r^0 \in [0, 1]$ with $\alpha = \gamma = 0.9$. The population size $N = 50, 100$ for Griewank function and $N = 100, 150$ for Rastrigin function. In both functions, the optimization algorithm runs 30 time, each of which maximum iteration is 10000.

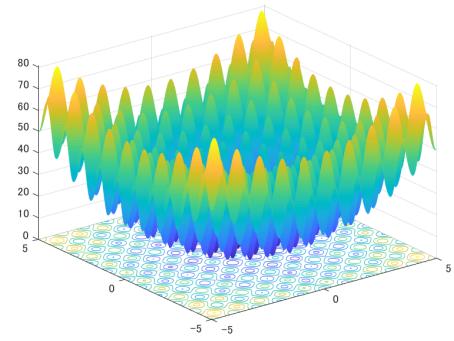


(a) Fitness landscape

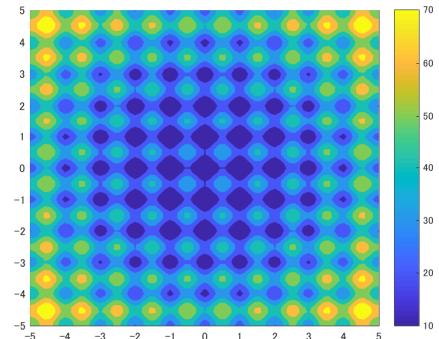


(b) Contour plot

Fig. 3: F_1 : Griewank Function



(a) Fitness landscape



(b) Contour plot

Fig. 4: F_2 : Rastrigin Function

5. Result and Discussion

To evaluate NSBA, this paper compares the performance between NSBA and BA in terms of (a) the number and ratio of the found peaks and (b) the convergence speed.

5.1 Number and Ratio of Found Peaks

From the viewpoint of the (a) criteria, Table. 2 compares both algorithms in terms of the Mean and SD of the number of the found peaks and the PR value of averaged over 30 runs. From this table, NSBA performs better than BA because both the number of the found peaks and the PR value of NSBA is larger than those of BA in both functions with the different population size. Figs. 5 and 6 show the location of the solutions (marked with the white circle) of both algorithms at the final iteration. BA converged to the single global optimum on any run, while NSBA distributes the solutions to not only the single global optimum but also the other local optima.

5.2 Convergence Speed

From the viewpoint of the (b) criteria, Fig. 7 measures the convergence speed of the peak ratio (PR) of NSBA and BA. In this figure, the vertical axis indicates the peak ratio while horizontal axis indicates the iteration. The lines with the

black and white circle indicate the PR value of NSBA and BA, respectively. As shown in Fig. from 5(a) to 5(d), the PR value in BA sharply decreases to almost 0 % until 1000 iterations and keeps the same value in both functions. In comparison with the BA, the PR value of NSBA in F_1 increases around 70 % until 1000 iterations and then gradually decreases to around 30-50 % after 1000 iterations. However, the PR value of NSBA in F_2 does increase but gradually decreases from 20 % to 10 % after 1000 iterations. This results suggests that the search mechanism (ii) works to spread solutions away toward sparse area, however, NSBA has strong convergence to the global best solution and the personal best solutions around it.

5.3 Analysis of Results

5.3.1 Population Size

In order to investigate the influence of the population size, this subsection analyzes the results of BA and NSBA in the different population size. From Table 2, the number of the found peaks and the PR value of BA do not change by the different population size in both functions, while those of NSBA increase as the population size increases. Concretely, the number of the found peaks and the PR value of NSBA increase from 6.8 and 40 % (with 50 population size) to 7.267

Table. 2: Found Peaks and Peak Ratio of BA and NSBA

Function	BA			NSBA		
	Mean	SD	PR	Mean	SD	PR
$F_1 (N = 50)$	1.0	0	5.89 %	6.8	0.7024	40.0 %
$F_1 (N = 100)$	1.0	0	5.89 %	7.267	0.5735	42.75 %
$F_2 (N = 100)$	1.0	0	0.87 %	7.9333	0.8929	6.56 %
$F_2 (N = 150)$	1.0	0	0.87 %	8.0667	0.7717	6.67 %

and 42.75 % (with 100 population size) in F_1 and from 7.9333 and 6.56 % (with 100 population size) to 8.0667 and 6.67 % (with 150 population size) in F_2 . This results suggests that the large size of the population contributes to increasing the performance of NSBA while does not contributes to BA because the distributed solutions by NSBA can cover many local optima as the population size increases, while the converged solution by BA mainly cover the only global optimum even though the population size increases. Fig. 7 also supports the above good/bad influence of the population size in NSBA, *i.e.*, the PR value of NSBA in F_1 decreases from 70 % to 30 % in the case of the small (50) population size, while that in F_1 keeps around 50 % after 6000 iterations in the case of the large (100) population size.

5.4 Distribution of Solutions

In order to investigate the reason why the PR values of NSBA in both function decreases in Fig. 7 (*i.e.* why NSBA cannot keep the maximum PR value), this subsection analyzes how the solutions are distributed in Fig. 8 which shows the location of the solutions in 2D dimension at the 1000 iterations. From Fig. 8, NSBA can cover almost all peaks at the 1000 iterations on F_1 but the number of the found peaks decreases after the 1000 iterations. This because NSBA has the strong convergence to the best solution in the evolutionary process, even though NSBA has a potential of finding the global optimum and many local optima. The same tendency of the decrease of the found peaks can be found in F_2 even though the shape of the PR value is different.

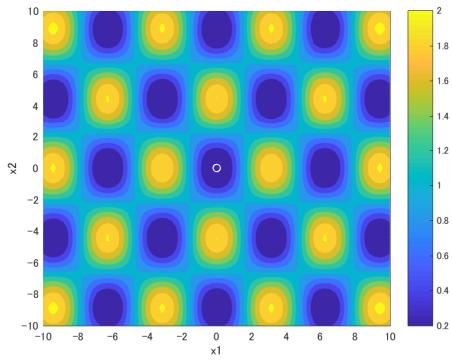
6. Conclusions

This paper focused on bat algorithm (BA) and extended it to propose the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible in multimodal optimization. Through the comparisons between NSBA and BA in the test-bed multimodal functions, this paper validated the effectiveness of NSBA. Concretely, the following implications have been revealed: (1) although NSBA and BA succeeded to find a global optimum, NSBA found more

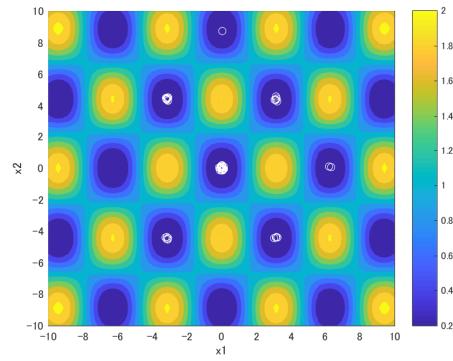
number of local optima than BA in both Griewank and Rastrigin functions; and (2) the number of local optima in NSBA increased as the number of populations increased, while that in BA did not change even though the number of populations increased in both functions. Our future prospects are summarized as follows: (1) applying NSBA into other benchmark functions, and (2) increasing the performance to find the large number of local optima.

References

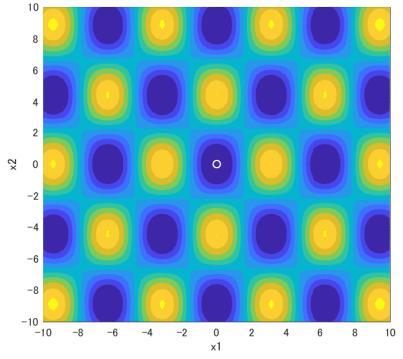
- [1] R. Eberhart, J. Kennedy: A New Optimizer Using Particle Swarm Theory, *Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan)*, IEEE Service center, Pis-cataway, NJ, No. 1, pp. 39–43, 1995.
- [2] X. S. Yang: Firefly Algorithms for Multimodal Optimization, *in: Stochastic Algorithms: Foundations and Applications*, SAGA, Vol. 5792, pp. 169–178, 2009.
- [3] X. S. Yang: A Metaheuristic Bat-Inspired Algorithm, *in: Nature Inspired Cooperative Strategies for Optimization(NICSO 2010)*, Springer, Berlin, Vol. 284, pp. 65–74, 2010.
- [4] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pp. 329–336, 2008.
- [5] M. Molga, and C. Smutnicki, Test functions for optimization needs (2005). Retrieved June 2013, from <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>.
- [6] H. Pohlheim, GEATbx Examples: Examples of Objective Functions (2005). Retrieved June 2013, from http://www.geatbx.com/download/GEATbx_ObjFunExp1_v37.pdf.
- [7] R. Thomsen,"Multimodal Optimization using Crowding-based Differential Evolution," *In the IEEE Congress on Evolutionary Computation*,2004. CEC2004, vol.2, pp. 1382-1389, 19-23 June, 2004.
- [8] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization", *Evol. Comput. Mach. Learn. Group*, RMIT University, Melbourne, VIC, Australia, Tech. Rep., 2013.



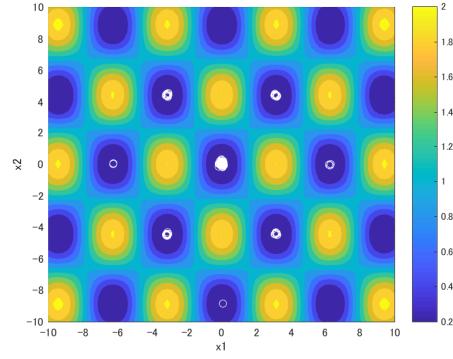
(a) $F_1 : (N = 50)$



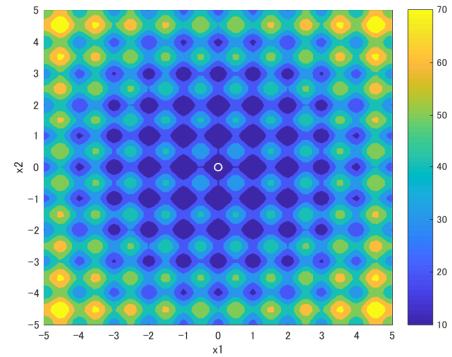
(a) $F_1 : (N = 50)$



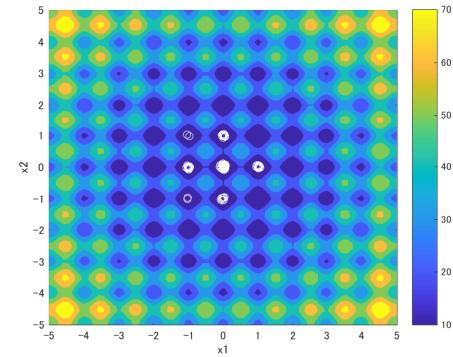
(b) $F_1 : (N = 100)$



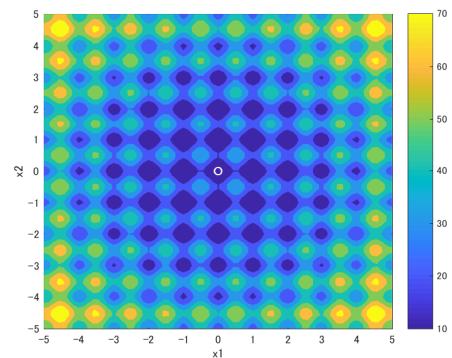
(b) $F_1 : (N = 100)$



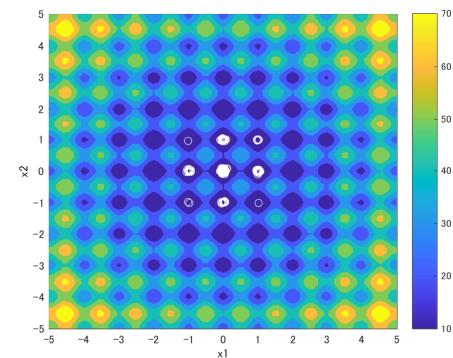
(c) $F_2 : (N = 100)$



(c) $F_2 : (N = 100)$



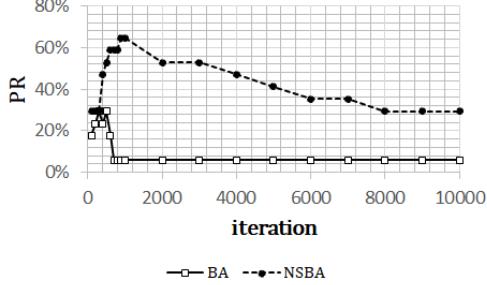
(d) $F_2 : (N = 150)$



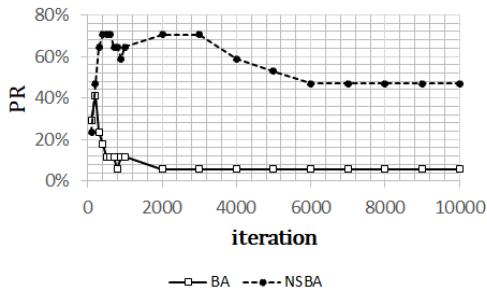
(d) $F_2 : (N = 150)$

Fig. 5: BA

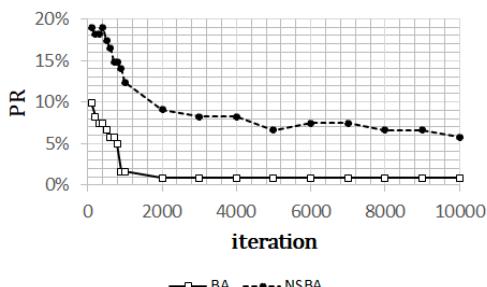
Fig. 6: NSBA



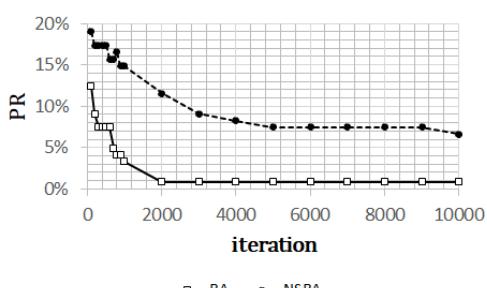
(a) $F_1 : (N = 50)$



(b) $F_1 : (N = 100)$



(c) $F_2 : (N = 100)$



(d) $F_2 : (N = 150)$

Fig. 7: Convergence Speed of Peak Ratio implemented by BA and NSBA

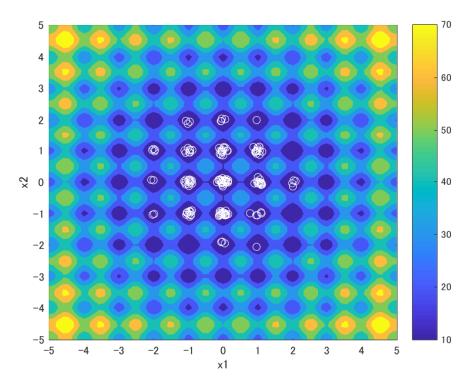
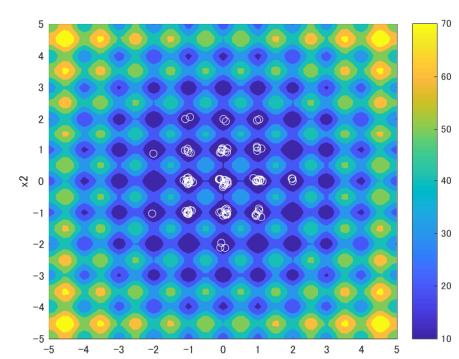
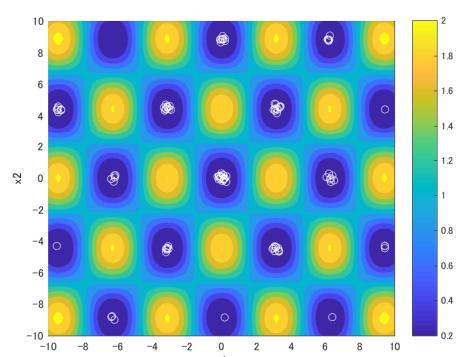
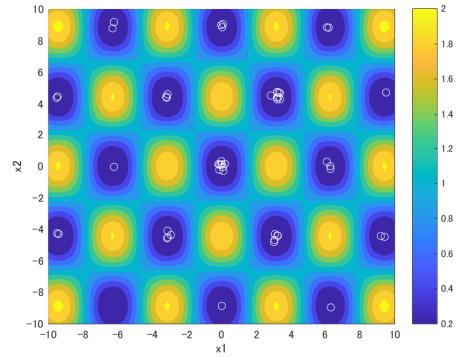


Fig. 8: Distribution of population of NSBA at 1000 iteration

Analysis of Epidemic of Seasonal Influenza in Closed Space

Saori Iwanaga [†] Keigo Kawaguchi ^{††}

^{†, ††} Japan Coast Guard Academy, Kure, Japan

[†] s-iwanaga@jcg.a.ac.jp, ^{††} g14012e@jcg.a.ac.jp

Abstract: We analyzed the epidemic of seasonal influenza occurred in our academy during January, 2017. Using our epidemic data, we chased the channel of infection in our academy as closed space and we estimated 79 % of channel of infection on weekdays. Seasonal Influenza has incubation period, which is composed of exposed period and infectious period but have no symptom. We introduce a refinement to the former epidemic model to take into incubation period and propose discrete-time SEPIR model for seasonal influenza and estimated the transition parameters. We acknowledged that infectious period but have no symptom is key for the epidemic of seasonal influenza. By SEPIR model and the estimated parameters, it is expected to examine what kind of measures is effective by simulating epidemic of seasonal influenza.

Keywords: Epidemic model, seasonal influenza, data analysis, closed space.

1. Introduction

During January 2017, an epidemic of influenza A, that is seasonal influenza [1], occurred in our academy. After three consecutive holidays from January 7 to January 9, students returned dormitory and started to take classes. On Friday January 13, two students were in the academy infirmary. The outbreak spread rapidly and the maximum number of patient was 27 on January 16 and 18. The epidemic reached a peak in one week and was over in two weeks. At last, 37 (25 %) students, especially 33 (32 %) of freshmen and sophomores were affected. Of about 60 teachers who had some contacts with the students didn't develop similar symptoms.

There are some training courses and about 300 boys and girls students in our academy. Among them, about 200 students are in undergraduate course and some of them were absent because of the ship training. A total of 150 undergraduate students were at that time. According to our medical doctor, the outbreak was different from usual year. He also said that about five students develop influenza at the same dormitory floor in an epidemic wave of influenza season. Before six academic years, about 10.1 students per year develop influenza A. But in the year, there are two waves of epidemic and totally 43 students had influenza A.

In this study, using our epidemic data, we chased the channel of infection in our academy as closed space and we estimated channel of infection on weekdays. Seasonal Influenza has incubation period, which is composed of exposed period and infectious period but have no symptom. We introduce a refinement to the former epidemic model to

take into incubation period and propose discrete-time SEPIR model for seasonal influenza and estimated the transition parameters. We acknowledged that infectious period but have no symptom is key for the epidemic of seasonal influenza. By SEPIR model and the estimated parameters, it is expected to examine what kind of measures is effective by simulating epidemic of seasonal influenza, which is useful to manage group life.

2. Related Works

(1) SIR Model

In SIR model [2], simple population is considered which means no birth, death, or migration. The state transition diagram of an individual is shown in Fig. 1(a). "S" means susceptible state, "I" means infectious state and "R" means recovered state. An individual with susceptible state "S" can transit to infectious state "I" by contact with infectious individual "I". There are some contacts between susceptible and infectious individual and the probability of the happening is determined by the respective numbers of "S" and "I". Take into account a mean infectivity rate β (per day), the susceptible individual transit to infectious as given in Eq (1). An individual move from "I" to "R" depend on how long it have been in "I" state. By introducing the recovery rate γ (per day) which is the inverse of the infectious period, this leads to far more straightforward equation as giving by Eq (3). In these equations, I is the total number of individuals who are infectious. S is the number of susceptible individuals. R is number of recovered individuals who have successfully cleared the infection.

Copyright © IES 2018

$$\frac{ds}{dt} = -\beta SI \quad (1)$$

$$\frac{di}{dt} = \beta SI - \gamma I \quad (2)$$

$$\frac{dr}{dt} = \gamma I \quad (3)$$

(2) SEIR Model

Keeling et.al [4] deal with incubation period and propose mathematical SEIR model which many childhood infectious diseases (such as measles, rubella, or chickenpox) follow. They introduce a refinement to the SIR model to take into account incubation period. The state transition diagram of an individual is shown in Fig. 1(b). They added "E", that is exposed state, to SIR model. Exposed individual "E" is infected but not yet infectious. Only infectious individual "I" infect susceptible individuals "S". Take into account transition rate σ (per day), exposed individual transit to infectious as given in Eq (6). There, exposed period is given by $1/\sigma$. E is the number of exposed to infectious individuals.

$$\frac{ds}{dt} = -\beta SI \quad (4)$$

$$\frac{de}{dt} = \beta SI - \sigma E \quad (5)$$

$$\frac{di}{dt} = \sigma E - \gamma I \quad (6)$$

$$\frac{dr}{dt} = \gamma I \quad (7)$$

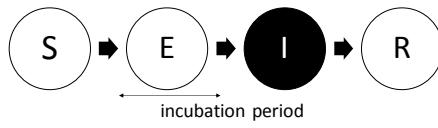
There is a report of influenza in boarding school of England in 1978 [3]. At first, there only three students were influenza. A total of 763 boys between the ages of 10 and 18 were at risk. Among the boys, there were 113 younger boys between the age of 10 and 13. Then, 512 (67 %) of 763 boys developed influenza and the epidemic reached a peak in seven days and was over in 13 days. Especially 83 % of younger boys were affected. Keeling et.al [4] used SIR model and fitted to the data. Estimated parameters are $\beta = 1.66$ per day and $1/\gamma = 2.2$ days using a simple squares method. The basic reproductive ratio R_0 , which is determined by $\beta/\gamma = 3.65$ is within the range of human's influenza, between 3 and 4 [5].

We also adopt SIR model and fitted to our academy's data. Our academy is also boarding school. Estimated parameters are $\beta = 0.801$ per day and $1/\gamma = 3.88$ days using a simple squares method, given an R_0 of 3.14. Our case is also within the range of human influenza. Comparing with two cases, infectivity rate in our academy is lower than that in England. That is why the ratio of influenza patient in our academy is smaller than that in England. Because the patient who show symptoms is isolated in a sick room, susceptible students "S" cannot contact with infectious

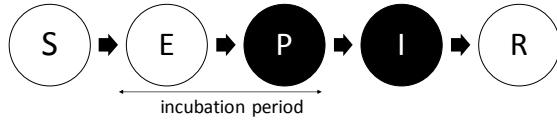
student "I" in our academy. Our case doesn't match SIR model.



(a) SIR model



(b) SEIR model



(c) SEPIR model

Fig. 1. The state change diagram of epidemic model: circle shows state of individual. S: Susceptible state, I: Infected state, R: Recovered state, E: Exposed state and P: Previously infectious state. Black circle means that it is possible to infect others.

3. Our Model

CDC [6] shows that most healthy adults can infect other people beginning one day before symptoms develop and up to five to seven days after becoming sick. Symptoms start one to four days after the virus enters the body. That means that you may be able to pass on the influenza to someone else before you know you are sick, as well as while you are sick. Some people can be infected with the influenza virus but have no symptoms. During this time, that is incubation period, those persons may still spread the virus to others. SEIR model doesn't have the period and doesn't match influenza. Many diseases have profiles that are individualistic and require specific model formulation [4].

We introduce a refinement to the SEIR model to take into account incubation period and propose discrete-time SEPIR model for seasonal influenza. Incubation [4] period is divided into two period, exposed period and infectious period but have no symptom. The state transition diagram of an individual is shown in Fig. 1(c). We added "P" status as the latter to SEIR model, which is previously infectious. Exposed individual "E" is infected but not yet infectious. Infectious individual "I" and previously infectious individual "P" can infect susceptible individual "S". Take into account a mean infectivity rate α (per day), the susceptible individual transit to infectious as given in Eq (8). Transition rate σ (per day) and recovery rate γ (per day) are also the inverse of the exposed period and the infectious period. Take into account transition rate τ (per day), previously infectious transit to infectious as given in Eq (9). Similar to former models, we set transition rate τ (per day)

is the inverse of the previously infected period. $P(t)$ is the number of previously infectious individuals and t represent days. $S(t)$ is the number of susceptible individuals. $E(t)$ is the number of exposed to infectious individuals. $I(t)$ is the number of infectious individuals. $R(t)$ is number of recovered individuals who have successfully cleared the infection.

$$\Delta S = S(t+1) - S(t) = -\alpha S(t)P(t) - \beta S(t)I(t) \quad (8)$$

$$\Delta E = E(t+1) - E(t) = \alpha S(t)P(t) + \beta S(t)I(t) - \sigma E(t) \quad (9)$$

$$\Delta P = P(t+1) - P(t) = \sigma E(t) - \tau P(t) \quad (10)$$

$$\Delta I = I(t+1) - I(t) = \tau P(t) - \gamma I(t) \quad (11)$$

$$\Delta R = R(t+1) - R(t) = \gamma I(t) \quad (12)$$

4. Estimation Parameters

(1) Our Epidemic Data

We have infected students' data in TABLE. 1. There are 37 infected students between January 9 and January 26. Base time is 08:30 on January 9. First columns ID is student identification number. Next 18 columns show the daily state of the students. "I" means infectious and is isolated in sick room. We also show the number of isolated student on the day at last row ND. Last columns ND is the isolated days of the student. The average of infected period is 3.86 days.

TABLE. 1 EPIDEMIC OF INFLUENZA A IN JANUARY, 2017.

S: Susceptible state, E: Exposed state, P: Previously infectious state. I: Infected state and R: Recovered state. Date with under bar means holiday.

ID	Date																		ND
	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1						I	I	I	I										4
2						I	I	I	I										4
3						I	I	I	I	I									5
4						I	I	I	I	I									5
5						I	I	I	I	I									5
6						I	I	I	I	I									4
7						I	I	I											3
8						I	I	I	I	I									5
9						I	I	I	I	I									4
10						I	I	I	I	I									4
11						I	I	I	I	I									4
12						I	I	I	I	I									4
13						I	I	I	I	I									4
14						I	I	I	I	I									4
15						I	I	I	I	I									4
16						I	I	I	I	I									4
17						I	I	I	I	I									4
18						I	I	I	I	I									4
19						I	I	I	I	I									4
20						I	I	I											3
21						I	I	I											3

22								I	I	I	I								4
23								I	I	I	I								4
24								I	I	I	I								4
25								I	I	I									3
26								I	I	I									3
27								I	I										2
28								I	I	I	I								4
29									I	I	I	I							4
30									I	I	I	I							4
31									I	I	I	I							4
32									I	I	I	I							4
33									I	I	I								3
34									I	I	I								3
35										I	I	I	I						4
36										I	I	I							3
37											I	I	I	I	I				5
ND	0	0	0	0	0	2	7	21	27	25	27	12	9	7	3	1	1	1	0

(2) Period of the States

To calculate transition rate, student's health condition of the day is important. For SEPIR, we must classify the states of students but infectious in TABLE. 1. We can easily set the state "R" on blank cell at right hand side of "I". We set the state "S", "E" and "P" on blank cell at left hand side of "I". When students change from "E" to "P" or from "P" to "I"? That is, how long does it take to change the state?

We can calculate back infected students in all rooms because the students are not so many and managed in our academy. Students live in along the daily schedule as shown in TABLE. 2. Every day they have eight hours' sleeping time in the bedroom and three hours and fifteen minutes' self-study time in study room. On weekday, they have up to four classes of 90 minutes in the classroom and exercise in sports activities. The classes are compulsory. We focus on closed space such as bedroom and classroom, which are at risk of infection. Of course, students in our academy can go out, off business hours. According to students, it is too cold to go out for students on weekday in mid-winter. Therefore, we can regard our academy on weekday as closed space at that time. It is easily analyze the epidemic of influenza derived by two patients. On Friday January 13, only two students (student 1 and 2) developed influenza A. After that, influenza spread rapidly and 35 students developed in two weeks. We analyze the health condition of students' and channel of infection.

TABLE. 2 SCHEDULE OF STUDENTS ON WEEKDAYS

Time	Activity
00:00-06:30	sleep
07:10-07:40	breakfast
08:45-10:15	class: first period
10:30-12:00	class: second period
12:00-13:00	lunch
13:00-14:30	class: third period
14:45-16:15	class: fourth period
16:15-17:15	sports activities
17:15-18:15	dinner
19:00-22:15	study
22:30-24:00	sleep

Fig. 2(a) is the calculated number of exposed students in the dormitory's study rooms. X-axis is length of incubation period, that is days of "E" and "P". Y-axis is length of previously infectious period, that is days of "P". We set incubation period between one and four days [6]. For example, when incubation period is three days and previously infectious period is two days, three students are exposed in the study rooms. This is calculated as follows. As shown in TABLE. 3, student 14 studies in study room 304, he becomes "E" on January 12, "P" on January 13 and "I" on January 15. Another student 33 studies in the same room and he is "S" on January 14. On January 14, student 14 can infect others in the same study room and student 33 is exposed and becomes "E" next day. Another student 81 is exposed but not become "E". In other study rooms, two students are infected. Therefore, the total is three. In Fig. 2(a), we found that the longer the previously infected period is, the more the number of exposed students becomes. Because of the long previously infectious period, the possibility of infecting susceptible students increases.

Similarly, Fig. 2(b) is calculated number of exposed students in the dormitory's bedrooms. We also found that the longer the previously infected period is, the more the number of exposed students. But, the shorter the incubation period, that is, the shorter exposed period is, the more the number of exposed students becomes. Because the exposed period is short, susceptible time of the students becomes long. And the possibility of infecting susceptible students increases. The properties is also shown in the study rooms. By reviewing the data, we found that all exposed students in the study room are also exposed in the bedroom. Because a bedroom consists of some study rooms in our dormitory. We can put together bedroom and study room and deal with bedroom as representing for dormitory.

Fig. 2(c) is calculated number of exposed students in classrooms. Each student goes the classroom to attend class and according to the class schedule on weekday. We also found that the longer the previously infected period is, the more the number of exposed students becomes. We also found that the shorter the incubation period, that is, the shorter exposed period is, the more the number of exposed students. The number of exposed students in the classroom is more than that in the bedrooms.

We show the breakdown of exposed students on weekdays and on weekends in Fig. 3. We classify students by the channel of infection for incubation period and previously infectious period. X-axis is combination of length of incubation period and length of previously infectious period. A denominator means length of incubation period, that is days of "E" and "P". A numerator means and length of previously infectious period, that is days of "P". For each combination, there are breakdown of exposed students for some classes on weekdays and weekends.

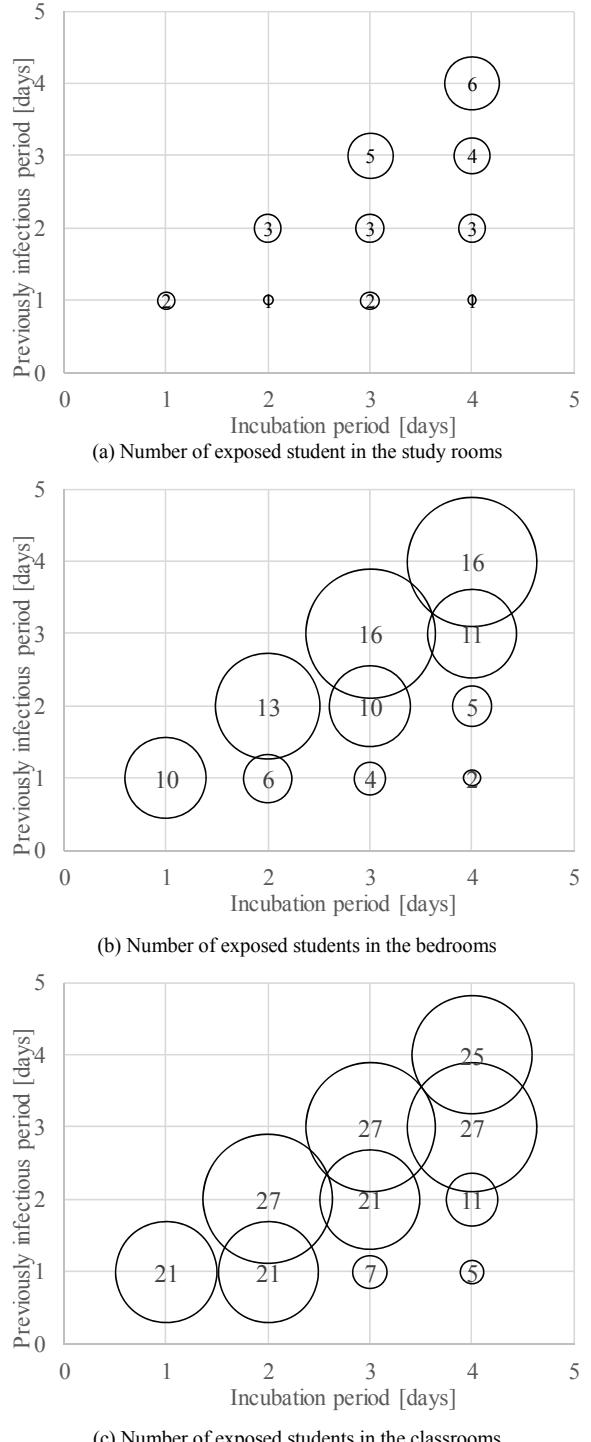


Fig. 2. Exposed and previously infectious students

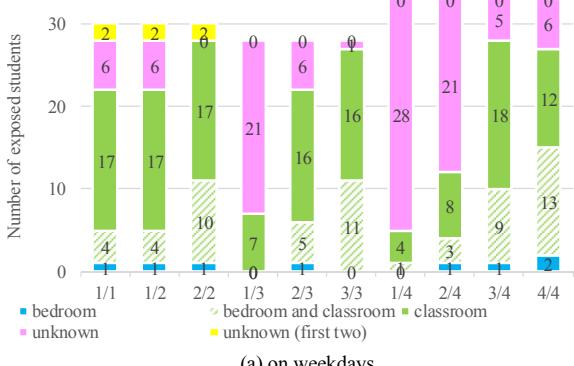
For example, when incubation period is three days and previously infectious period is two days (2/3), 10 students exposed in the bedrooms as shown in Fig. 2(b). And 21 students exposed in the classrooms as shown in Fig. 2(c). There, five students are exposed in both rooms as shown by

Copyright © IES 2018

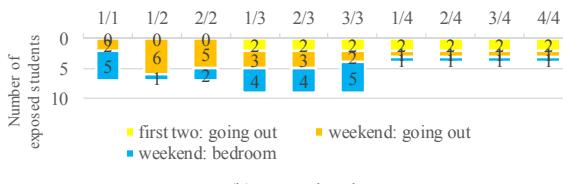
slashed green line bar and 16 students exposed in only classrooms as shown by green bar in Fig. 3(a). Among 10 students exposed in bed rooms, other five students exposed in bedrooms and only one student was infected on weekdays as shown as shown by blue bar in Fig. 3(a) and four students were infected on weekend as shown as shown by blue bar in Fig. 3(b). Assuming that the exposed students in both bedroom and classroom is classified into classrooms as shown by slashed green line bar, we can conclude that five students exposed in bedrooms. And we can regard that first infected two students is infected by going out on weekend between January 7 and 9 by yellow bar in Fig. 3(b). On other weekend between January 14 and 15, three students are infected by going out by orange bar. As for the rests six students, we cannot calculate the channel of infection as shown as unknown by pink bar in Fig. 3(a).

TABLE. 3 EXAMPLE: ONE DAY'S EXPOSED PERIOD AND TOW DAYS' PREVIOUSLY INFECTIOUS PERIOD IN STUDY ROOM 304

ID	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	ND
14	S	S	S	E	P	P	I	I	I	R	R	R	R	R	R	R	R	4	
33	S	S	S	S	S	S	E	P	P	I	I	I	R	R	R	R	R	3	
81	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	0	



(a) on weekdays



(b) on weekends

Fig. 3. The breakdown of exposed students

All things considered, among 10 combinations, the combination of three days' incubation period and three days' previously infectious period (3/3) is the best. Tough, we investigate only 72 % of time on weekdays, such as class, study and sleep, we can estimate 96 % of channel of infection on weekdays in our academy. It is possible to be overestimates the number of exposed students for the shorter incubation period as for combination 3/3. Because we found that the longer the previously infected period or

the shorter the incubation period is, the more the number of exposed students becomes in Fig. 2. As for combinations

2/2, we cannot analyze the channel of infection for first infected two students as shown yellow bar in Fig. 3(a). That is why the combination of three days' incubation period and two days' previously infectious period (2/3) is reasonable. It corresponds to our medical doctor's comment "about five students develop influenza at the same dormitory floor in an epidemic wave of influenza season". By the combination 2/3, we can estimate 79 % of channel of infection on weekdays in our academy.

In our academy's case, we found that three days' incubation period and two days' previously infectious period (2/3). As for incubation period, a suspicious student becomes infected on the fourth days, which means symptoms start four days after the virus enters the body. And students are able to infect other people beginning two day before symptoms, that is, our students are more infectious than people. Diseased students and calculated number of students is shown in Fig. 4.

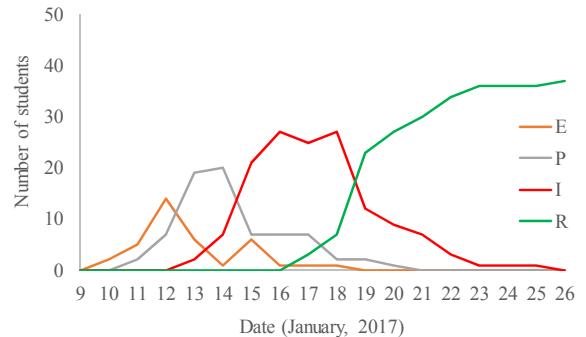


Fig. 4. Diseased students and calculated number of students

(3) The Transmission Rates

Transition parameters are calculated by three days' incubation period and two days' previously infectious period. Three days' incubation period corresponds to $1/\sigma$ and σ is 0.333. Two days' previously infectious period corresponds to $1/\tau$ and τ is 0.5. The average infectious period is 3.86 days, which correspond to the inverse of the infectious period $1/\gamma$. Therefore, recovery rate γ is 0.260.

As for the infectivity rate β , because infectious individual is isolated in sick room, we can set $\beta = 0$ in our academy. As for the infectivity rate α , we must focus on all students' state "P". For example, in bedroom 22, there are nine students. On January 12, there is one student with "P" and there are seven students with "S" and one student with "E". Because the number of students with "S" is same next day, that is, $\Delta S=0$. Independent of values $P(t)$ and $S(t)$, α is 0. To calculate for all state "P" of students, the average α is 0.0236 in bedrooms. As for classrooms, we calculate them and the average α is 0.0206. TABLE. 4 shows the average infectivity rate α of room's type. Details of rooms are shown in TABLE. 5. SR means the study room and identification

number ID and the number of students N are shown. BR means the bedroom and identification number ID and the number of students N are shown. CR means classroom and identification number ID and the number N of students are shown. ID is not real ID but general ID. For ID of study room or bedroom, first figure means the floor.

5. Discussion

The average infectivity rate α of the bedroom and the classroom are almost same. Here, we suppose that there is one student with "P" and the rests are with "S" in each type of room on average. That is, $S(t) = \text{Average}(N)-1$, which is shown in TABLE. 5 and $P(t)=1$. According to eq(8), the average ΔS is calculated. As for bedroom, the average ΔS is -0.150. As for classroom, the average ΔS is -0.537 and the less than bedroom. We found that large contagion is unlikely in dormitory's room as our medical doctor said. Especially, ΔS in classroom 555 ($N=150$) is calculated as -3.07 and is the least in all rooms outstandingly. There is class 62 which all students attend in classroom 555 on January 12, 2017 and epidemic of influenza occur after the class. There is another class 195 which all students attend in classroom 555 on January 23. But, epidemic of influenza didn't occur at all after the class. In TABLE. 1, we found there was one students with "I" and no students with "P" on January 23. But, on January 12, there were there was no students with "I" and seven students with "P". "P" is significant for the epidemic of influenza.

Of course, it is significant to prevent virus invasion to academy. But, it is difficult to observe the invasion from outside. So, it is recommended that the class which all students attend is avoided for incubation period after students returned dormitory from a long holiday.

By the way, on SEPIR model, only students with "P" infects suspicious students in our academy. But, originally, eq(8) and (9), "P" or "I" infects suspicious individual "S". If "I" individual infects "S" individual, the diseased students would be more than that of our case. Comparing the case in England [3], it is assume that diseased students stay in their bedrooms at that time in the England case.

TABLE. 4 THE AVERAGE INFECTIVITY RATE OF ROOM'S TYPE

	Bedroom	Classroom
Average α	0.0259	0.0206
Standard deviation of α	0.0660	0.0637

TABLE. 5 IDENTIFICATION NUMBER AND NUMBER OF STUDENTS OF STUDY ROOM, BEDROOM AND CLASSROOM

SR		BR		CR	
ID	N	ID	N	ID	N
201	3	310	4	11	4
202	3	311	3	12	3
203	3	312	2	13	3
204	2	313	3	14	3
205	3	314	3	15	3
206	4	315	2	16	1

207	3	316	3	17	2	1PB	15
208	3	317	2	22	9	1LA	13
209	2	401	3	23	10	1LB	14
210	4	402	3	24	8	1LC	14
211	3	403	3	25	8	1LD	15
212	3	404	3	26	9	12R	12
213	3	405	3	31	9	12C	33
214	2	406	4	32	9	12K	11
215	3	407	3	33	9	1JU	28
216	4	408	3	34	7	1KE	28
301	3	409	3	35	8	200	47
302	4	410	2	41	9	2EA	28
303	4	411	3	42	10	2EB	19
304	3	412	3	43	8	210	26
305	3	413	3	44	9	220	17
306	3	414	3	45	9	230	4
307	3	415	3	Avg	6.8	211	21
308	3	416	3			222	30
309	3	417	3			233	43
						300	44
						3EA	22
						3EB	22
						3PO	18
						3SA	26
						310	26
						320	14
						330	4
						430	3
						555	150
						Avg	27.0

6. Conclusion

We analyzed the epidemic of seasonal influenza occurred in our academy during January, 2017. Using our epidemic data, we chased the channel of infection in our academy as closed space and we estimated 79 % of channel of infection on weekdays. Seasonal Influenza has incubation period, which is composed of exposed period and infectious period but have no symptom. We introduce a refinement to the former epidemic model to take into incubation period and propose discrete-time SEPIR model for seasonal influenza and estimated the transition parameters. We acknowledged that infectious period but have no symptom is key for the epidemic of seasonal influenza. In our academy, incubation period is three days and student infects others beginning two day before symptoms develop. Because of large number of students in the room, students are infected in classroom more than in bedroom. We also found it is recommended that the event which many people attend is avoided incubation period after people entered or returned in the system for epidemic season. Next, by SEPIR model and the estimated parameters, it is expected to examine what kind of measures is effective by simulating epidemic of seasonal influenza.

Acknowledgment

I wish to thank Doctor Masaki Mukai for advice on analyzing data. He was a medical director of our academy at that time.

References

- [1] CDC flu, <https://www.cdc.gov/flu/keyfacts.htm>
- [2] Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. R. Soc. Lond. A 115, 700-721 (1927).
- [3] "Influenza in a boarding school", British Medical Journal, 4 March 1978.
- [4] Matt J. Keeling and Pejman Rohani, "Modeling Infectious Diseases in Humans and Animals", Princeton University Press, 2008.
- [5] Murray. J.D, Mathematical biology, Springer-Verlag, 1989.
- [6] CDC spread, <https://www.cdc.gov/flu/about/disease/spread.htm>

Appendix

TABLE. 6 PROPERTIES OF STUDENTS (ID: IDENTIFICATION NUMBER, SR: ROOM NUMBER OF STUDY ROOM, BR: BEDROOM AND CR: CLASSROOMS OF ALL STUDENTS)

ID	SR	BR	CR								
			A	B	C	D	E	F	G	H	I
1	409	43	100		12R		1KE	1LB	1MA	1PA	
2	411	44	200	220	2EA			211	233		
3	213	26	200	210	2EA			222	233		
4	303	15	100		1EA	12C		1KE	1LA	1MA	1PA
5	405	14	200	210	2EB			222	233		
6	203	22	200	210	2EB			222	233		
7	307	33	100		1EA	12K		1KE	1LC	1MA	1PA
8	406	42	100		1EA	12C		1KE	1LA	1MA	1PA
9	202	22	100			12C		1KE	1LA	1MB	1PB
10	206	23	200	230	2EA			222	211		
11	208	24	100		1EB	12R		1JU	1LA	1MA	1PB
12	215	26	200	210	2EB			222	233		
13	302	31	100		1EB	12C		1KE	1LD	1MB	1PB
14	304	32	100		1EA	12K		1KE	1LB	1MA	1PA
15	306	32	100		1EA	12K		1KE	1LA	1MB	1PB
16	404	42	100		1EB	12K		1JU	1LD	1MA	1PA
17	404	42	100		1EB	12C		1JU	1LD	1MB	1PA
18	412	44	100		1EB	12R		1KE	1LD	1MB	1PA
19	413	44	100		1EB	12R		1KE	1LC	1MA	1PA
20	308	33	100		1EA	12C		1KE	1LC	1MB	1PB
21	402	41	200	210	2EA			222	233		
22	308	33	200	210	2EB			222	233		
23	309	33	300	310	3EB		3PO				
24	317	35	200	230	2EA			222	211		
25	215	26	100		1EB	12C		1KE	1LC	1MA	1PA
26	407	42	100		1EA	12C		1JU	1LB	1MA	1PA
27	205	23	100		1EA	12C		1JU	1LA	1MA	1PA
28	208	24	300	310	3EB		3SA				
29	203	22	100		1EA	12C		1JU	1LB	1MA	1PA
30	209	24	200	210	2EB			222	233		
31	212	25	200	210	2EB			222	233		
32	413	44	300	310	3EB		3SA				
33	304	31	200	210	2EA			222	233		
34	408	43	300	310	3EA		3SA				
35	301	31	100			12R		1JU	1LD	1MA	1PA
36	305	32	200	230	2EA			222	211		
37	206	23	100		1EB	12C		1JU	1LC	1MA	1PA
38	201	22	100		1EB	12C		1KE	1LB	1MB	1PA
39	201	22	200	210	2EA			222	233		
40	201	22	300	330	3EA		3PO				
41	202	22	200	220	2EB				211	233	
42	202	22	300	320	3EB		3SA				
43	203	22	300	310	3EA		3SA				
44	204	12	100		1EB	12C		1JU	1LC	1MB	1PB
45	204	13	200	220	2EB				211	233	

46	205	23	200	210	2EA			222		233	
47	205	23	300	310	3EA			3PO			
48	206	23	100		1EB	12K		1KE	1LD	1MA	1PA
49	206	23	300	310	3EB			3SA			
50	207	23	100		1EB	12C		1JU	1LB	1MB	1PB
51	207	23	200	210	2EA				222		233
52	207	23	300	310	3EA			3PO			
53	208	24	200	220	2EB					211	233
54	209	24	300	320	3EB			3PO			
55	210	24	100			12C		1JU	1LC	1MA	1PA
56	210	24	200		1EB	12C		1JU	1LD	1MB	1PB
57	210	24	200	220	2EB					211	233
58	210	25	300	310	3EB			3PO			
59	211	25	100		1EA	12C		1JU	1LD	1MA	1PA
60	211	25	200	210	2EA				222		233
61	211	25	300	320	3EA			3PO			
62	213	26	300	330	3EA			3SA			
63	212	25	100		1EA	12C		1JU	1LB	1MA	1PA
64	212	25	300	330	3EA			3SA			
65	213	25	100		1EB	12K		1KE	1LB	1MA	1PA
66	214	11	200	220	2EA					211	233
67	214	11	300	330	3EA			3SA			
68	215	26	200	220	2EA					211	233
69	216	26	100			12C		1KE	1LD	1MA	1PA
70	216	26	100		1EA	12C		1KE	1LD	1MB	1PB
71	216	26	200	210	2EA				222		233
72	216	26	300	310	3EB			3PO			
73	301	31	200	220	2EB					211	233
74	301	31	300	310	3EA			3PO			
75	302	31	100		1EB	12C		1JU	1LD	1MA	1PA
76	302	31	200	220	2EA					211	233
77	302	31	300	310	3EA			3PO			
78	303	13	100		1EB	12C		1KE	1LB	1MB	1PB
79	303	12	200	210	2EB				222		233
80	303	11	400	430	4EB						
81	304	31	300	320	3EB			3SA			
82	305	32	100			12R		1JU	1LB	1MB	1PA
83	305	32	300	310	3EB			3PO			
84	306	32	200	220	2EA					211	233
85	306	32	300	320	3EA			3SA			
86	307	32	200	220	2EB					211	233
87	307	32	300	320	3EB			3SA			
88	308	33	100		1EA	12K		1JU	1LD	1MA	1PA
89	309	33	100		1EB	12R		1KE	1LA	1MB	1PA
90	309	33	200	220	2EA					211	233
91	310	34	100		1EA	12C		1KE	1LB	1MB	1PA
92	310	34	200	220	2EA					211	233
93	310	33	300	310	3EA			3SA			
94	310	33	400	430	4EA						
95	311	11	100			12C		1KE	1LD	1MA	1PB
96	311	16	200	220	2EB					211	233
97	311	17	300	320	3EB			3SA			
98	312	34	100		1EA	12K		1JU	1LC	1MA	1PA
99	312	34	300	310	3EB			3SA			
100	313	34	100		1EB	12K		1KE	1LA	1MA	1PA
101	313	34	200	210	2EB				222		233
102	313	34	300	310	3EA			3PO			
103	314	35	100		1EA	12C		1JU	1LC	1MB	1PA
104	314	35	200	210	2EB				222		233
105	314	35	300	320	3EA			3PO			
106	315	17	200	230	2EA				222	211	
107	315	14	300	320	3EB			3PO			
108	316	35	100		1EB	12R		1JU	1LB	1MB	1PA
109	316	35	200	220	2EB					211	233
110	316	35	200	220	2EA					211	233
111	317	35	300	310	3EB			3SA			
112	401	41	100			12C		1KE	1LC	1MA	1PA
113	401	41	200	210	2EA				222		233
114	401	41	300	320	3EB			3PO			
115	402	41	100		1EA	12R		1JU	1LC	1MA	1PA
116	402	41	300	310	3EB			3SA			

Copyright © IES 2018

The 22nd Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES2018)

117	403	41	100		1EA	12R		IJU	ILC	1MA	1PB
118	403	41	100		1EA	12R		1KE	ILD	1MA	1PA
119	403	41	300	310	3EB		3SA				
120	404	42	300	320	3EA		3PO				
121	405	14	100		1EA	12C		1KE	1LB	1MB	1PB
122	405	12	300	320	3EB		3SA				
123	406	42	100		1EB	12C		1KE	ILD	1MA	1PA
124	406	42	200	210	2EA			222		233	
125	406	42	300	310	3EA		3SA				
126	407	42	100		1EA	12C		1KE	ILA	1MA	1PA
127	407	42	200	220	2EB			211	233		
128	408	43	100		1EB	12C		IJU	ILA	1MA	1PB
129	408	43	200	210	2EA			222		233	
130	409	43	200	210	2EA			222		233	
131	409	43	300	320	3EB		3SA				
132	410	43	200	210	2EA			222		233	
133	410	43	300	320	3EA		3SA				
134	411	44	100			12C		IJU	ILA	1MA	1PA
135	411	44	300	310	3EB		3SA				
136	412	44	100		1EA	12R		IJU	ILC	1MA	1PA
137	412	44	300	310	3EA		3SA				
138	413	44	200	210	2EB			222		233	
139	414	45	100		1EA	12C		IJU	ILA	1MA	1PA
140	414	45	200	210	2EA			222		233	
141	414	45	300	310	3EA		3PO				
142	415	15	100		1EB	12K		1KE	1LB	1MB	1PB
143	415	15	200	210	2EA			222		233	
144	415	13	300	310	3EA		3SA				
145	416	45	100			12C		IJU	ILC	1MA	1PA
146	416	45	200	210	2EA			222		233	
147	416	45	300	310	3EB		3PO				
148	417	45	100		1EA	12K		IJU	ILA	1MA	1PA
149	417	45	300	310	3EA		3SA				
150	417	45	400	430	4EB						

31	11	1	3	300							
32	11	1	4	430							
33	11	2	1	ILA							
34	11	2	1	ILB							
35	11	2	1	ILC							
36	11	2	1	ILD							
37	11	2	2	210							
38	11	2	2	220							
39	11	2	2	230							
40	11	2	3	3EA							
41	11	2	3	3EB							
42	11	2	4	430							
43	11	3	1	IKE							
44	11	3	1	IJU							
45	11	3	2	210							
46	11	3	2	211							
47	11	3	3	300							
48	11	3	4	430							
49	11	4	1	IJU							
50	11	4	1	IKE							
51	11	4	2	222							
52	11	4	3	310							
53	12	1	1	100							
54	12	1	2	210							
55	12	1	2	220							
56	12	1	2	230							
57	12	2	2	210							
58	12	2	2	220							
59	12	2	2	230							
60	12	2	4	430							
61	12	3	2	200							
62	12	4	all	555							
63	13	1	1	12C							
64	13	1	1	12K							
65	13	1	1	12R							
66	13	1	2	200							
67	13	1	3	3PO							
68	13	1	3	3SA							
69	13	1	4	430							
70	13	2	1	12C							
71	13	2	1	12K							
72	13	2	1	12R							
73	13	2	2	200							
74	13	2	3	310							
75	13	2	3	320							
76	13	2	3	330							
77	13	2	4	430							
78	13	3	1	IPA							
79	13	3	1	1PB							
80	13	3	2	210							
81	13	3	2	220							
82	13	3	2	230							
83	13	3	3	300							
84	13	3	4	430							
85	16	1	1	ILA							
86	16	1	1	ILB							
87	16	1	1	ILD							
88	16	1	1	ILD							
89	16	1	2	233							
90	16	1	2	230							
91	16	1	3	310							
92	16	1	3	320							
93	16	1	3	330							
94	16	1	4	430							
95	16	2	1	100							
96	16	2	2	200							
97	16	2	3	3PO							
98	16	2	3	3SA							
99	16	2	4	430							
100	16	3	1	12C							

The Influence of Social Media Writing on Online Search Behavior for Seasonal Events: The Sociophysics Approach

Nozomi Okano[†] Masaru Higashi[†] Akira Ishii[†]

[†] Faculty of Engineering, Tottori University, Koayam, Tottori 680-8552, Japan
ishii@tottori-u.ac.jp

Abstract: Using seasonal topics as the study subject, in this study, we focus on the timing gap between social media writing and online search behavior. To conduct our analysis, we used the mathematical model of search behavior, comprising the sociophysics approach. The seasonal topics selected were St.Valentine's Day, Halloween and New Year countdown. We also picked up the event like Christmas and Halloween. We analyzed the influence of blogs and Twitter on search behavior and found a deviation of interest in terms of timing. We also analyzed Japanese seasonal event of eating Eho-maki in February 3 and eels at the day of the ox in midsummer.

Keywords: blog, Twitter, Google Trends, St.Valentine's Day, Halloween, countdown

1. Introduction

Some people believe that Twitter and Blogs are enthusiastically posted, but most people are supposed to search on the Internet. Therefore, analysis of the search behavior of people of society is very important in grasping social movements. In this research, we analyze theoretically by the idea of social physics, how television information, net information, or Blog, Twitter etc. influences people's search behavior.

The subjects of the analysis are seasonally limited dates, and many people participate in the event. Specifically, Valentine's Day, Halloween, Christmas, and New Year's Countdown. In addition, in the habit peculiar to Japan, we also analyzed Eho-maki, which eats scroll sushi on February 3, and an event to eat eel on the Midsummer Occasion Day of Summer (the day of ox in midsummer). In particular, Specifically, Valentine's Day, Halloween, Christmas, and New Year's Countdown last for only one day and thus, it is easier to analyze the rising excitement and subsequent wane in interest. In many European countries, people return to their homelands for Christmas and enjoy a long Christmas break. In contrast, in Japan, however, Christmas is a one-day event in which young people have Christmas parties with friends but do not return home. In Japan, therefore, it is not until the New Year's vacation of the following week that people return to their hometowns. Thus, in Japan, Christmas is very similar to Halloween.

In Japan, Feb.3 is the traditional day to eat "Eho-maki". Eho-maki are thick sushi rolls shown in fig.1 which is believed to bring good fortune if eaten while facing the year's "Eho" (good luck direction of the year) like fig.2. Not all, but many Japanese people have Eho-maki at this day.

On the day of the ox in midsummer Japanese have a



Fig. 1: Eho-maki in Japan.



Fig. 2: How to eat eho-maki in Japan.

custom to eat eel which started in the Edo period, 18th century. Eel is a popular food for Japanese people, and it is expensive, so eating eel on the day of the ox in midsummer once a year is a big concern. On the day of the ox in midsummer Japanese have a custom to eat eel which started in the Edo period. The

day of the ox, which is named after one of the twelve animals of the Chinese zodiac. According to one legend, long before the scientific reasons were established, in the 1700s well-known scholar Gennai Hiraga came up with the custom as part of a marketing ploy to boost limp summer sales when the owner of a struggling eel store asked the wise man for some business advice. In fig.3, we show the typical cooking of eel in Japan for the day of the ox in midsummer.



Fig. 3: Typical cooking of eel in Japan.

Thus, this study examines the peaks and falls in interest in these time-limited events, focusing on the medium used to perform searches and on what cohort of the population perform those searches. The mathematical model of search behavior is used for the analysis[1, 2].

2. Theory

In the theory of search behavior [1, 2], the interest and concern on a certain topic can be calculated using a mathematical model of differential equations. Here, we introduce $I(t)$ as the interest or concern on a certain topic. We construct a mathematical model based on the mathematical model for the hit phenomenon within a society presented as a stochastic process of interactions of human dynamics in the sense of many body theory in physics [3, 4]. As in the model in [3, 4], we assume that the intention of humans in a society is affected by the three factors: advertisement, communication with friends, and rumors. Advertisements act as external forces; communications with friends are a form of direct communication and its effect is considered as interaction with the intention of friends. The rumor effect is considered as the interaction among three persons and a form of indirect communication as described. In the model, we use only the time distribution of advertisement budget as an input, and word-of-mouth (WOM) represented by posts on social network systems is the observed data for comparison with the calculated results. The parameters in the model are adjusted by the comparison with the calculated and observed social media posting data.

According to [3, 4], we write down the equation of purchase intention at the individual level $I_i(t)$ as

$$\frac{dI_i(t)}{dt} = \sum_{\xi} c_{\xi} A_{\xi}(t) - a I_i(t) + \sum_j d_{ij} I_j(t) + \sum_j \sum_k p_{ijk} I_j(t) I_k(t) \quad (1)$$

where t is the time, d_{ij} , p_{ijk} , and $f_i(t)$ are the coefficient of the direct communication, the coefficient of the indirect communication, and the random effect for person i , respectively[3]. The advertisement and publicity effects are included in $A_{\xi}(t)$ which is treated as an external force. The index ξ means sum up of the multi media exposures. Word-of-mouth (WOM) represented by posts on social network systems like blog or twitter is used as observed data which can be compared with the calculated results of the model. The unit of time is a day.

Here, it is assumed that the height of interest $I(t)$ of people attenuates exponentially. Although it is known that this is known to occur in movies and the like [3], attention such as events and anniversaries is known to attenuate by a power function. [5, 6] In the case of social interest, we attenuate the intermediate between the exponential function and the power function [7], but here we simply adopt exponential decay.

We consider the above equation for every consumers in the society. Taking the effect of direct communication, indirect communication, and the decline of audience into account, we obtain the above equation for the mathematical model for the hit phenomenon. Using the mean field approximation, we obtain the following equation as equation for averaged intention in the society. The derivation of the equation is explained in detail in ref.[3].

$$\frac{dI(t)}{dt} = \sum_{\xi} c_{\xi} A_{\xi}(t) + (D - a)I(t) + PI^2(t) \quad (2)$$

This equation is the macroscopic equation for the intention of whole society. Using this equation, our calculations for the Japanese motion picture market have agreed very well with the actual residue distribution in time [3]. The advertisement and publicity effects are obtained from the dataset of M Data and the WOM represented by posts on social network systems are observed using the system of Hottolink. We found that the indirect communication effect is very significant for huge hit movies.

2.1 Extension to include Twitter and Blog

In the new mathematical model for search behavior, we use daily blog and Twitter postings as the external force. Therefore, we extend the above mathematical model for hit phenomena to include the effects of Twitter and blog as external field as follows.

$$\begin{aligned}\frac{dI(t)}{dt} &= C_{TV}A_{TV}(t) \\ &+ C_{NetNews}A_{NetNews}(t) + C_{Twitter}A_{Twitter}(t) \\ &+ C_{blog}A_{blog}(t) + (D - a)I(t) + PI^2(t)\end{aligned}\quad (3)$$

In the above equation(3), $I(t)$ is the intention to search a certain topic using Google Trend and C_{TV} , $C_{NetNews}$, $C_{Twitter}$ and C_{blog} correspond to the strength to influence willingness to search the certain topic. $(D - a)I(t) + PI^2(t)$ correspond to the direct and indirect communications in the previous mathematical model for hit phenomena. In the model of the present paper, these terms correspond to the direct and indirect effect of searching the topic by other people.

On the real calculation, we use advertisement time data on TV from M Data co. Ltd. and the internet news site data, daily Twitter posting data and daily blog posting data on the certain topic from Hottolink co.ltd. The daily search data comes from Google Trend as the reference of our calculation.

The parameters C_{TV} , $C_{NetNews}$, $C_{Twitter}$, C_{blog} , D and P are determined in similar way as the previous model by using the metropolis-like Mote Carlo method as noted in the previous paper[3, 4]. We define here "R-factor" to check the correctness of the adjustment of parameters.

$$R = \frac{\sum_i(f(i) - g(i))^2}{\sum_i(f(i)^2 + g(i)^2)}, \quad (4)$$

where $f(i)$ and $g(i)$ correspond to the calculated $I(t)$ and the observed number of Google Trend data. The R-factor is originally defined by J B Pendry[8] to adjust the positions of atoms on surface in the calculation of low energy electron diffraction experiment where measured electric current - voltage curve compared with the corresponding calculation. The smaller the value of R, the better the functions f and g . Thus, we use a random number to search for the parameter set that minimizes R. This random number technique is similar to the Metropolis method[9], which we have used previously[3]. We use this R-factor as a guide to obtain the best parameters for each calculation in this paper.

We employ the Monte Carlo method like Metropolis method[9] to fine the minimum of R. This is very similar for finding the minimum of total energy in the first principle calculation. In the real calculation to adjust the parameters C_x , D , and P , we should take care of the local minimum trapping like the first principle calculation in material physics. It is well-known that there are several ways to find the minimum condition like the steepest descent, the equation of motion method and the conjugate gradient method. Even in the actual calculation of the first principle calculation or the density functional theory, we should be careful of the danger of local minimum trapping. In this paper, the way we employ is just do the calculation using the several initial value in the

Metropolis-like method to avoid the local minimum trapping. To check the accuracy of the parameters adjusting, we use the R-factor value. For every calculation which we show in this paper, the R-factor is below 0.01.

Actually, the parameters C_{TV} , $C_{NetNews}$, $C_{Twitter}$, C_{blog} , D and P in equation (3) can be considered as functions of time, because people's attention changes over time. However, if we introduce the functions $C_{TV}(t)$, $C_{NetNews}(t)$, $C_{Twitter}(t)$, $C_{blog}(t)$, $D(t)$ and $P(t)$, we can tune any phenomena by adjusting these functions. Thus, we retain C_{TV} , $C_{NetNews}$, $C_{Twitter}$, C_{blog} , D and P as constant values to examine whether equation (3) can be applied to any social phenomena.

3. Data

For the investigation of this article, we should use Google Trends data for the target data of our calculation. The data of Google Trends can be obtained on the Google Trends page.

The daily posting number to Twitter and Blog are obtained from the service "Kuchikomi Kakaricyo" of Hottolink co.ltd. The mass media advertisement data can be obtained from M Data co.ltd. via the "Kuchikomi Kakaricyo" service.

4. Result

Analyze the events 1 month before and 1 month after the events, and calculate each coefficients C_{TV} , $C_{NetNews}$, $C_{Twitter}$, C_{blog} , D and P so that the calculated $I(t)$ is combined with the measured value by Google Trend to best match.

The calculation results of all coefficients for Christmas are shown in the fig.4. For Christmas, the calculation results of all coefficients for the first month before Christmas and one month after Christmas are shown in the fig.4.

In particular, here we examine the difference whether search behavior is affected by Blog or influenced by Twitter. Shown in the fig.5 are C_{blog} and $C_{Twitter}$ values before and after Christmas. Looking at the results, Twitter's influence is big before Christmas and blog influence is big after Christmas.

In the same way, the results calculated for Halloween, New Year's countdown, Valentine's Day are shown in the fig.6, fig.7 and fig.8, respectively. For Holloween and Valentine's Day, we analyze the events 1 month before and 1 month after the events. For New Year's countdown, we analyze the event 1 week before and 1 week after the event in order to avoid the effect of the Christmas before Dec.24. From these, it is possible to read the common tendency that Twitter influences strongly before the event, and after the event the effect of Blog is strong.

As mentioned above, qualitatively, Halloween, Christmas, Countdown and Valentine's Day are in agreement that it is influenced by Twitter before the event and after the event is affected by Blog.

Next, we introduce the calculation results of Eho-maki and Eel of the Midsummer's Day eating foods decided on a fixed date in fig.9 and 10. As you can see from the results, the influence of Blog and Twitter on people's search behavior is opposite to the previous examples. In the case of this foods, it is from Blog that is influenced before the event and it is influenced by Twitter after the event. The qualitative behavior is same for the two event foods.

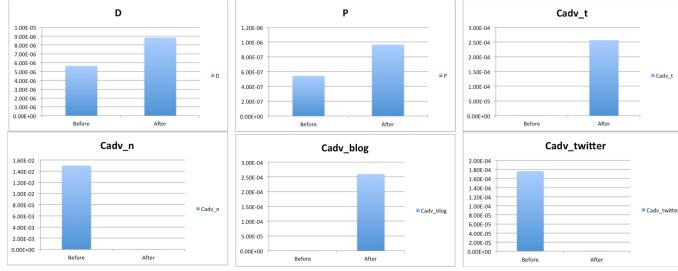


Fig. 4: Result of D , P , CTV , $C_{NetNews}$, C_{blog} , $C_{twitter}$ in calculation for Christmas.

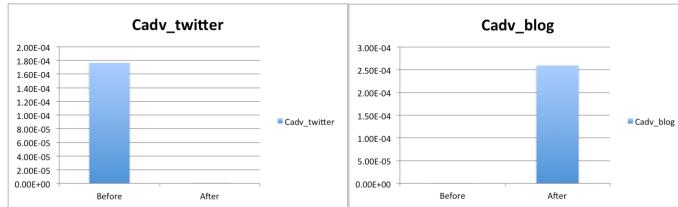


Fig. 5: Result of calculation of $C_{Twitter}$ and C_{blog} before and after Christmas.



Fig. 6: Result of calculation of $C_{Twitter}$ and C_{blog} before and after Halloween.

5. Discussion

We analyzed Christmas, Halloween, New Year Countdown, Valentine's Day, Ewaki Roll, and Eel of the Midsummer Day of the ox as events with limited time. Looking at the analysis results, the results were divided for four of Christmas, Halloween, Countdown, Valentine's Day, and the two of Eho-maki and Eel.

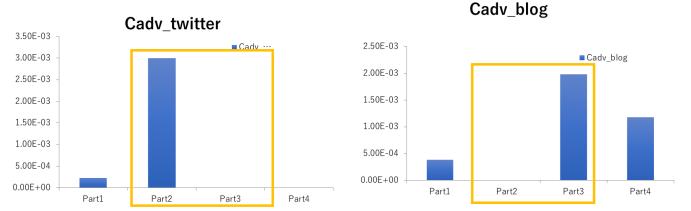


Fig. 7: Result of calculation of $C_{Twitter}$ and C_{blog} before and after New Year Countdown. The analysis is done for a week before the Christmas, a week before the New Year's Day, a week after the New Year's Day and a week after January 8.

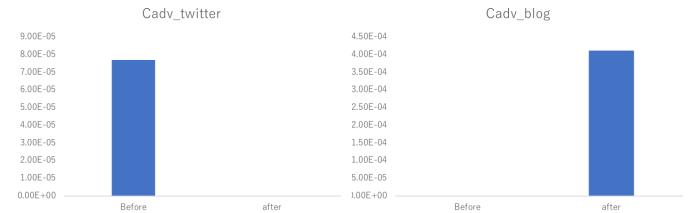


Fig. 8: Result of calculation of $C_{Twitter}$ and C_{blog} before and after Valentine's Day.

These two groups are thought to depend on surprises or happening for the event and its preparation. In Halloween, there are many matters to investigate beforehand, such as what kind of costumes they themselves, what kind of costumes they are going to do, what kinds of costumes will be popular this year. At Christmas in Japan, there are not many things to consider beforehand, such as what kind of surprise there are lovers to book a Christmas dinner with. In the New Year's countdown, since the taste of the countdown is different for each gathering, it is necessary to gather information in advance according to which place to go to, which event to go to and which bar to go to. At Japanese Valentine's Day, women collect information in advance, whether women make their own chocolate for her lover or they purchase high-end chocolate at some famous brand shop.

Meanwhile, there are few kinds of Eho-maki to eat as an event, and there is no element to look into in advance as there are also decided how to eat. Also, there is no surprise when eating Eho-maki. As for the Eel to eat on the Midsummer Day of the ox, as shown in the fig.3, the method of cooking has been decided traditionally. It is impossible to make home made, so Japanese people have to eat Eel at a restaurant. Therefore, the information to be checked in advance is the only restaurant to eat. Besides, there is not much difference in Eel's cuisine for each restaurant.

In this way, events that need to check information sufficiently in advance are affected by Twitter. On the contrary, in the case of an event where there is no surprise and there



Fig. 9: Result of calculation of $C_{Twitter}$ and C_{Blog} before and after the Eho-maki.

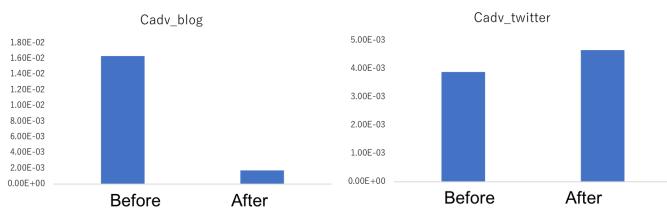


Fig. 10: Result of calculation of $C_{Twitter}$ and C_{Blog} before and after the day of the ox in midsummer.

is no need to check in advance, the effect of Twitter at the prior stage is small, and the influence of Blog is expected to be relatively high.

Therefore, if search actions that are strongly influenced by Twitter are observed beforehand, those who come to the event are collecting information. For those people, event-related marketing will be effective. In this way, the method of this research can be applied to marketing.

6. Conclusion

Posting on Blog and Twitter is an act performed by some people in society. On the other hand, search behavior is used by most people who use the Internet. Therefore, analysis of search behavior on the Internet is very important in social analysis in that it can also target people without voice.

In this research, we have analyzed such search behavior on the Internet using mathematical model of search behavior. As an object, we observed preliminary excitement and post cool down at the event to be held in a short time. The events analyzed are Halloween, Christmas, Countdown, Valentine's Day, Eho-maki, Eel of the Midsummer Day of the ox in Japan. According to the analysis, it turned out that these are divided into events requiring preparatory preparations for surprises, and events not being prepared. Twitter has a strong influence on events requiring advance preparations for surprises. In other events the effect of Twitter is after the incident.

This research result is expected to be applicable to marketing.

Acknowledgement

The authors are grateful for helpful discussion to Yasuko Kawahata of Gunma University, Japan.

References

- [1] A Ishii, T Wakabayashi, N Okano and Y Kawahata, "Research on social media writing and search behavior about seasonal topics using sociophysics approach" Proceedings of WMSCI2018 21-24
- [2] N Okano, M Higashi, T Wakabayashi, Y Kawahata and A Ishii, "Analysis of seasonal events on social media and internet search using sociophysics model" Proceedings of ICNAAM2018 in press
- [3] Ishii A, Arakaki H, Matsuda N, Umemura S, Urushidani T, Yamagata N and Yoshida N, "The 'hit' phenomenon: a mathematical model of human dynamics interactions as a stochastic process", *New Journal of Physics* 14 (2012) 063018
- [4] Ishii A. (2017) "Analysis and Predictions of Social Phenomena via social media using Social Physics method", A. Ishii, INFORMATION 20, 7073-7086
- [5] Sano Y, Yamada K, Watanabe H, Takayasu H and Takayasu M: Empirical analysis of collective human behavior for extraordinary events in the blogosphere, *Phys.Rev.E87*, 012805, 2013.
- [6] Sano Y: Empirical analysis and modeling of word frequency time series in social media, *doctoral thesis*, Tokyo Institute of Technology, 2013
- [7] Ishii A and Koyabu T, "Analysis of behavior of attenuation of social memories on movie and social scandal using sociophysics approach" *Proceedings of the The 47th ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, Honolulu, Dec. 5-8, 2015, Hawaii, pp.204-209
- [8] Pendry JB, 1980, "Reliability factors for LEED calculations", *J. Phys. C13* 937 (1980)
- [9] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, and Teller E, 1953 "Equations of State Calculations by Fast Computing Machines", *Journal of Chemical Physics* 21 (1953) 1087

The Emergence of Flocking Behaviors with Indirect Reward

Hiroshi Sato [†] Naoya Sekiguchi [†]

[†] National Defense Academy, Yokosuka, Japan
hsato@nda.ac.jp

Abstract: In the natural world, there are many animal species that swarm and flock. The benefits of flocking are (1) a group of the animal can aware the danger better than being alone, (2) making a large group can prevent an enemy from attacking, (3) being in the group increase the chance of mating, (4) individuals can keep their body temperature in the cold environment. However, it is not clear how they find themselves to make a group even if they don't know anything about the benefit of flocking. In this study, we investigate the behaviors of agents that model the fishes in the tank of water. The agents can learn from the environment and other agents using reinforcement learning. We find computer what kind of factors is essential to the discovery of flocking from the simulation results.

Keywords: Flocking Behaviors, Reinforcement Learning, Neural Network

1. Introduction

In the natural world, there are many animal species that swarm and flock. The benefits of flocking are (1) a group of the animal can aware the danger better than being alone, (2) making a large group can prevent an enemy from attacking, (3) being in the group increase the chance of mating, (4) individuals can keep their body temperature in the cold environment. However, it is not clear how they find themselves to make a group even if they don't know anything about the benefit of flocking.

When we simulate artificial animal behaviors, two methods have been mainly used. The first is a top-down approach in which we analyze the actual behavior of living creatures and implement agents based on a mathematical. The second method is a bottom-up approach in which we try to reproduce group behavior using machine learning.

The Boid model[1] is the major example of the former model. Other models which are implemented as a mathematical model can be thought as the variation of Boid model. With the simple mathematical formulation, the Boid model is excellent in drawing of flocking behavior in Computer Graphics. However, the mathematical expressions are just for a reproduction of the trajectory only in a specific situation under the simple environment, and it does not adapt flexibly to various environments. Therefore, the latter seems to be more suitable for expressing the adaptability of flocks in a changing environment like nature.

As the latter method, there are various approaches such as genetic algorithm and reinforcement learning. For example, Morihiro et. al. [2] proposed a framework in which agents can acquire appropriate interaction with other agents using reinforcement learning and can reproduce group behavior.

The research indicated that it is possible to create herding behavior using reinforcement learning, but the model used there does not consider the conditions in the real environment, biological information, etc. A reinforcement learning used in the model gives positive and negative rewards by actions according to the distance between other agents. This value is calculated by the characteristics of the separation, alignment, and coupling of the Boid model. The research study about the emergence of herding behavior of the prey which comes from the interaction between the predator in the environment, but predator's behavior is just passing through the environment in a straight line, the concept of chasing the prey by the predator was ignored.

In this research, we propose a reinforcement learning model using objects, parameters, and rewards that seem to be natural. We also aim to investigate the characteristics of prey expressed by group behavior simulation. The situation assumed in this research is that the predation of fish swimming in an aquarium (Prey-Predator relation), targeting the emergence of flocks by prey not giving a reward which corresponds to making a group.

2. Learning Model of Agents

2.1 Q-Learning

In this research, we use the Q-learning[3] as a learning method of the agent which is described in the previous section. Q-learning is a method of giving an evaluation called a Q-value to a series of state-action pairs and learning this evaluation value at the same time. In Q-learning, Q-value is updated using Q-value. The Q-value of the action linked to the reward is increased when you get the reward. The behavior of the

agent is selected based on the action taking the maximum Q-value among the following states. The goal of Q-learning is to construct an appropriate state-behavior evaluation value by repeating this operation.

Equation (1) shows the Q-value updating equation in Q-learning.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \quad (1)$$

Here, s represents a state, and a represents an action. r is the reward when taking action a in state s . γ is the discount rate and represents how much emphasis is placed on the Q-value obtained in the previous action.

2.2 Q-Learning using Neural Network

In this paper, we implemented reinforcement learning using a neural network[4] for prey agent. In the conventional reinforcement learning, state-action space is discretized and description using Q-Table is performed. However, in the situation assumed in this research, the number of states becomes enormous when we use the conventional Q-Table, so we decided to use a neural network as a function to calculate the Q-value. Update of the weight of the neural network takes an error of the Q-value calculated by the previous action and the current action and uses the Q-value with the highest evaluation as the training data. The training signal used in the neural network is expressed by equation (2).

$$T = r + \gamma \max_{a'}(Q(s', a') - Q(s, a)) \quad (2)$$

The error is shown in equation (3).

$$L_\theta \frac{1}{2}(T - Q_\theta(s, a))^2 \quad (3)$$

Reinforcement learning with a neural network can be implemented according to equation (3). The concrete framework of reinforcement learning in this research is shown in Fig. 1

3. Simulation Settings

3.1 Environmental Setting

In this study, we simulate a group of fish swimming in the aquarium. The simulation status is shown in Fig. 2. In a circular tank environment, there are a large number of preys (green), a small number of predators (red), foods (yellow) and obstacles (white). Details of the implemented environment are shown in Table 1.

3.2 Agent Setting

Prey is a learning agent. In this research, we implemented the agent based on biological spec. Each prey learns independently by reinforcement learning mentioned in section 2.

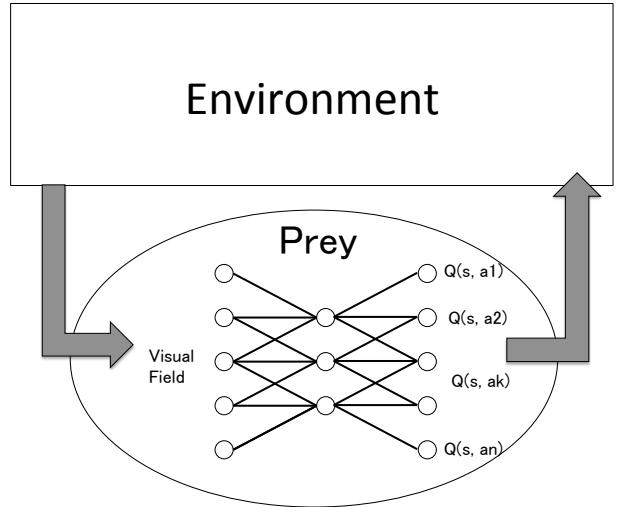


Fig. 1: Framework of Q-Learning using neural network

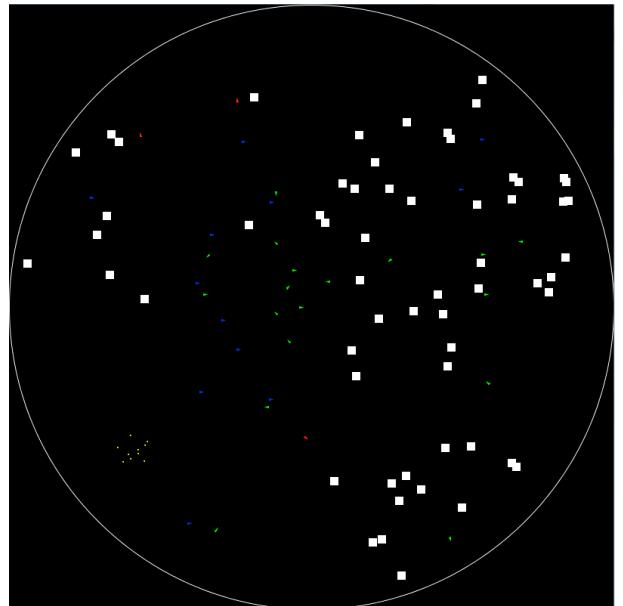


Fig. 2: An appearance of computer simulation

Predator chases prey and eats them. As a predatory activ-

Table. 1: Specification of the simulation

Property	Description
Area	A circle with radius 800 unit
Number of Agents	Prey: 30 Predator: 4 Food: 10 Block: (0, 10, 20, 30)
One Cycle	(1) until all prey are eaten (2) until 50,000 step passed

ity, we define that prey dies if it contacts predator. Died prey is removed from the environment during the episode. Also, when predation, a negative reward is given for prey. Food is set in the environment so that if prey can consume the food when it is touched. By eating food, the prey gets physical fitness recovery and positive reward for reinforcement learning. Block is an obstacle in the environment.

The prey perceives the environment through visual information. Fig. 3 shows the view field of agents. Recently, many research deal with the field of view as a set of pixels [5] [6], but it needs heavy computational load. In this study, the field of view does not consist of pixels, but eight segments separated by 40° due to the limitation of computational resource. The direction of advance of Prey was set to 0° . The dead angle of the fish was set to 40° backward. Each view segment has nine bits (three bits for counting prey, two bits for counting predator, two bits for counting food, and two bits for counting). Therefore the entire view field has 72 bits.

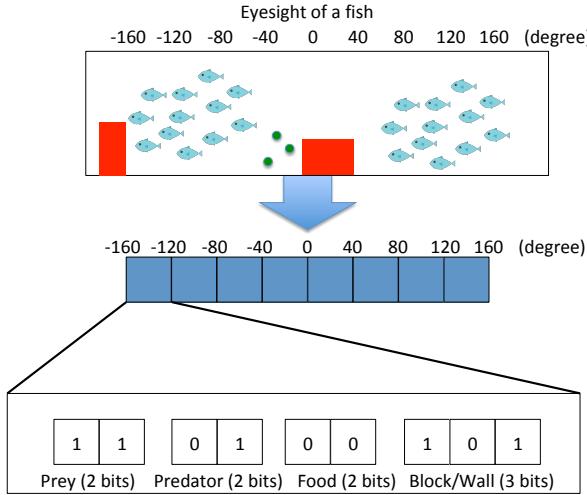


Fig. 3: Environmental information as an input of neural network

As stated above, the prey has Q-learning which has a neural network instead of Q-table. The neural network consists of three layers — an input layer, a hidden layer, and an output layer. The input layer has 72 nodes, the hidden layer has 100 nodes, and the output layer has 12 nodes. The initial weight of the network is determined randomly. We use the sigmoid function and ramp function as the activation function.

4. Computer Simulation

4.1 Purpose and Evaluation

The purpose of this simulation is to discover a combination of parameters such that flocking behavior is created and to examine what is important to derive the behavior of swarm.

As the evaluation method, we adopt the degree of agreement between the population density of the herd and the velocity vector of the herd. For estimating the density of flocks, expressions of kernel density estimation are shown, and degree of coincidence of velocity vectors is expressed by cosine similarity.

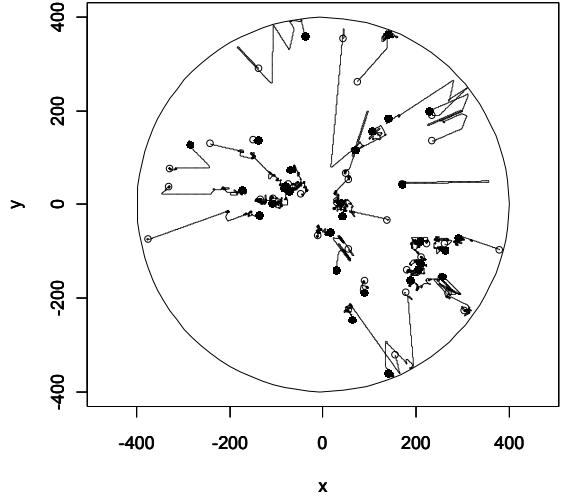


Fig. 4: The behaviors from Ramp function as the activation function. White circle: Initial position of the prey, Black circle: End position of the prey.

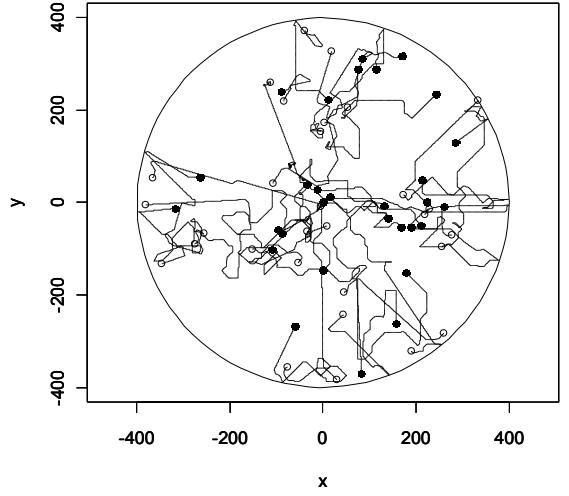


Fig. 5: The behaviors from Sigmoid function as the activation function: Sigmoid function. White circle: Initial position of the prey, Black circle: End position of the prey.

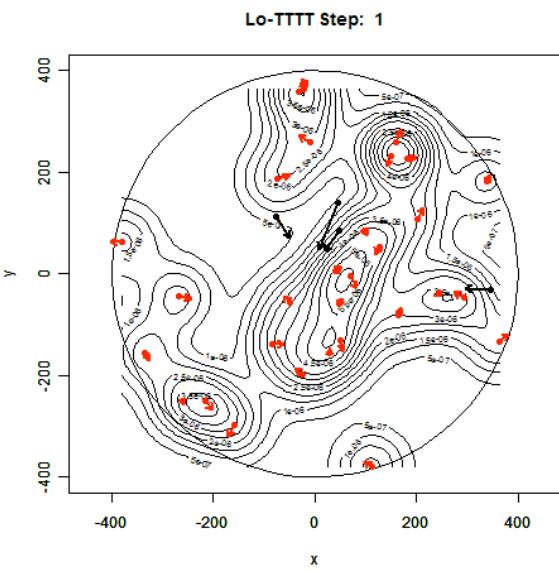


Fig. 6: Typical behavior of the flock (1). In this case, the prey evaluates the speed of the flock. Red: prey, Black: predator.

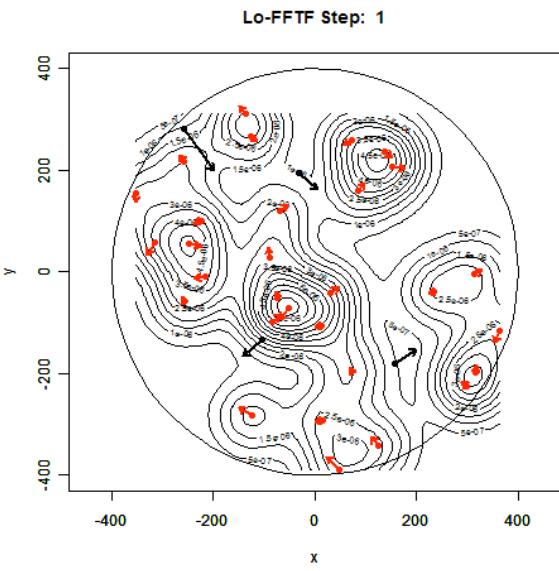


Fig. 7: Typical behavior of the flock (2). In this case, the prey evaluates the density of the flock. Red: prey, Black: predator.

4.2 Experiment 1

Firstly, we examined the learning results due to differences in the setting of the neural network. The behavior of the agent has changed greatly due to the difference in the activation function. Representative patterns in experimental results are shown in Fig. 4 and Fig. 5. When the prey used the sigmoid function, the emerged flocking behavior has a tendency to

move around, but when the prey used the ramp function, the emerged flocking behavior has the tendency to stay longer.

4.3 Experiment 2

Next, we examined the learning results due to the difference in environment settings. The activation function of the neural network, with/without food, with/without obstacles, Predator always exists/intermittently, with/without water flow. We represent the parameter setting as "Lo-TFTT" which means "prey with logistic (sigmoid) activation function, with food, without obstacles, with predator which always exist in the environment, with water flow.

Two examples of typical behavioral flocking pattern in the experiment results are shown in Fig. 6 (speedy flock) and Fig. 7 (dense flock). We tried other settings among all parameter combination.

In the above setting, the factor that most influenced the formation of the herd was the predator. When the predator is always in the environment, the herd is not formed and the prey tends to be dispersed in the environment. When the predator appears intermittently, prey individuals tend to densely populate and run in parallel.

The next factor that influenced next was the presence of obstacles. In the absence of obstacles in the environment, Prey tended to diffuse into space. It turned out that a flock was formed as a result of the presence of obstacles.

The next factor affected was the presence of food. In the case of food, the search behavior of prey became active, and in the absence of food, prey was devoted to escaping from the predator.

5. Conclusion

In this research, we investigated what kind of factors are required for reinforcement learning agents with information from view as input as an agent-based simulation. Experiments showed that the behavior of Prey changes greatly when changing the activation function of the work in the same environment. Next, it was shown that when we further advanced the learning, the environment suitable for the environment appears. By reinforcement learning using the neural network, it is considered that the biological model could be presented that can respond flexibly to changes in the environment. Finally, we analyzed the behavior of prey that occurred by changing the surrounding environment. In each condition, the presence or absence of feed affects the search behavior of prey, whether or not the obstacles and predator are always present affects the density and behavior of prey, and the water flow can increase the density and parallelism I understood. In the future, we will verify the validity of the result by conducting a more detailed simulation.

References

- [1] Reynolds, Craig W. "Flocks, herds and schools: A distributed behavioral model", ACM SIGGRAPH computer graphics, Vol. 21, No. 4, pp. 25–34, 1987
- [2] K. Morihiro, H. Nishimura, T. Isokasa, N. Matsui, Modeling for Flocking-Behavior emergence by Reinforcement Learning, The 49th Association Symposium on Automated Control, OS 7, 2006 (In Japanese)
- [3] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8.3-4: pp. 279–292, 1992
- [4] K. Yamada and K Ohkura, Network Parameter Setting for Reinforcement Learning Approaches Using Neural Network, Journal of The Japan Society of Mechanical engineers, Vol. 78, No. 792, pp. 2950–2961, 2012
- [5] The Whole Brain Architecture Initiative, Life in Silico (LIS), <https://github.com/wbap/lis>, (2016)
- [6] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, et. al. Emergence of Locomotion Behaviours in Rich Environments, arXiv:1707.02286 (2017)

When the Selfish Herd is Unsafe in the Middle

Wen-Chi Yang [†]

[†] Henan Institute of Technology, Xinxiang 453003, China

w.yang@hait.edu.cn

Abstract: The selfish herd hypothesis suggests the competition for secure central positions can shape social animals into cohesive groups in predation. However, this scenario is challenged by the fact that several species of prey organisms exhibit herding tendencies even if the interior of a group carries the predation risk as well. This paper investigates this issue by adopting an evolutionary model to simulate prey's adaptation at various internal risk levels. In the game, agents are put in a periodic lattice to look for positions where they receive better fitness scores than the others, subject to the constraint on body overlaps. The fitness score is a function of the number of neighbors at each time step, which controls the risk inside a group. The experiments demonstrate that under the selective force at the individual level, prey agents consistently evolve a herding tendency even though the internal risk becomes considerably higher than the risk at the group border. It is because the strategy developed for one's own relative fitness indirectly leads to the emergence of selfish herds. This type of selfish herd has lower mobility and higher cohesiveness compared with the emergent pattern when the group center is riskless.

Keywords: selfish herd, intraspecific competition, individual selection, collective behavior, evolutionary game, artificial life

1. Introduction

Various species of organisms exhibit herding tendencies under the threat of predation. For example, shoal fish assemble densely packed aggregates when facing predatory hazards [1]. Vigilant birds and sheep also flock tightly as a response to danger [2, 3]. Ecological studies have highlighted the anti-predator functions brought from these collective patterns [4, 5, 6]. Hence, the survival pressure has long been inferred the primary force which drove gregarious prey into compact groups under predation risk [7].

Regarding how this flocking behavior was selected in evolution, the selfish herd hypothesis [8] has long been considered a reasonable explanation. From the original model to recent extensions and variations [9, 10, 11], it was demonstrated that given the margin of a group more dangerous than the central area, evolution would favor those prey agents which were less frequent to stay at the group edge. Consequently, prey agents evolved to move inward a group for better individual fitness, and the flocking behavior emerged.

Most fieldwork studies on predator-prey interactions did support the assumption that being inside a group is more secure than being at the margin [3, 12, 13]. The phenomenon naturally occurs in several hunting behaviors since a predator usually chases a prey population from outside and attacks the nearest individual [8]. Moreover, it was reported some predatory fish specifically prefer the border prey to the inner ones regardless of the distance [4, 5]. These facts suggested

that the selfish herd scenario would happen on lots of predator-prey systems.

As indicated by Hamilton himself [8], however, a conundrum is that there are also predators which attack the inside of prey groups, but the herding tendency of their prey remains. For example, sailfish and swordfish slash fish schools by their long bills and then feed on those injured prey [14]. Thresher sharks and killer whales also practice similar skills by their tails [15, 16]. Besides, many predatory cetaceans, from dusky dolphins to humpback whales, treat the whole fish aggregate as a hunting target [17, 18]. In these cases, prey fish still flock tightly even though hiding in the crowd is no longer safe.

The insecure-inside condition has been simulated by several modeling works. However, in most models [19, 20, 21], the flocking behavior was a predetermined property, rather than an option of adaptation, and the focuses were on the analogy to natural animals' collective motion. Some have investigated prey's behavioral evolution under specific environments where the group center is insecure [22]. In these simulations, the herding tendency did not evolve once the predation risk biased to the group interior. This discordance implies the evolution of animal flocking behavior may not be modeled adequately by the traditional selfish herd scenario.

An issue is that previous selfish herd models used to simplify prey agents into mobile particles so that the overlapping condition between individual bodies was tolerated. Nevertheless, the individual movement in real animals is strictly obstructed by the physical existence of other conspecifics. This

difference may cause some movement rules demonstrated in previous selfish herd models, like directly entering the group center [8], unrealistic and infeasible to real animals. It has recently been revealed that the selfish herd scenario can model a broader range of collective patterns with an explicit constraint on overlapping conditions [23]. However, this work was based on the typical selfish herd environment, where the group center is riskless.

In this paper, the overlap-free selfish herd model [23] is adopted and extended to reexamine the behavioral evolution of group-living animals when the middle of a herd also has predation risk. Through the design to let the positional risk no more a monotonic transformation of the number of neighbors, the research question can be simulated and analyzed in an effective and robust way. The following sections will first display the overlap-free selfish herd model designed for the experiments and then report the simulation outputs. Discussion and conclusion of the findings are put in the last part, which emphasize the secure-inside condition is not necessary to the formation of selfish herds.

2. Design of the Model

The spatial evolutionary game proposed in [23] is introduced and modified for the experiments. As a general picture of the present model, agents are put on a periodic lattice to find neighbors for cover from predation. In the duration of movements, the fitness value of each agent is evaluated based on the number of neighbors at each time step. After sufficiently many time steps, those agents with the least fitness values are eliminated and replaced by the offspring agents of the better-adapted ones.

The lattice size is set to 120×120 , and the population size is set to 200. Compared with the original setting, as 500 agents on a 100×100 lattice, the population density is much reduced to ensure the necessity of the evolution of flocking behavior. The occupancy state, s_o , of each of the 120×120 grids is defined as follows: if the grid is vacant, the state is set to 0; otherwise, it is set to 1, as being occupied (Fig. 1A). In this game, each grid can only be occupied by at most one single agent, which design avoids the overlapping condition between agents.

The neighboring situation of a grid, s_n , is the number of agents on the eight surrounding grids (Fig. 1B). In other words, it is the summation of the occupancy states of the four adjacent grids and the four diagonal grids, which set is termed by M :

$$s_n = \sum_{g \in M} s_o^g. \quad (1)$$

The purpose of taking the diagonal grids into account is to smoothen the difference in the number of neighbors after a move, which improves the robustness of simulations.

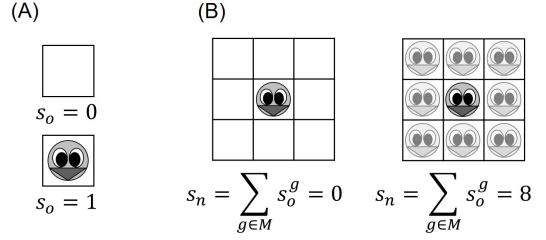


Fig. 1: Definition of occupancy states and neighboring situations. As shown in subfigure A, the occupancy state of a grid is 0 if the grid is empty. Otherwise, it is 1. The neighboring situation is the addition of the eight nearby grids' occupancy states. Subfigure B displays the minimum and maximum values of this index.

The local information about the occupancy states and neighboring situations of the four adjacent grids is acquired by an agent to make its movement decision at each time step, which is a choice between heading one of the four adjacent grids. If the target location has been occupied, the agent then stays at the same place, to satisfy the overlap-free constraint [23]. Hence, from the viewpoint of an agent, the decision is to stay, by heading an occupied grid, or to move towards a certain vacant grid (Fig. 2A). Exceptions are the ones which adjacent grids are all empty or all occupied. In these situations, one can only move, or stay, accordingly.

Each agent's strategy, as the mapping from local information to movement decisions, is implemented by an evolutionary neural network from the NEAT algorithm [24], composed of 10 input nodes and 4 output nodes (Fig. 2B). The arrange-

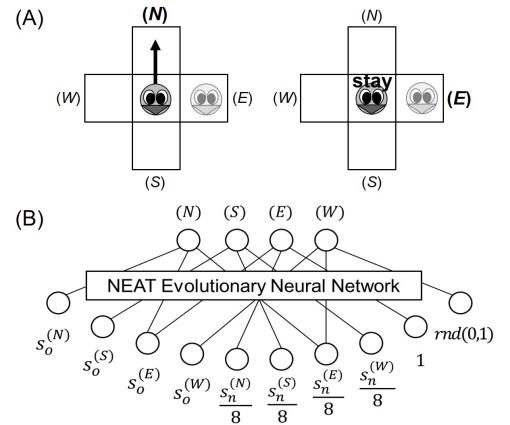


Fig. 2: The movement decision of an agent. At each time step, an agent chooses either to stay, by heading an occupied grid, or to move towards an empty adjacent grid. This choice is controlled by a NEAT evolutionary neural network, which outputs the movement direction based on the local information.

ment of input nodes is that eight nodes are used to store the occupancy states and neighboring situations of the four adjacent grids, another one node sends the constant value 1, and the last node delivers a uniformly distributed random number for the construction of a mixed strategy if necessary. The output nodes represent the four directions: north, south, east and west, respectively. Given a set of input values, the output node with the maximal value then triggers its associated movement direction.

The evolutionary procedure follows that in [23]. After each agent's fitness value, u_i , got evaluated (described later), the worst 10% individuals are eliminated in this generation and replaced by the same number of new agents. These new agents are bred from selected pairs of well-performed ones, based on the reproduction probability:

$$r_i = \frac{1.02^{-rnk(u_i)}}{\sum_{j=1}^N 1.02^{-rnk(u_j)}}, \quad (2)$$

where N is the population size and $rnk(u_i)$ is the rank of agent i ordered by its fitness value u_i . Through the NEAT operators of crossover and mutation [24], these offspring agents inherit the beneficial network topologies of their parents with certain additional innovations, and hence can be more adapted than the discarded ones. These agents then interact with one another and experience the evolutionary process again until the 2000th generation is attained.

The individual fitness value is defined as the average security score of an agent per time step in a given period, i.e. 2000 time steps in the model. The security score, s_s , is a function of the neighboring situation, s_n , and expresses the survival benefit of staying at a particular position. For example, it was set $s_s = s_n$ in [23], as the typical selfish herd scenario, where one is safer if surrounded by more neighbors. Here this function is redefined to simulate the condition of insecure central positions.

Based on the neighboring situation, whether an individual is in the middle of a herd can be detected. For example, those individuals surrounded by the maximum number of neighbors, i.e. $s_n = 8$, must be in the group center. Decreasing the security scores of large s_n values then equals adding the predation risk to the internal area of prey herds. In this model, the security score is defined by

$$s_s = s_n - \mu \times \max(s_n - 4, 0), \quad (3)$$

where $\mu \geq 0$ controls the internal risk level of a group.

This function illustrates that before half of the eight nearby grids are filled with agents, the security score of the focal position increases gradually, as a general case of the dilution effect. However, the score is then depressed with the number of additional neighbors, as the insecure-inside situation. A detail is the score of large s_n values can be negative given a large μ , so an agent inside a herd is possible to obtain a

negative fitness value. It does not affect the evolutionary procedure because the reproduction probability, r_i , is calculated by ranks rather than absolute fitness values.

3. Results

The results are organized into five parts. At first, an overview of the simulation outputs at different internal risk levels, μ , is presented, where three types of emergent patterns are identified. In the following three subsections, the evolutionary trajectories of these three types are analyzed. The last subsection explains the underlying mechanism which causes the morphing of herds. From these reports, the causation between the herding tendency and the distribution of predation risk in the model can be disclosed.

3.1 Simulation Overview

The internal risk level, μ , is experimented in the range from 0 to 33 by 1700 simulation runs at 17 sampling points (Fig. 3). Regarding the meaning of the μ values, the following three points are notable. When $\mu = 0$, the security score is equal to the number of neighbors, i.e. $s_s = s_n$. Therefore, the inner members are less at risk than the outer ones in a group. When $\mu = 1$, the scores for all $s_n \geq 4$ are equal, and hence the risk distribution within a herd is relatively even. When $\mu = 2$,

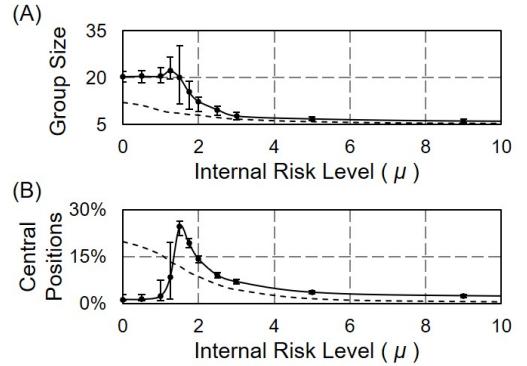


Fig. 3: The size and cohesiveness of emergent herds over internal risk level. In panel A, the measured group size displays the herding tendency disappears after around $\mu = 2$. In panel B, the measured proportion of central positions shows an emergence at around $\mu = 1.5$. Each spot and its associated error bar are the mean value and standard deviation of 100 simulation outputs under the sampled μ value. The dotted lines provide results from a null model where body overlaps are allowed. The trends after $\mu = 10$ are monotonous and omitted. Two components are measured in the same group if there is an agent of one component being the neighbor of an agent in the other one. To avoid noises, agents in isolation or pairs are not counted into groups.

the security score of $s_n = 8$ is equal to that of $s_n = 0$. Risk levels greater than this value cause the interior of a herd more dangerous than any other positions.

The average group size per time step at the final generation is used as the index of the emergent herding tendency under a given μ . As plotted in Fig. 3A, a prey group consistently holds around 21 agents when $\mu \leq 1.25$. The mean group size then decreases steadily before $\mu = 3$, with a qualitatively small deviation among simulation samples, except when $\mu = 1.25$, as the turning point of the emergent patterns. After that, a group is composed of only 6 agents on average in a robust way, which implies the herding tendency disappears.

To quantify the cohesiveness of emergent herds, the occupied grids which $s_n \geq 6$ is treated as the internal area of groups, and the proportion of this area in the final generation is used to represent the degree of cohesion shaped by a certain μ . As plotted in Fig. 3B, when the internal risk level is lower than 1.25, the central positions are rare, at around 1%. However, when the risk level exceeds 1.25, the number of agents inside groups sharply grows and reaches the peak at around 25% of the whole population by $\mu = 1.5$. While the internal risk continues to rise, this percentage declines gradually, and converges at 2%.

The above analyses show that the emergent patterns at different internal risk levels can be organized into three types. As shown in Fig. 4, when $\mu \leq 1.25$, termed as the cases of *mild* internal risk, the collective pattern consists of large and elongated herds without considerable central positions. When $1.25 < \mu \leq 2$, termed as the cases of *moderate* internal risk, large and round herds with a significant internal area emerge. When $\mu > 2$, referred to as the cases of *severe* internal risk, agents evolve to form small heaps locally, and most individuals are at the edge. The typical evolutionary trajectories of the three types are displayed and analyzed in

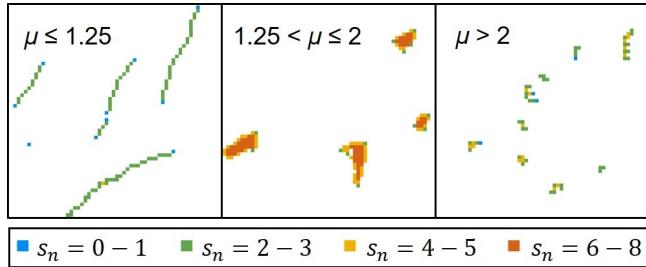


Fig. 4: Three types of the emergent patterns. When the internal risk is mild, not greater than 1.25, agents compose into elongated herds. When the internal risk is moderate, between 1.25 and 2.0, round herds are assembled. When the internal risk level exceeds 2.0, many small heaps appear in the final state. For a clear visualization, only a quarter of the whole lattice is displayed.

the following subsections.

As supplementary information about the significance of involving the constraint on body overlaps, the simulation results from a null model, which discards the overlap-free constraint, are displayed by dotted curves in Fig. 3. It shows without the overlap-free constraint, agents evolve into cohesive herds at $\mu = 0$. These herds on average contain 12 agents each, with a high cohesion degree that nearly 20% of group members are in the internal area. With the increase of the internal risk level, both the group size and the cohesion degree decrease gradually until the herding tendency disappears. This result is the same as traditional selfish herd models [22], where the agents can pass through others' bodies to the inside (outside) of a herd if the outer (inner) positions are more dangerous. However, as explained before, this substantial overlap between individual bodies cannot happen in real animals.

3.2 Adaptation to Mild Internal Risk

When $\mu < 1$, the group center is more secure than the margin. It is the typical selfish-herd environment [8], and hence the evolutionary trajectory repeats that in the original overlap-free selfish herd model [23]. When $1 \leq \mu \leq 1.25$, the internal area of a herd is equally secure as, or even slightly riskier than, the border region. However, the evolutionary trajectory is similar to that of the secure-inside condition. Since the simulation outputs are relatively consistent (Fig. 3), in the following part, a representative simulation under $\mu = 1$ is displayed for a qualitative analysis without the loss of generality.

At the initial generation, each prey agent is randomly assigned a basic NEAT neural network [24] and hence exhibits random moves. After a few generations, prey agents evolve to adopt a more beneficial tactic that they stay with their neighbors in any neighboring situations (s_n), as the *coward strategy* marked in Fig. 5A. This adaptation increases the fitness values compared with the initial random moves (Fig. 5B).

While the evolution continues, agents develop the tactic of a conditional departure that agents give up their current position when the number of neighbors is less than 3 (blue and green dots in Fig. 5A, marked as the *explorer strategy*). This situation happens when agents are in isolation, in pairs or in small heaps. As a result, agents exhibit the herding tendency by seeking and staying in sufficiently large groups, where they are in company with more than 3 neighbors. Through this adaptation, the proportion of central positions (the red curve in Fig. 5C) grows to a significant level, which reflects steady and round herds are assembled. Since the internal risk is mild and the security scores of most central positions are still high, the average fitness value among prey agents is maximized (Fig. 5B).

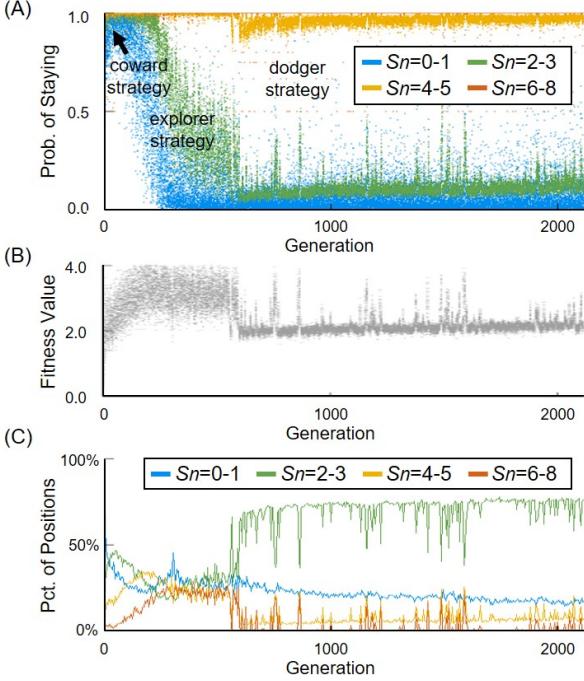


Fig. 5: An evolutionary trajectory given $\mu = 1$. A representative evolutionary trajectory towards the elongated herds is illustrated from many aspects. Panel A plots the probabilities an agent chooses to stay at the same grid, given four kinds of neighboring situations. Panel B plots the fitness values of all agents along generations, by which the mean value and deviation are displayed. Panel C plots the proportions of different neighboring situations in evolution, grouped into four cases. In panel A, only agents which have the option between staying and moving are counted.

When the selection pressure keeps falling on the population, agents must look for more secure positions to surpass others. As the *dodger strategy* marked in Fig. 5A, agents finally evolve to reduce the probability of staying when $s_n \leq 5$, which neighboring situation includes the outer region of a group in two-dimensional space. Therefore, the steady and round herds *dissolve* and *crawl* due to the continuous departure of the agents which are exposed to the border. Consequently, prey agents self-organize into elongated and drifting herds (Fig. 4), and obtain lower fitness values through this transition (Fig. 5B). A further discussion about this morphing dynamic is put in the last subsection.

3.3 Adaptation to Moderate Internal Risk

When $1.25 < \mu \leq 2$, the risk at central positions becomes considerably higher than that at the border. For example, given $\mu = 1.5$, the security scores of $s_n = 4$, $s_n = 6$ and $s_n = 8$ are $s_s = 4$, $s_s = 3$ and $s_s = 2$, respectively. Intuitively, agents should evolve to repulse from the interior and

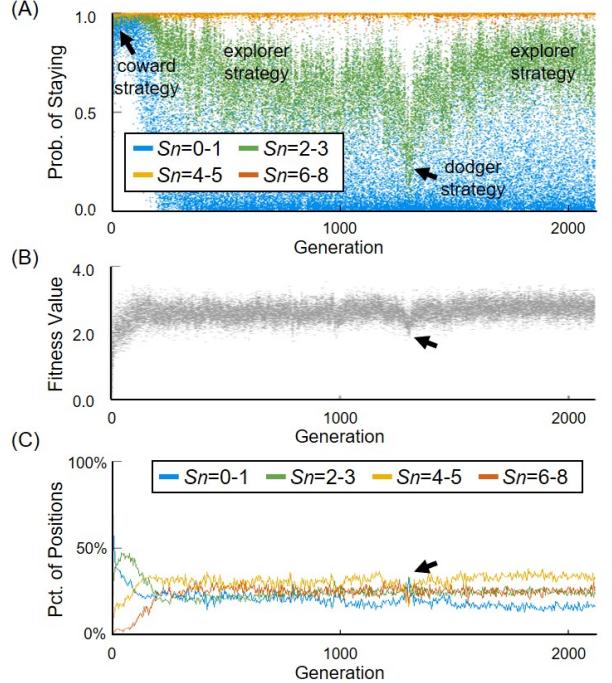


Fig. 6: An evolutionary trajectory given $\mu = 1.5$. A representative evolutionary trajectory towards the steady herds is illustrated by the three panels. The arrows show there is a failed invasion of another strategy. The explanation of these panels is in Fig. 5.

look for the group edge. However, in the overlap-free model, agents inside a group can be trapped by their neighbors from escaping. Hence, from an agent's viewpoint, the tradeoff is whether to stay at the secure outer region bearing the probability of being buried into the less profitable interior. Based on the consistency of simulation outputs (Fig. 3), a representative simulation run given $\mu = 1.5$ is displayed in Fig. 6 to highlight the characteristics of evolutionary trajectories under moderate internal risk.

In the beginning generations, agents evolve to stay with others in any neighboring situation, marked as the *coward strategy* (Fig. 6A). After more generations, the selective force drives prey agents to search for large groups to stay, i.e. giving up the flocking behavior when $s_n \leq 3$, as the *explorer strategy* marked in Fig. 6A. This behavioral tactic then leads to the formation of steady and round herds (Fig. 4), where the proportion of internal area, i.e., occupied grids which $s_n \geq 6$, increases to around 25% (Fig. 6C). So far, the transitions in evolution are qualitatively similar to what observed at a mild internal risk level (Fig. 5).

Nevertheless, under moderate internal risk, this *explorer strategy* remains stable with generations. As an example of the *explorer strategy*'s stability in evolution, an invasion of the *dodger strategy* can be observed at around the 1300th

generation in Fig. 6. In the period of this invasion, agents exhibit a higher frequency of leaving under the condition of $s_n = 2\text{--}3$ (the arrow in Fig. 6A) and lead to lower fitness values (the arrow in Fig. 6B) due to the transformation of positional distributions (the arrow in Fig. 6C). However, the perturbed state cannot maintain stable at a moderate internal risk level. Agents quickly adopt the *explorer strategy* again after few generations (Fig. 6A).

The deviation of fitness values in the prey population (Fig. 6B) is higher than that under mild internal risk (Fig. 5B). It is because an inner individual of a herd hardly has opportunities to improve its fitness in the overlap-free environment. This phenomenon causes the measured fitness values relatively uneven in a stable state.

3.4 Adaptation to Severe Internal Risk

When the internal risk level exceeds the threshold $\mu = 2$, the third type of adaptation occurs in the evolutionary game. Under this condition, any other positions are much more beneficial than the group center. For example, given $\mu = 5$, the security score of $s_n = 8$ is -12, which means 12 units lower than the neighboring situation $s_n = 0$, i.e. being in isolation. Considering the highest security score is 4 by $s_n = 4$, this difference is a severe penalty on individual fitness value.

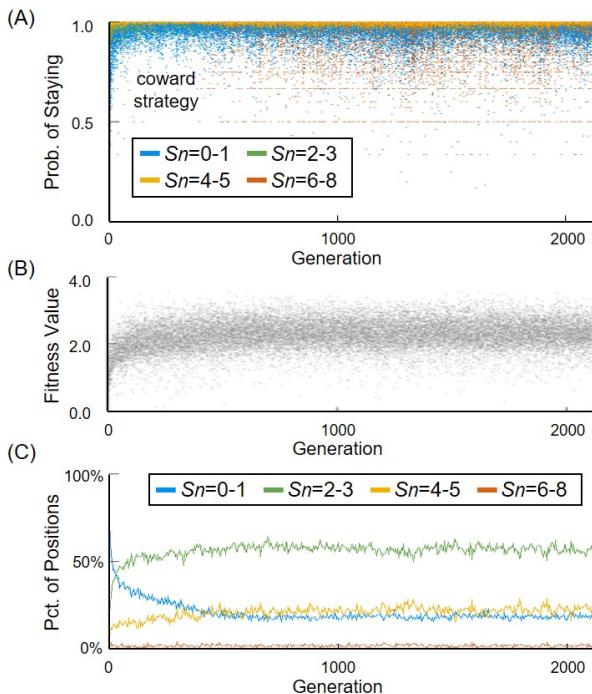


Fig. 7: An evolutionary trajectory given $\mu = 5$. A representative evolutionary trajectory towards small heaps is illustrated by the three panels. It is observed only those agents in the group center leave their positions so the herding tendency cannot emerge. The explanation of these panels is in Fig. 5.

Despite whether this risk distribution can happen in the real world, the simulations demonstrate the behavioral evolution still consistently reaches a stable state (Fig. 3). Like the trajectories in the previous two types, the replacement of random walks by the *coward strategy* (Fig. 7A), i.e., staying with few neighbors locally, happens quickly. Since agents form small heaps rather than being solitary, the average fitness value slightly increases (Fig. 7B).

After hundreds of generations, a variation of the *coward strategy* evolves (Fig. 7A). Apart from the feature of grouping locally, agents exhibit an additional repulsive behavior when $s_n \geq 6$ (red dots in Fig. 7A), which reflects the tactic of escaping from the group interior. Therefore, the shapes of small heaps become irregular without an internal area (Fig. 4). This emergent pattern shows the herding tendency, as the behavior to form large herds, cannot evolve when the risk within a group becomes seriously high.

Although the adaptation is demonstrated stable, the distribution of agents' fitness values is not clearly converged (Fig. 7B). It is due to the different sizes as well as different neighboring situations between the small heaps. When the internal risk level is not severe, it was demonstrated that the deviation of fitness values should be reduced from the development of the *explorer strategy*. However, this transition is unfavored in evolution given a severe internal risk level.

3.5 The Emergence of Group Morphing

The simulations have demonstrated that when $\mu \leq 1.25$, prey agents evolve to leave the border and lead to the emergence of collective movements. When the group center is safer than the edge, the benefit to this adaptation was considered due to transforming the central positions into the insecure border [23]. However, in the range $1 \leq \mu \leq 1.25$, the internal area is equally or more dangerous than the outer region, so the fitness value of an exposed inner agent can even be improved. Therefore, the underlying mechanism is different in this case.

According to the snapshots of simulations, it is found that the morphing dynamic in the insecure-inside situation is driven by the competition between border agents. When $\mu \geq 1$, being with 3–5 neighbors receive the highest fitness value, which happens in the border area. However, on the two-dimensional lattice, the border of a herd can be further separated into sides and corners. An agent at the side position has around 4–5 neighbors and gains a better fitness value than being at the corner with only 2–3 neighbors.

Under this situation, leaving the corner to transform the nearby agents' positions from the side to the corner can be beneficial in evolution. As a chain effect, when agents leave the corner sequentially, the side agents are exposed to the corner, the inner agents are exposed to the margin, and the collective motion emerges (Fig. 8).

However, with the growth of μ , this dynamic brings in-

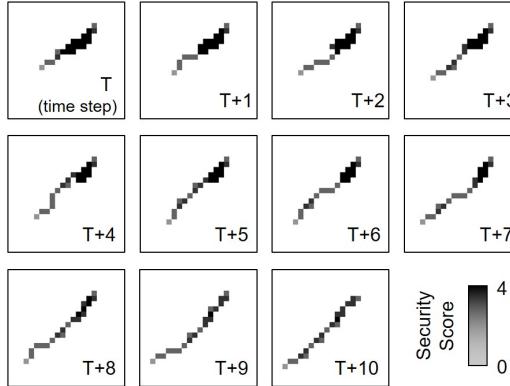


Fig. 8: The morphing of a herd under mild internal risk. When the internal risk is mild, the corner agents, which receive lower security scores than the side agents, tend to leave the herd. The departure brings other agents to the corner. As a chain effect, the herd exhibits a collective movement. These snapshots are taken from a simulation given $\mu = 1$.

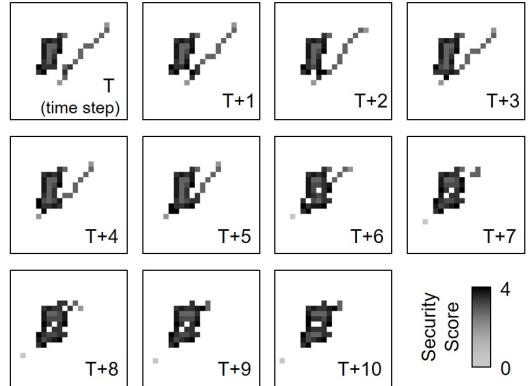


Fig. 9: The steadiness of a herd under moderate internal risk. When the internal risk is large enough, leaving the border can profit the inner agents. Hence, the corner and side agents evolve to stay at the same position, and lead to the steadiness of a herd. These snapshots are taken from a simulation given $\mu = 1.5$.

creasing advantage to the inner agents. Hence, the benefit of leaving a corner position from harming the side agents will final insufficient to cover the loss from profiting the inner agents. As demonstrated by the simulations, when $\mu > 1.25$, the difference from the insecure center to the secure border drives the corner agents to stay, rather than a departure. Eventually, steady herds emerge in a stable state under moderate internal risk (Fig. 9).

4. Discussion and Conclusion

The experiments have demonstrated how the predation risk inside a herd can affect the behavioral evolution of social prey. When the group center is safer or slightly riskier than the margin, the *dodger strategy*, as the repulsion from the border positions, is demonstrated stable in evolution and leads to the formation of elongated and moving herds. In contrast, when the internal risk level rises above a threshold, the strategy to find and stay in a large group is consistently exhibited by prey agents, which leads to cohesive and steady herds. However, when the internal risk of a group is extremely high, the herding tendency cannot evolve. Prey agents flock locally and form numerous small and irregular heaps.

These findings reveal that when the existence of physical bodies is further considered, the precondition of the original selfish herd scenario, as dangerous borders versus riskless centers, may be unnecessary to the evolution of animals' herding tendency. The simulations given $\mu \leq 2$ demonstrated the selfish herd can evolve stably even if the risk inside a group is considerably higher than the border region. Moreover, the experiments showed that the herding pattern is shaped by the internal risk level. When the risk biases to the outer region,

the movement of a prey herd is triggered by the departure of those agents which are at the most dangerous border positions, e.g. the corners. When the risk biases to the central region, prey herds stop moving and convert to a round shape, which contains a significant internal area.

In this model, the internal risk level is fixed during an evolutionary trajectory, and hence the property of emergent herds maintains the same in a stable state. Given a species of gregarious prey capable to recognize the type of predators, it can be reasonably inferred that evolution should shape prey to assemble different types of selfish herds in front of different predators. A possible example is various kinds of shoal fish in the open ocean. These social animals usually practice collective motion in predation, where fish at the frontal positions bear the highest risk [25], as the corner positions in the present model. Besides, they usually transform their herd into a milling and round aggregate (the *bait ball*) when facing giant predators like dolphins and whales [16, 17]. These predators lead to high risk inside fish schools. Although the dynamics were, separately, viewed as prey's anti-predator reactions and predators' countermeasures, the accordant simulation outcomes may suggest that intraspecific competition could also enhance these formations in evolution.

Lastly, it is clear that the flocking behavior of social prey can be driven by several factors, e.g. anti-predator functions and cost-benefit tradeoffs [5, 6]. However, the present work highlights that when the impact of animal bodies is considered, the selfish herd theory may explain more predator-prey systems, and can reasonably address the pending question about why animals flock in the insecure-inside condition. These findings may reveal a novel perspective for interpreting collective animal behaviors in nature.

References

- [1] A. E. Magurran and T. J. Pitcher, "Provenance, shoal size and the sociobiology of predator-evasion behaviour in minnow shoals," *Proc. R. Soc. Lond. B*, vol. 229, no. 1257, pp. 439–465, 1987.
- [2] C. Carere, S. Montanino, F. Moreschini, F. Zoratto, F. Chiarotti, D. Santucci, and E. Alleva, "Aerial flocking patterns of wintering starlings, *Sturnus vulgaris*, under different predation risk," *Animal Behaviour*, vol. 77, no. 1, pp. 101–107, 2009.
- [3] A. J. King, A. M. Wilson, S. D. Wilshin, J. Lowe, H. Haddadi, S. Hailes, and A. J. Morton, "Selfish-herd behaviour of sheep under threat," *Current Biology*, vol. 22, no. 14, pp. R561–R562, 2012.
- [4] W. Romey, A. Walston, and P. Watt, "Do 3-d predators attack the margins of 2-d selfish herds?," *Behavioral Ecology*, vol. 19, no. 1, pp. 74–78, 2007.
- [5] C. C. Ioannou, V. Guttal, and I. D. Couzin, "Predatory fish select for coordinated collective motion in virtual prey," *Science*, p. 1218919, 2012.
- [6] G. Richardson, P. Dickinson, O. H. Burman, and T. W. Pike, "Unpredictable movement as an anti-predator strategy," *Proc. R. Soc. B*, vol. 285, no. 1885, p. 20181112, 2018.
- [7] J. Krause and G. D. Ruxton, *Living in groups*. Oxford University Press, 2002.
- [8] W. D. Hamilton, "Geometry for the selfish herd," *Journal of theoretical Biology*, vol. 31, no. 2, pp. 295–311, 1971.
- [9] A. J. Wood and G. J. Ackland, "Evolving the selfish herd: emergence of distinct aggregating strategies in an individual-based model," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 274, no. 1618, pp. 1637–1642, 2007.
- [10] R. S. Olson, A. Hintze, F. C. Dyer, D. B. Knoester, and C. Adami, "Predator confusion is sufficient to evolve swarming behaviour," *Journal of The Royal Society Interface*, vol. 10, no. 85, p. 20130305, 2013.
- [11] I. Karsai, E. Montano, and T. Schmickl, "Bottom-up ecology: an agent-based model on the interactions between competition and predation," *Letters in Biomathematics*, vol. 3, no. 1, pp. 161–180, 2016.
- [12] S. V. Viscido and D. S. Wethey, "Quantitative analysis of fiddler crab flock movement: evidence for selfish herd behaviour," *Animal behaviour*, vol. 63, no. 4, pp. 735–741, 2002.
- [13] A. De Vos and M. J. O'Riain, "Sharks shape the geometry of a selfish seal herd: experimental evidence from seal decoys," *Biology Letters*, vol. 6, no. 1, pp. 48–50, 2010.
- [14] P. Domenici, A. Wilson, R. Kurvers, S. Marras, J. E. Herbert-Read, J. F. Steffensen, S. Krause, P. Viblanc, P. Couillaud, and J. Krause, "How sailfish use their bills to capture schooling prey," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 281, no. 1784, p. 20140444, 2014.
- [15] S. P. Oliver, J. R. Turner, K. Gann, M. Silvosa, and T. D. Jackson, "Thresher sharks use tail-slaps as a hunting strategy," *PLoS One*, vol. 8, no. 7, p. e67380, 2013.
- [16] T. Similä and F. Ugarte, "Surface and underwater observations of cooperatively feeding killer whales in northern norway," *Canadian Journal of Zoology*, vol. 71, no. 8, pp. 1494–1499, 1993.
- [17] R. L. Vaughn-Hirshorn, E. Muzi, J. L. Richardson, G. J. Fox, L. N. Hansen, A. M. Salley, K. M. Dudzinski, and B. Würsig, "Dolphin underwater bait-balling behaviors in relation to group and prey ball sizes," *Behavioural processes*, vol. 98, pp. 1–8, 2013.
- [18] D. Wiley, C. Ware, A. Bocconcelli, D. Cholewiak, A. Friedlaender, M. Thompson, and M. Weinrich, "Underwater components of humpback whale bubble-net feeding behaviour," *Behaviour*, vol. 148, no. 5, pp. 575–602, 2011.
- [19] Y. Inada and K. Kawachi, "Order and flexibility in the motion of fish schools," *Journal of theoretical Biology*, vol. 214, no. 3, pp. 371–387, 2002.
- [20] M. Zheng, Y. Kashimori, O. Hoshino, K. Fujita, and T. Kambara, "Behavior pattern (innate action) of individuals in fish schools generating efficient collective evasion from predation," *Journal of theoretical biology*, vol. 235, no. 2, pp. 153–167, 2005.
- [21] S.-H. Lee, H. Pak, and T.-S. Chon, "Dynamics of prey-flock escaping behavior in response to predator's attack," *Journal of theoretical biology*, vol. 240, no. 2, pp. 250–259, 2006.
- [22] R. S. Olson, D. B. Knoester, and C. Adami, "Evolution of swarming behavior is shaped by how predators attack," *Artificial life*, vol. 22, no. 3, pp. 299–318, 2016.
- [23] W.-C. Yang, "When the selfish herd is too crowded to enter," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2247–2254, IEEE, Nov 2017.
- [24] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.
- [25] D. Bumann, J. Krause, and D. Rubenstein, "Mortality risk of spatial positions in animal groups: the danger of being in the front," *Behaviour*, vol. 134, no. 13, pp. 1063–1076, 1997.

Agent-Based Modeling and Complex Network Analysis for Clarifying Changes in the Japanese Textile and Apparel Industry B2B Networks

Yusaku Ogai [†]

Yoshiyuki Matsumura [†]

Yusuke Hoshino [‡]

Masanori Goka [§]

Toshiyuki Yasuda ^{*}

Kazuhiro Ohkura ^{*}

[†] Shinshu University, Japan

[‡] Musashino University, Japan

[§] Fukuyama University, Japan

^{*} Toyama University, Japan

^{*} Hiroshima University, Japan

Abstract: In this study, we estimated the changes in the business-to-business (B2B) networks in the Japanese textile and apparel industry using data from 2000 companies from 2011 to 2016. We used complex network analysis to evaluate the properties of the B2B networks. The factor of innovation in information communication technology (ICT) and logistics technology for generating B2B networks was examined in an agent-based model. In complex network analysis, power-law properties and centralization to hub companies were found over the years. Moreover, centralization of the B2B networks due to the innovation in ICT and logistics technology was demonstrated using simulation experiments. Therefore, changes in the networks were explained as centralization to hub companies caused by innovation in ICT and logistics technology.

Keywords: Business to Business networks, innovation of ICT and logistics technology, complex systems, scale-free networks, agent-based modeling.

1. Introduction

1.1 Innovation in information communication technology (ICT) and optimized logistics technology in B2B networks

Technological innovation has given rise to changes in our social features and lifestyles. For example, innovative technologies such as information technology, communication technology, and optimized logistics technology can be integrated to optimize social infrastructures. However, integrating these technologies is difficult for isolated companies, as the former have recently evolved and become quite complicated. Reaching a consensus between related supplier companies is necessary to integrate these innovative technologies. For this purpose, a Japanese ICT company, Softbank, has proposed a swarm strategy in business to business (B2B) networks [1]. This management strategy is proposed on the basis of the systems science. The concept is related to the process of evolution as an optimization to environmental fluctuations [2]. The new company was established as a result of co-funding within the mobility company and the ICT company [1]. Integration of the various types of technologies can be achieved by the swarm of companies.

Network analysis and simulation experiments can be used for the objective management of B2B networks to produce innovative strategies such as the swarm-strategy. For example, complex network analysis is a methodology for estimating the B2B networks state [3]. The Barabási-Albert model is the best known model demonstrating the relationship between the power-law of the networks and the adaptive behaviors of nodes [3]. Complex networks with power-law properties are defined as scale-free networks. Scale-free properties are discussed not only in the Barabási-Albert model but also in the context of real social network data [4]. Small-world networks are found in the analysis results of various social networks [5]. In addition, a simulation model based on an equation is known as a method for evaluating the dynamics of the social state. In this method, a top-down approach such as a stochastic model using Brownian-motion is applied to demonstrating fluctuations of economic indexes [6]. As a bottom-up approach, agent-based modeling is proposed for predicting social state [7]. A segregation model developed by an economist describes emergence of homogeneous communities from heterogeneous communities as an emergent phenomenon [8]. These methods and ideas which are used by an economists are applied in urban planning and opti-

mization. It is possible to upgrade social infrastructures on the basis of these methods by the geometrical relationship between humans and their environments.

Therefore, evaluating the state of the B2B networks is possible by applying materials and methods effectively. In terms of materials, data on typical industries are needed. As discussed in greater detail in Section of 2.1, we focused on the data on B2B networks in the Japanese textile and apparel industry. In terms of methods, B2B networks composed of companies are regarded as complex networks, and agent-based modeling is applied to simulate processes of emergence the B2B networks. In addition, innovation in ICT and logistics technology is introduced in an agent-based model.

1.2 Network analysis as a deductive approach and agent-based modeling as an inductive approach

We propose two research questions to quantify changes in the B2B networks in the Japanese textile and apparel industry. This paper is composed of two sections based on these questions.

The first question investigates the statistical properties of the B2B networks and is based on a deductive approach. The first question is based on the deductive approach. We have quantified these statistical properties. The research hypothesis is not designed to answer this first question.

The second research question examines the origins of changes in the B2B networks and is based on an inductive approach. The second question is based on the inductive approach. Taking into account the historical backgrounds the industry, in which the innovation in ICT and logistics technology has influenced B2B networks, an agent-based model for testing an effect of the innovation is proposed in this study. To answer the second question, we have generated networks using an agent-based model. We discuss changes in the B2B networks based on the results of the two approaches.

2. Japanese textile and apparel industry

2.1 Historical background related to ICT and logistics technology

Technological innovation such as ICT and logistics technology are strongly related to the B2B networks in the Japanese textile and apparel industry. There are traditional industries, and mass production industries in this industry. Because of the industrial structure, the characteristics of typical industries can be estimated considering their historical backgrounds. For instance, this industry originates from manufacturing groups in local areas in the 18th century [9]. Buyers in specific local communities have expanded hub companies through international trading [10]. In addition, old fashioned

groups remains as a traditional manufacturing in Japan [10]. However, new groups that are not dependent on the domestic industry are also growing in this industry [11]. On account of these complex structures, the bullwhip effect in B2B networks is discussed in the context of multi-agent systems[12]. The bull-whip effect describes fluctuations in the number of products caused by sharing incomplete information in ordering upstream to downstream in the industry. A supply chain management system for sharing information in manufacturing and retailing orders has been introduced to solve the bullwhip effect [13][14]. Moreover, a Specialty store retailer of a Private label Apparel (SPA) method, in which the processes of manufacturing and retailing apparel products are self-contained, has become a main management style in this industry [15].

2.2 The datasets of the B2B networks

The company name and supplier company name with a summary of men's apparel, women's apparel, total apparel, children's apparel, inner wear, jeans, and designers are entered into the Fashion and Bland guide offered by textile and apparel industry newspaper company in Japan [16]. We applied these data to the analysis of the B2B networks. We have generated complex networks on the basis of the data on the company and supplier name in all departments for the analysis. To evaluate the change over time in the B2B networks, we have used datasets from 2011/12, 2013/14, and 2015/16.

3. Network analysis and agent-based modeling

3.1 Network analysis for the first research question

3.1.1 Concepts of indexes in network analysis

We used four indexes to quantify statistical properties of the whole network : edge number (E), average degree ($\langle k \rangle$), network density (D), and network path length (L). In this section, the basic concept of the complex network analysis in this study is discussed. A network is composed of nodes and edges. In this paper, a node is defined as a company name and a supplier name. An edge is defined as trading between the two companies. We visualized network structures on the basis of the data of nodes and edges. A node is v_i , an edge is E_{ij} , and the total number of nodes is N . The total number of edges is E , and the degree, k , represents the number of business partners. The average number of degrees is $\langle k \rangle$. The network density represents the density of the edges in a B2B networks. The total number of edges in the networks is divided by the total number of edges in a complete network (Eq.1). This index is written as D . The network path length is

the index describing the intervals between companies within the B2B networks (Eq.2), and this index is written as L . The number of total intervals $d(v_i, v_j)$ between nodes is divided by the total number of edges within a complete network.

$$D = \frac{2 \cdot \Sigma E_i}{N(N - 1)} \quad (1)$$

$$L = \frac{2 \cdot \Sigma d(v_i, v_j)}{N(N - 1)} \quad (2)$$

3.1.2 Analysis of degree distribution

The degree distribution is calculated using the frequency of degree k within the whole network [3]. Figure 1 presents a conceptual illustration of the processes in this analysis. In degree distribution, the horizontal axis is defined as a size of degree k . The vertical axis is defined as a frequency of degree $p(k)$. Using this analysis, it is possible to quantify the statistical properties of the number of degrees within the whole network. We also converted the degree distribution to log-log plotting. In this study, a linear model with a cutoff in log-log plotting was hypothesized as a fitting model. This model is often applied to evaluate scale-free properties. In addition, scale-free properties are identified in various types of social networks [4]. Scale-free networks are composed of a lot of nodes with few edges and a few nodes with very large edges.

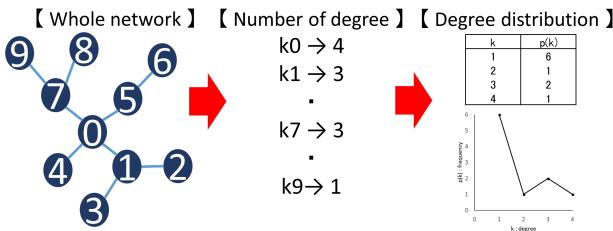


Fig. 1: Conceptual figure of the analysis based on degree distribution

3.2 Methods for the second research question

3.2.1 Concepts of the agent-based model in this study

Agent-based modeling is a method used for computation in the social sciences. In this study, a simple model is proposed, consisting of scoring interactions to characterize agents. This method assumes agents in a field [14]. Here, an agent means a node in the networks. Interactions between agents are equal to edges in the networks. Agents are located in a one-dimensional field using their named number. The distances between agents are calculated as Manhattan-distances $r_{ij} = |i - j|$ based on agent i and selected agent j . Selected agent j is determined using the selection rules presented in

Section of 3.2.2. Interactions scored as characterizations of agents represent trading between companies, as described in Section of 3.2.3. An agent i is updated as $i = i + 1$ when an agent i is updated in the interactions. The next time step is updated as $t = t + 1$ on the completion of updating the state of an agent i .

3.2.2 Selection rules of the agent-based model

The qualitative properties of the B2B networks in the industry are similar to a river stream network[17][18]. In this study, a hub company, a middle hub company and another company were assumed as the companies type. The rules for selecting an agent using a score were constructed as a stochastic model for generating a hub agent(Eq.3), a middle hub agent (Eq.4), and another agent (Eq.5). These rules (Eqs.3-5) are selected in an agent at random. The rules are similar to the interaction rules in a boid model[19]. Specifically, selecting agents with the highest scores is similar to the rule of cohesion (Eq.3). Selecting agents with a higher scores than an agent is similar to the rule of alignment (Eq.4). Finally, selecting agents at random is similar to the rule of separation (Eq.5).

$$j_\alpha = j_{U_{max}(t)} \quad (3)$$

$$j_\beta = j_{U_{random(U_i < U_j)}(t)} \quad (4)$$

$$j_\gamma = j_{random(N)} \quad (5)$$

3.2.3 Interaction rules of the agent-based model

The interaction rules of this model consist of a trading equation using a unit vector of the score (Eq.6). The unit vector of the score between an agent i and selected agent j represents a unit of score in the exchanging. When an agent i has a higher score than selected agent j , a unit vector is added to the score of $U_j(t)$ in selected agent j . When a unit vector is added to the score of $U_j(t)$ in selected agent j , a unit vector is subtracted from the score of $U_i(t)$ in an agent i . Similarly, when a unit vector is added to the score of $U_i(t)$ in an agent i , a unit vector is subtracted from the score of $U_j(t)$ in selected agent j . In addition, a gravitational characterization $1/r_{ij}$ which is introduced to the score of $U_i(t)$ based on the Manhattan-distance between agents is composed in this model (Eq.7). The model, which is similar to a gravitational potential equation, is often analyzed to evaluate the effects of geographical positioning in economic analysis [20]. We applied this method to an agent-based model for generating networks. In addition, the distance weight coefficient, w is multiplied by r_{ij} in this model, thus estimating the effect of the innovation in ICT and logistics technology.

$$U_i(t) = U_i(t - 1) + \frac{U_i(t) - U_j(t)}{|U_i(t) - U_j(t)|} \quad (6)$$

$$U_i(t+1) = \frac{U_i(t)}{(w \cdot r_{ij})} \quad (7)$$

3.2.4 Experimental conditions of the agent-based simulation

The number of agents was defined as the number of companies and was set to $N = 1500$. The initial scores of the agents were set to $U_i(0) = [0, 100], i \in N$. The distance weight coefficient, w , was changed at the experimental conditions. The experimental conditions were set to $w = 1, 0.8, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01$. The distance weight coefficient, w , as explain variable was tested based on these each condition. These simulation experiments were performed 10times at each the experimental conditions.

The distance weight coefficient is intended to decrease the cost of transportation with changes to the innovation in ICT and logistics technology. In the simulation experiments, factors as innovation in ICT and logistics technology were introduced, taking into account historical the background of the Japanese textile and apparel industry. In this study, the networks generated by agents were applied to estimate a result of the simulation experiments. The networks were captured when the sum of scores of all agents reached a maximum value under experimental conditions. The networks were visualized as diagrams. In addition, the statistical properties of the networks were calculated on the basis of the degree distribution discussed in the previous section. A linear fitting model was applied to the degree distribution and was determined with a cutoff using the best coefficient of determination. The results of the simulation experiments were applied to the statistical analysis.

4. Results

4.1 Network analysis results

Figure 2 represents diagrams of the network structures of the Japanese textile and apparel industry over the years. The whole networks are visualized in a circular layout on the basis of the types of companies, and the indexes of these networks are represented in Table 1. In this table, the edge number decreased from $E = 2309$ to $E = 1572$ over time. In addition, the average degree decreased from $\langle k \rangle = 2.65$ to $\langle k \rangle = 2.47$. Moreover, the network density increased over time, representing the density of trading. These networks were thus compacted from $D = 0.00152$ to $D = 0.00195$. The network path length increased from $L = 5.66$ to $L = 5.72$ from 2011/12 to 2013/14. However, this parameter decreased from $L = 5.72$ to $L = 5.55$ from 2013/14 to 2015/16.

Figures 4, 5, and 6 illustrate the degree distribution of the whole networks in 2011/12, 2013/14, and 2015/16, respectively. Table 2 presents a linear fitting model to the

degree distribution of the whole networks. These results demonstrate that scale-free networks are detected as common over the years. The statistical properties were evaluated as $p(k) \sim C \cdot k^a$. The significance levels of the slope coefficients, a , were $p < 0.05$. Specifically, the slope coefficients in this fitting model decreased as follows : $a = -2.26$ in 2011/12, $a = -2.33$ in 2013/14, $a = -2.51$ in 2015/16.

4.2 Networks generated by agents with changes in distance weight

Figure 2 presents the networks generated by agents with changes in the distance weight coefficient. These results demonstrate that the number of hubs decreased with changes to the experimental conditions.

Figure 7 shows the degree distribution with changes to the distance weight coefficient. Table 3 presents the results of regression analysis on the results of agent-based simulation experiments. Here, the statistical properties of the results in the simulation experiments were evaluated as $p(k) \sim C \cdot k^a$. All the significance levels of the slope coefficients a were $p < 0.05$. These results show that the tendency of the slope coefficients in this fitting model for the data decreased continuously with changes in the coefficient w : specifically $a = -2.28$ then $w = 1$, $a = -2.56$ then $w = 0.3$, and $a = -3.78$ then $w = 0.01$.

Figure 8 presents the slope coefficients a in the results of real data and the results of simulation experiments. Decreasing the coefficient of w implied a decreasing a cost of transportation caused by the innovation. Therefore, simulation experiments are included a concept of change over time. The results show that a common tendency of changes in the networks with change over time from the slope coefficients for the two approaches. In simulation results, slope coefficient in the fitting models of the degree distribution decreased with changes in the coefficient. Similarly, result of real data analysis also showed a decrease in the slope coefficients in the fitting models of the degree distribution over time.

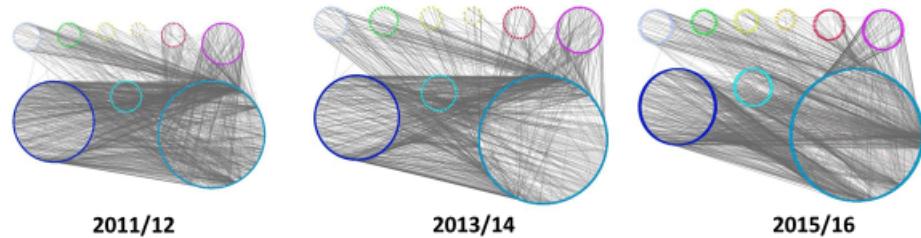


Fig. 2: Changes in networks from 2011/12 to 2015/16.

These networks were visualized by Cytoscape on the basis of a circular-layout with the types of companies.

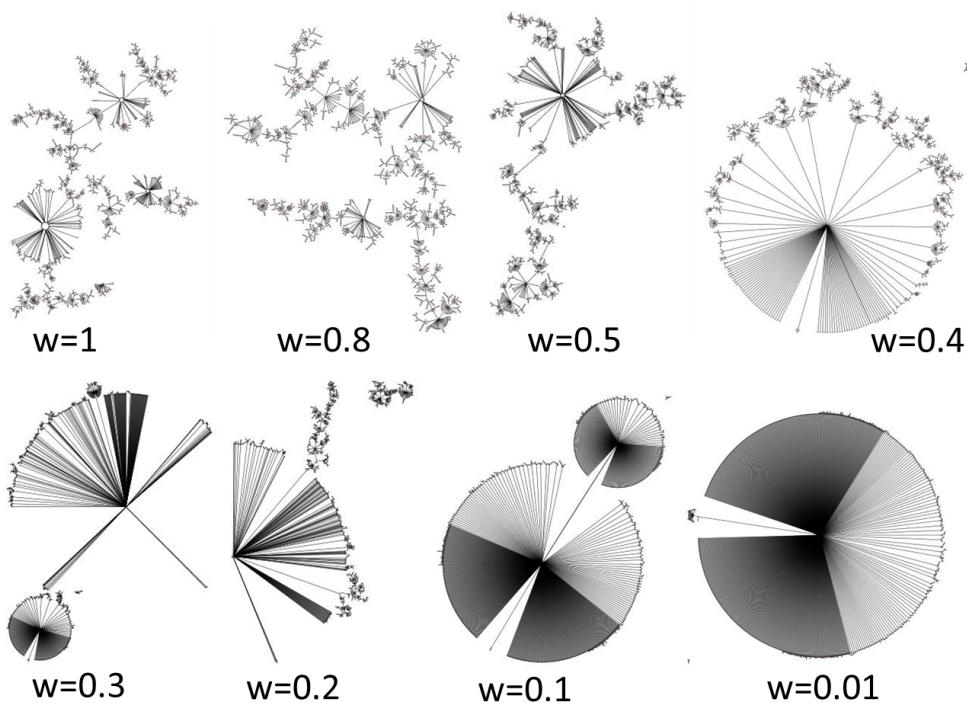


Fig. 3: Changes in networks with the distance weight coefficient as the innovation in ICT and logistics technology
The networks were captured when the sum of the points of all agents has the maximum $t = t_{U_{max}}$. These results are from one trial of the simulation experiments.

Table. 1: Results of complex network analysis based on the four indexes of real data

	Edge number [company]	Average degree [company]	Density [normalized]	Length [company]
2011/12 Whole	2309	2.65	0.00153	5.66
2013/14 Whole	1726	2.48	0.00178	5.72
2015/16 Whole	1572	2.47	0.00195	5.55

Table. 2: Linear fitting model of regression analysis of the degree distribution of real data

Years	C (segment)	a (slope)	R ²	t value	p value
2011/12 ($k \leq 10$)	3.21	-2.26	0.923	-9.81	9.76E-06
2013/14 ($k \leq 11$)	3.13	-2.33	0.926	-10.6	2.12E-06
2015/16 ($k \leq 13$)	3.16	-2.51	0.930	-12.1	1.09E-07

Table. 3: Linear fitting model of the regression analysis of the degree distribution of the simulation experiments

w(10times average)	C (segment)	a (slope)	R ²	t value	p value
1	3.05	-2.28	0.940	-15.3	1.50E-10
0.8	3.05	-2.32	0.941	-16.5	6.85E-12
0.5	2.95	-2.15	0.928	-13.9	5.40E-10
0.4	2.98	-2.18	0.945	-15.0	1.41E-09
0.3	3.13	-2.56	0.938	-10.3	1.73E-05
0.2	3.17	-2.74	0.886	-4.83	1.69E-02
0.1	3.27	-3.20	0.856	-4.23	2.42E-02
0.01	3.36	-3.78	0.835	-3.89	3.01E-02

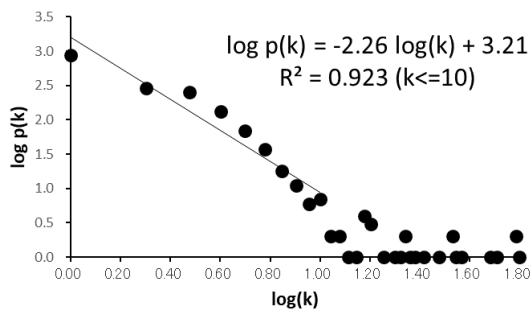


Fig. 4: Degree distribution in 2011/12

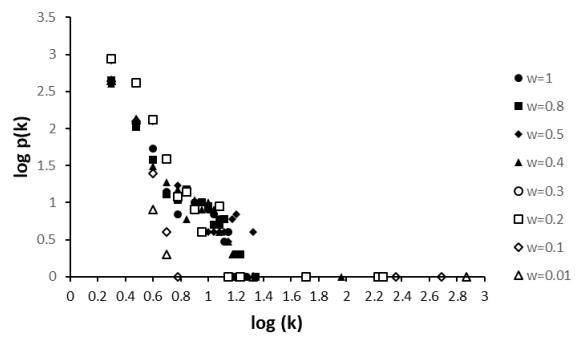


Fig. 7: Degree distribution in the simulation experiments

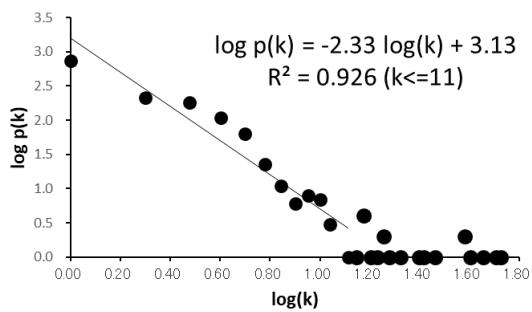


Fig. 5: Degree distribution in 2013/14

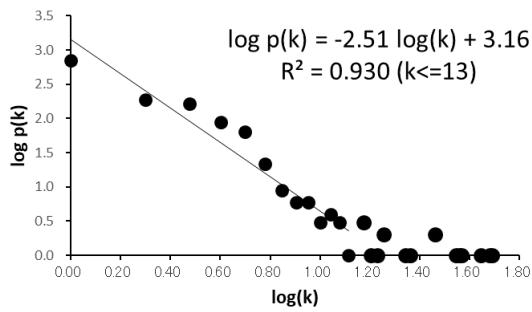


Fig. 6: Degree distribution in 2015/16

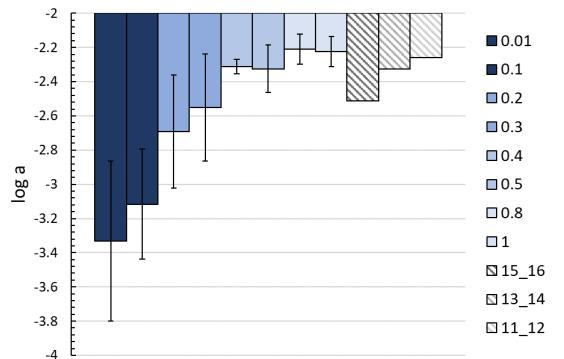


Fig. 8: Slope coefficients in the two approaches
Error bar represents the standard deviation (10 times) in the simulation experiment.

5. Discussion

First, we discuss the statistical properties of the Japanese textile and apparel industry with respect to the first question. The edge number, E , represents the amount of trading in the networks. This index decreased, and the networks were compacted over time (Table 1). The average degree, $\langle k \rangle$ represents the average number of business partners in the whole network. We interpreted the result of this index as a decrease in the amount of trading in these network (Table 1). The network density D represents the density of trading in these networks (Table 1). The network path length, L , represents the intervals between companies, however, this index was not useful for interpreting the changes in the networks. On comparing older datasets with newer datasets in this index, the results showed greater centralization in 2015/16 than in 2011/12 (Table 1). Moreover, the degree distribution indicated that the networks were centralized to fewer hub companies over time. In fact, the tendency of the slope coefficients a was shown to decrease over time (Figures 3-5, Table 2). When the slope coefficients decreased, the network structures were centralized in fewer hub companies. As a result of the complex network analysis, we quantified that B2B networks in the Japanese textile and apparel industry were centralized in a hub by decreasing the amount of trading throughout the entire network. Moreover, reducing the amount of trading within the networks was demonstrated quantitatively on the basis of the complex network analysis.

Second, we discuss the results of simulation experiments considering changes in the distance weight coefficient, w , which represents innovation in ICT and logistics technology. Networks generated by agents were centralized with changes in the distance weight coefficient (Figure 6). Moreover, the slope coefficient a of degree distribution analysis decreased with changes to the distance weight coefficient (Figure 7; Table 3). These results demonstrate that the influence of a hub agent increases with changes in the distance weight coefficient. Therefore, the results of the simulation experiments show that innovation in ICT and logistics technology affects the changes in the networks generated by agents.

Finally, we discuss the results of comparing two approaches based on the results of the slope coefficient (Figure 8). Implementing the factor of technological innovation is difficult because there are not rigid cause and effect in real social data. Simulation conditions were set considering the factor of innovation with change over time as the distance weight coefficient. In addition, the Japanese textile and apparel industry is strongly related to ICT and logistics technology, considering its historical background. Here we can complement the absence of factors in the two approaches. In addition, the statistical properties of the results of the two approaches indicate a common tendency of centralization to hub companies and agents in these networks. Therefore, there is a

common tendency between the results of the real data analysis and simulation experiments. Changes in the B2B networks in the Japanese textile and apparel industry are explained as the centralization to hub companies with innovation in ICT and logistics technology. We suggest that this innovation also affects the construction of hub company in the B2B networks.

6. Conclusion

In generally, the changes in B2B networks are strongly related to the innovation in ICT and logistics technology. Changes in B2B networks in the Japanese textile and apparel industry were quantified as a centralization to hub companies with over time. The first research question in this study was answered in that the statistical properties of the networks were estimated as a power-law. Innovation in ICT and logistics technology also affected changes in the networks generated by agents in the simulation experiments. The second research question in this study was addressed in that the changes in the networks caused by innovation in ICT and logistics technology were demonstrated by an agent-based model. On the basis of the results from the two approaches, network centralization was shown to be caused by innovation in ICT and logistics technology in the Japanese textile and apparel industry. In addition, we described changes in the B2B networks quantitatively on the basis of complex network analysis and agent-based modeling.

7. Acknowledgments

This research was supported by JSPS 17K00302, 24220012, and 23240100.

References

- [1] CNN BUSINESS, SoftBank and Toyota want driverless cars to change the world, <https://money.cnn.com/2018/10/04/technology/softbank-toyota-masa-son/index.html>.
- [2] Manuele Brambilla, Eliseo Ferrante, Manuro Birattari, Marco Dorigo, "Swarm robotics: a review from the swarm engineering perspective", *Swarm Intelligence*, Vol. 7, No. 1, pp. 1–41, 2013.
- [3] Re'ka Albert, Albert-László Barabási, "Statistical mechanics of complex networks", *REVIEWS OF MODERN PHYSICS*, Vol. 74, No. 1, pp. 47–97, 2002.
- [4] Albert-László Barabási, Eric Bonabeau, "Scale-Free Networks", *Scientific American*, Vol. 288, No. 5, pp. 60–69, 2003.
- [5] Duncan J. Watts, Steven H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, Vol. 393, No. 4, pp. 440–442, 1998.

- [6] Alexandru Mandes, Peter Winler, "Complexity and model comparison in agent based modeling of financial markets", *J Econ Interact Coord*, Vol.12, No.10, pp.469-506, 2017.
- [7] Nigel Gilbert, Klaus G. Troitzsch, "Simulation for the social scientist second edition", Open University Press, 1999.
- [8] Thomas C. Schelling, "Dynamic models of segregation", *Journal of Mathematical Sociology*, Vol.1, pp.143-186, 1971.
- [9] Andrew Gordon, "The Evolution of Labor Relations in Japan: Heavy Industry, 1853-1955", Cambridge, Mass Council on East Asian Studies, Harvard University : Distributed by Harvard University Press, 1985.
- [10] Hideki Yamawaki, "The Evolution and Structure of Industrial Cluster in Japan", *Small Business Economics*, Vol. 18, No. 1, pp. 121–140, 2002.
- [11] Paul L. Robertson, Richard N. Langlois, "Innovation, Networks, and Vertical Integration", *Forthcoming in Research Policy*, Vol. 24, No. 4, pp. 543–562, 1994.
- [12] Thierry Moyaux, Sophie D'Amours, "Supply Chain Management and Multiagent Systems: An Overview", *Studies in Computational Intelligence*, 2006.
- [13] Duygu Turker, Ceren Altuntas, "Sustainable supply chain management in the fast fashion industry: An analysis of corporate reports", *European Management Journal*, Vol. 32, No. 1, pp. 837–849, 2014.
- [14] Soo Wook Kim, "Organizational structures and the performance of supply chain management", *Int. J. Production Economics*, Vol. 106, No. 1, pp. 323–345, 2007.
- [15] Takuya Urakami, "Specialty store strategy within Japanese apparel wholesalers: and empirical analysis", *Journal of Fashion Marketing and Management: An International Journal*, Vol. 14, No. 4, pp. 634–647, 2010.
- [16] Senken-shimbun company, "FASHION BUSINESS DATA BANK (Japanese)", 2011/12, 2013/14, 2015/16.
- [17] Renzo Rosso, Baldassare Bacchi, "Fractal Relation of Mainstream Length to Catchment Area in River Networks", *Water Resources Research*, Vol. 27, No. 3, pp. 381–387, 1991.
- [18] Margaret Bruce, Lucy Daly, "Lean or agile A solution for supply chain management in the textiles and clothing industry?", *Textiles and Pear science*, Vol. 24, No. 2, pp. 151–170, 2004.
- [19] Craig W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, Vol. 21, No. 4, pp. 25–34, 1987.
- [20] Joshua J. Lewer, Hendrik Van den Berg, "A gravity model of immigration", *Economics Letters*, Vol. 99, No. 1, pp. 164–167, 2008.

Opinion Dynamics Theory for Analysis of Consensus Formation and Division of Opinion on the Internet

Akira Ishii [†] Yasuko Kawahata [‡]

[†] Faculty of Engineering, Tottori University, Koayam, Tottori 680-8552, Japan

[‡] Faculty of Social and Information Studies, Gunma University, 4-2 Aramaki-machi, Maebashi, Gunma, 371-8510, Japan
ishii@tottori-u.ac.jp

Abstract: The massive amount of text data on the web has facilitated research on the quantitative analysis of public opinion, which could not be visualized earlier. In this paper, we propose a new opinion dynamics theory. This theory that is intended to explain agreement formation and opinion breakup division in opinion exchanges on social media such as Twitter. With the popularization of the public network, we have become able to communicate with instantaneity and interactivity beyond the temporal and spatial constraints. Research on quantitatively analyzing the distribution of opinion on public opinion that has not been visualized so far utilizing massive web text data is progressing. Our model is based on the Bounded Confidence Model, that expresses opinions in as continuous quantity values. However, in the Bounded Confidence Model, it was assumed that people with different opinions move not in disregard but ignoring opinions. Furthermore, in our theory, it modeled so that it can expresser model incorporates the influence from of the external pressure outside and the phenomenon depending on the surrounding situation.

Keywords: opinion dynamics, Bounded Confidence Model, sociophysics

1. Introduction

Opinion dynamics has a long research history, and many studies have been conducted mainly in the field of sociology [1, 2, 3, 4, 5, 6, 7, 8]. Early studies assumed linearity; however, models incorporating nonlinearity have also been studied [9, 10, 11, 12, 13, 14, 15]. Theoretical progress of the recent years on opinion dynamics are described in the review paper of Sîrbu et. al.[16] Consensus formation has been studied based on the local majority rule as an application of the renormalization group theory in physics [17, 18]. Moreover, applying the theory of magnetic physics, the theory for comparing opinion agreement and opposition to the direction of magnetic moment of magnetism has been studied in the field of social physics [19, 20, 21]. Many mathematical theories on opinion dynamics treat opinions as discrete values of +1 and 0, or +1 and -1. In contrast, certain theories consider opinions as continuous numerical values that can change through the exchange of opinions with others. The bounded confidence model is a representative model of the theory that handles the continuous transition of opinion [22]. In this study, we propose a theory that expresses opinions as continuous values and deals with changes in the opinion values due to the exchange of opinions with others. Moreover, we assume that the opinion of each people can be both positive or negative values. For example, in a study of Tweet on political situation in the United States, there is a study

to classify political opinions from conservative to liberal by one-dimensional axis[23]. In this research, we assume that differences in opinion can be represented by one-dimensional axis values as in this reference. Based on this theory, it is possible to express the division of opinion in society, assuming that opinions of people who disagree with each other are exchanged, and the opinions of both are further divided. Such a division of opinions is a phenomenon often seen on social media such as Twitter.

2. Modelling opinion dynamics

Our model is based on the original bounded confidence model of Hegselmann-Krause[22]. For a fixed agent, say i , where $1 \leq i \leq N$, we denote the agent's opinion at time t by $I_i(t)$. According to Hegselmann-Krause [22], opinion formation of agent i can be described as follows.

$$I_i(t+1) = \sum_{j=1}^N D_{ij} I_j(t) \quad (1)$$

This can be written in the following form.

$$\Delta I_i(t) = \sum_{j=1}^N D_{ij} I_j(t) \Delta t \quad (2)$$

where it is assumed that $D_{ij} \geq 0$ for all i, j in the model of Hegselmann-Krause. Using Based on this definition, $D_{ij} = 0$

means that the opinion of agent i is not affected by the opinion of agent j .

Here, as a result of exchanging opinions, consider the possibility that the opinions of two people with different opinions change move in different directions. Let's us think about consider the distribution of opinions with in the positive and negative directions of a one-dimensional axis. In this case, the value range of $I_i(t)$ is $-\infty \leq I_i(t) \leq +\infty$. If we get the continue value of $I_i(t)$ from -1 up to 1, we can introduce

$$Opinion_i(t) = \tanh(I_i(t)). \quad (3)$$

We modify the meaning of the coefficient D_{ij} as the coefficient of trust. We assume here that $D_{ij} > 0$ if there is a trust relationship between the two persons, and $D_{ij} < 0$ if there is distrust relationship or consensus between the two persons.

Let $A(t)$ be the pressure at time t from the outside and denote the reaction difference for each agent is denoted by the coefficient c_i . Therefore, the change in opinion of the agent can be expressed as follows.

$$\Delta I_i(t) = c_i A(t) \Delta t + \sum_{j=1}^N D_{ij} I_j(t) \Delta t \quad (4)$$

We assume here that D_{ij} and D_{ji} are independent. Usually, D_{ij} is an asymmetric matrix; $D_{ij} \neq D_{ji}$. Moreover, D_{ij} and D_{ji} can have different signs.

Long-term behavior requires attenuation, which means that topics will be forgotten over time. Here we introduce exponential attenuation. The expression is as follows.

$$\Delta I_i(t) = -\alpha I_i(t) \Delta t + c_i A(t) \Delta t + \sum_{j=1}^N D_{ij} I_j(t) \Delta t \quad (5)$$

3. Opinion dynamics for two agents

Let us first consider the case where the opinions of the two agents are the same. In the calculation below, we set $A(t)$ to be Adv as a constant value. In the all calculations in this paper, we assume that $A(t)$ is constant for simplicity in order to pay attention to the effect of D_{ij} . In the actual simulation of the real society behaviors, the external effect $A(t)$ is also significant and time-dependent.

In this case, both opinions are positive. If $D_{ij} > 0$, D_{ij} and $I_j(t)$ is positive. Thus, the opinion $I_i(t)$ moves in the positive direction as shown in fig.1. This means that by having a conversation with an agent of the same positive opinion, agent i will change its opinion to be more and more positive. Similarly, if the opinions of both agents are the same negative opinion, the opinions become more and more negative.

In Fig.2, A is a positive opinion and B is a negative opinion, but it is a case where they trust each other. Namely,

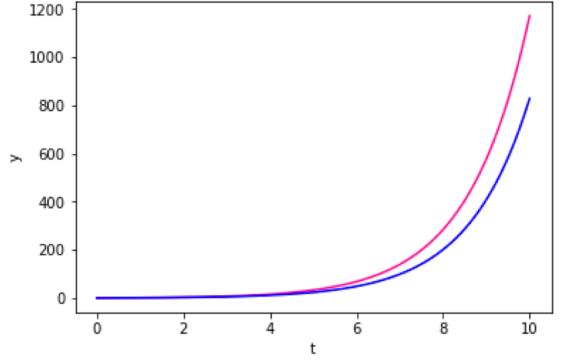


Fig. 1: Calculation result for $N=2$. $Adv= 0.5$, $D_{AB} = 1.0$, $D_{BA} = 0.5$. The initial value is $I_A(0) = 0.005$, $I_B(0) = 0.2$.

we consider the case where the opinions of the two agents are opposite: $I_A(t) > 0$ and $I_B(t) < 0$ where both D_{AB} and D_{BA} are positive values. In this case, $D_{AB}I_B(t)$ is negative. Thus, the opinion of agent A moves to negative because of the effect of agent B having the opposite opinion. The reason why the opinion of agent A moves negatively is that there is a trust relationship with agent B. For the case of $I_A(t) < 0$ and $I_B(t) > 0$ with $D_{AB} > 0$, the result is the same.

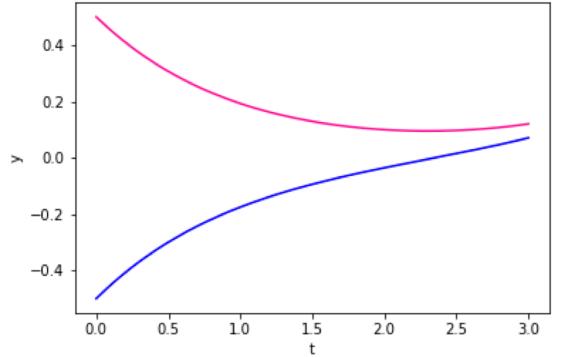


Fig. 2: Calculation result for $N=2$. $Adv= 0.005$, $D_{AB} = 1.0$, $D_{BA} = 1.0$. The initial value is $I_A(0) = 0.5$, $I_B(0) = -0.5$.

Next, we consider the case where the opinions of the two agents are opposite : $I_A(t) > 0$ and $I_B(t) < 0$ where $D_{AB} < 0$. The calculated result is shown in Fig.3. In this case, $D_{AB}I_B(t)$ is positive. Thus, the opinion of agent A moves to more positive. It means that, in a discussion with an agent in disagreement, as there is no trust relationship with that agent, we consider that agent A held his/her opinion more firmly. Similarly, the agent B held his/her opinion more firmly, too. For the case of $I_A(t) < 0$ and $I_B(t) > 0$ with $D_{AB} < 0$, the result is same. This result shows that the dialogue of people who do not trust each other never leads to an agreement.

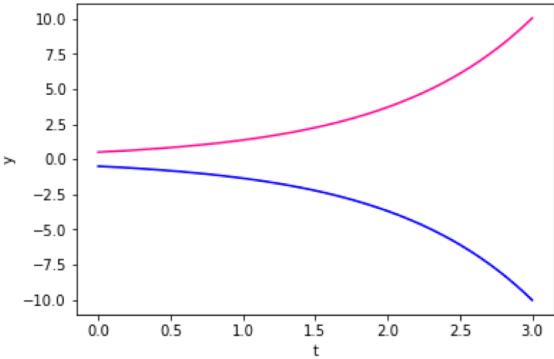


Fig. 3: Calculation result for $N=2$. $Adv=0.005$, $D_{AB} = -1.0$, $D_{BA} = -1.0$. The initial value is $I_A(0) = 0.5$, $I_B(0) = -0.5$.

4. Opinion dynamics for three agents

Next, calculations in the case of three people are shown. A has a positive opinion, B has a negative opinion, and a third person C has an almost neutral opinion. If C's opinion is zero, it obviously does not affect A or B. Even if it is neutral, it is important that C have an opinion on which side. In the following it is assumed to have a slightly positive opinion.

In a situation similar to Fig.3, C is almost neutral in opinion, not much trusted by A and B, and D_{AC} and D_{BC} are small values. The calculation result is shown in Fig.4, but it is almost the same as Fig.3, and A and B repel each other. That is, the influence of C is very small in this case.

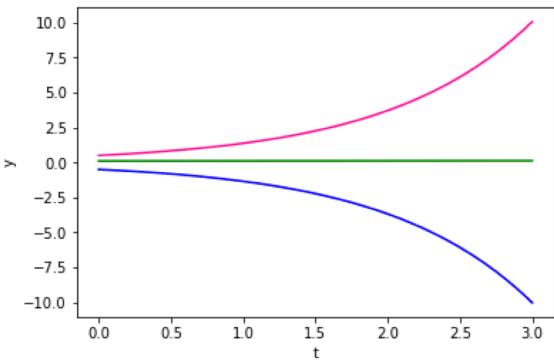


Fig. 4: Calculation result for $N=3$. $Adv = 0.005$, $D_{AB} = -1.0$, $D_{AC} = 0.1$, $D_{BA} = -1.0$, $D_{BC} = 0.1$, $D_{CA} = 0.1$, $D_{CB} = 0.1$. The initial value is $I_A(0) = 0.5$, $I_B(0) = -0.5$, $I_C(0) = 0.1$.

However, if C is strongly trusted by both A and B, the situation will be different. Here we consider that A is a positive opinion and B has a negative opinion and they do not trust each other. However, both have strong confidence in C. Then, as shown in Fig.5 and Fig.6, the opinions of A and B work in the direction converging with C as a vector. In

this case, the initial value of opinion of C is a weak positive opinion, but if it is a weak negative opinion, the opinions of A and B are gathered in a negative direction.

A and B's opinion are getting closer by brokerage of C who is trusted. Approach to C's opinion. In other words, C is a mediator with strong political power that can solve conflict. One example of the person C would be the former president of the Republic of South Africa, Nelson Rolihlahla Mandela who instructed the Republic of South Africa so that all peacefully settled.

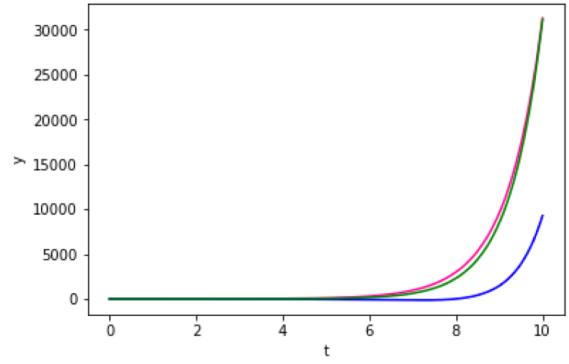


Fig. 5: Calculation result for $N=3$. $Adv = 0.005$, $D_{AB} = -1.0$, $D_{AC} = 1.5$, $D_{BA} = -1.0$, $D_{BC} = 1.5$, $D_{CA} = 1.0$, $D_{CB} = 1.0$. The initial value is $I_A(0) = 0.5$, $I_B(0) = -0.5$, $I_C(0) = 0.1$. $t = 0$ to 10 .

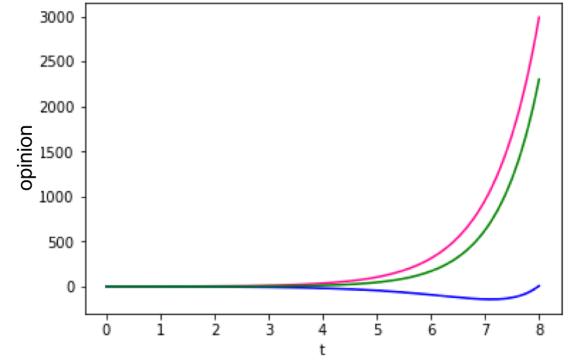


Fig. 6: Calculation result for $N=3$. $Adv = 0.005$, $D_{AB} = -1.0$, $D_{AC} = 1.5$, $D_{BA} = -1.0$, $D_{BC} = 1.5$, $D_{CA} = 1.0$, $D_{CB} = 1.0$. The initial value is $I_A(0) = 0.5$, $I_B(0) = -0.5$, $I_C(0) = 0.1$. $t = 0$ to 8 .

5. Discussion

From a simple result of $N = 2$ and $N = 3$, we can guess to some extent what kind of social phenomena this theory can explain even in the case of many people. The basic equation to solve many person problem is

$$\Delta I_i(t) = -\alpha I_i(t)\Delta t + c_i A(t)\Delta t + \sum_{j=1}^N D_{ij} I_j(t)\Delta t \quad (6)$$

The programming of this equation with general N person is very easy.

In Fig.7, there is the most ideal social setting. If all N people trust each other strongly, society can be guessed that conflict disappears, opinion of people is the same, problem does not occur. As you can see immediately, this situation can easily be calculated with the mathematical model of this study. It is ideal, so to speak, it is similar to the world of song Imagine of John Lennon. It is not realized in actual society. But if the village where a small number of people live is relatively isolated, it would be a possible setting.

Suppose a group of N people is divided into two: a group with positive opinions and a group with negative opinions. All members of the group with positive opinions are connected by trust, and all members of the group with negative opinions are also connected by trust. In this case, consensus building within each group occurs smoothly. However, assume that there is no trust relationship between the group with positive opinions and the group with negative opinions. Therefore, $D_{ij} < 0$ between these two groups.

The conflicting opinions are promoted because the group with positive opinions and the group with negative opinions have no mutual trust relationship. Consequently, it leads to the division of society. The typical example is ecology people and anti-ecology people. In the political history of Japan, at the time of the Meiji Restoration turmoil, the factions wanting to preserve the old regime and the factions establishing a new government fought violently, and this was finally resolved in the civil war of 1868 in Japan. This is a typical example of society being divided. It is believed that a social consensus did not result because the two sides fought intensely and there was no trust relationship. The schematic illustration of this social break is shown in fig.8.

Here, we consider a person isolated in a group. The opinion of the group is positive, whereas only the opinion of the isolated person is negative. As a group consisting of N people, $N - 1$ persons excluding the isolated person do not trust the isolated person, while that isolated person also does not trust the other $N - 1$ persons. In this case, $D_{ij} < 0$ between the isolated person and the other persons in the group. In this case, the opinions of the $N - 1$ people in the group become increasingly positive, and the opinion of the isolated person becomes increasingly negative. Therefore, an isolated person and another member of the group are completely divided.

Even if $N - 1$ people in the group do not trust an isolated person, the situation would be different if the isolated person trusts the other $N - 1$ persons in the group. In this case, let i be an isolated person, $D_{ij} > 0$ and $D_{ij} < 0$. The opinion of

Consensus formation

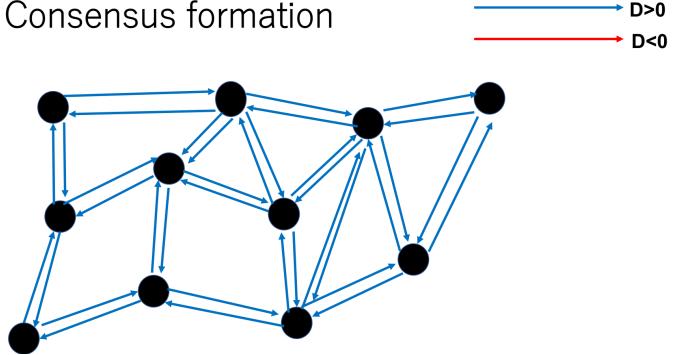


Fig. 7: Schematic illustration of consensus formation of this theory.

Social break

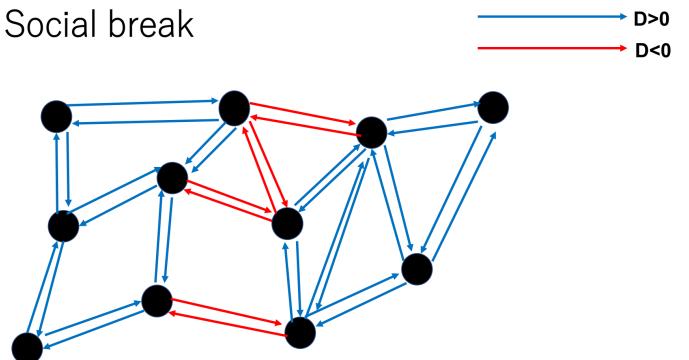


Fig. 8: Schematic illustration of social break of this theory.

the isolated person gradually approaches the group's opinion owing to the influence of the other people in the group and external forces such as media, such that isolation is gradually resolved. Moreover, if other people in the group change to trust the people who are isolated, the isolation will be resolved more quickly. Whether this situation can be solved, for example, by trusting this person by a small number of people can be calculated by the mathematical model of this research. The schematic illustration of this social isolation is shown in Fig.9.

If an individual is afraid of isolation, it is considered that the social group or society as a whole cannot exclude individuals for opinion of members. This fear of isolation is a case study that often results in silence rather than expressing an opinion. Media is an important factor related to both dominant ideas and people's perception of dominant ideas, and the assessment of the social environment does not necessarily correlate with reality.

People fear social isolation, and try to avoid isolation as $D_{ij} > 0$ among them. Moreover, when trying to enter the majority in society, each individual becomes confused as to which opinion is the majority. Based on mass media

information and the local majority of surrounding people, we estimate which opinion is the majority opinion. In our new opinion dynamics theory, the effect of mass media can be included by $A(t)$.

As a result of combining these effects, the majority is formed rapidly. Even in the opinion of the majority in the early stage, if opinions are contrary to the information from mass media and surrounding people, people with different opinions become silent as $D_{ij} > 0$ within the surroundings. Thus, the silence spiral [25] can also be explained by our new opinion dynamics theory.

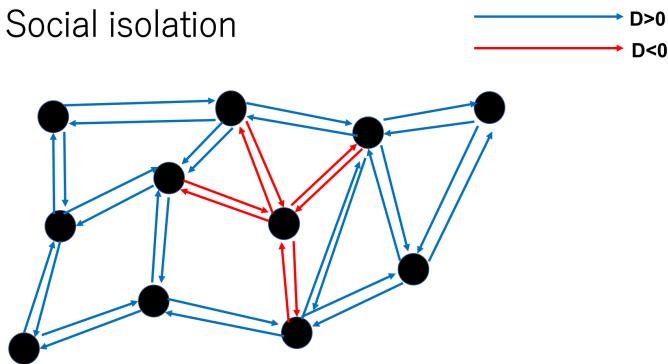


Fig. 9: Schematic illustration of social isolation of this theory.

In future calculations, the external effect $A(t)$ as mass media information or statement from government will also play an important role in the simulation using our new opinion dynamics theory of this paper. In that case, the external effect $A(t)$ is the time-dependent function. Moreover, the coefficient c_i can be positive or negative value. In the case of negative c_i , the person i obtain negative effect from the external effect $A(t)$.

6. Conclusion

In this research, we presented a theory of opinion dynamics that considers the opinion of each person a continuous value, rather than a discrete value. Opinions are represented by real numbers ranging from positive to negative. We introduce "trust" and "distrust" as a coefficient of each person pairs. In addition to the influence of opinion exchanges within each group, we constructed a mathematical model that incorporates external pressure. Using this theory, we can mathematically express many phenomena that can occur in a group in society.

In this new opinion dynamics theory, it is possible to calculate the dynamics of a complicated system mixed with people's trust and suspicion. Also, as there is no upper limit on the opinion, we can explain the situation where opinions are getting sharper and sharper. Simulation of a large number of people is also prepared. In the future, we will compare and

examine which case is assumed whether this theory conforms to actual data concerning speech in actual political and social problems.

As a future prospect, we would like to conduct researches on reactions to various social risks, discussions on Fake News. Definition differs depending on the scope such as macro disposers, civil disputes, shocks of market transactions etc. Generally occurring in a microscopic range such as disease, crime, ethnic discrimination.

Acknowledgement

The authors are grateful for discussion with Prof. Serge Galam.

References

- [1] French J R P (1956) A formal theory of social power. *Psychological Review* 63. pp. 181-194.
- [2] Harary F (1959) "A criterion for unanimity in French 's theory of social power". In Cartwright D (Ed.), *Studies in Social Power*. Institute for Social Research, Ann Arbor.
- [3] Abelson, R P (1964), "Mathematical models of the distribution of attitudes under controversy". In Frederiksen N and Gulliksen H (Eds.), *Contributions to Mathematical Psychology*, New York, NY: Holt, Rinehart, and Winston.
- [4] De Groot M H (1974) Reaching a consensus. *J. Amer. Statist. Assoc.* 69. pp. 118 - 121.
- [5] Lehrer K (1975) Social consensus and rational agnoiology. *Synthese* 31. pp. 141 - 160.
- [6] Chatterjee S (1975) Reaching a consensus: Some limit theorems. *Proc. Int. Statist. Inst.* pp.159 - 164.
- [7] Chatterjee S and Seneta E (1977) Toward consensus: some convergence theorems on repeated averaging. *J. Appl. Prob.* 14. pp. 89 - 97.
- [8] Wagner C G (1978) Consensus through respect: a model of rational group decision-making. *Philosophical Studies* 34. pp. 335 - 349.
- [9] Krause U (1997), "Soziale Dynamiken mit vielen Interakteuren. Eine Problemkizze". In Krause U and Stöckler M (Eds.) *Modellierung und Simulation von Dynamiken mit vielen interagierenden Akteuren*, Universität Bremen. pp. 37 - 51.
- [10] Beckmann T (1997) Starke und schwache Ergodizität in nichtlinearen Konsensmodellen. Diploma thesis Universität Bremen.
- [11] Hegselmann R and Flache A (1998) Understanding complex social dynamics - a plea for cellular automata based modelling. *Journal of Artificial Societies and Social Simulation*, vol. 1 no. 3.
- [12] Krause U (2000), "A discrete nonlinear and non-autonomous model of consensus formation". In Elaydi S, Ladas G, Popenda J and Rakowski J (Eds.), *Communications in Difference Equations*, Amsterdam: Gordon and Breach Publ. pp. 227 - 236.
- [13] Deffuant G, Neau D, Amblard F, and Weisbuch G (2000) Mixing beliefs among interacting agents. *Advances in Complex Systems* 3. pp. 87 - 98.

- [14] Dittmer J C (2001) Consensus formation under bounded confidence. *Nonlinear Analysis* 47. pp. 4615 - 4621.
- [15] Weisbuch G, Deffuant G, Amblard F and Nadal J P (2001), Interacting agents and continuous opinion dynamics. <<http://arXiv.org/pdf/cond-mat/0111494>>
- [16] Sîrbu A., Loreto V., Servedio V.D.P., Tria F. (2017) Opinion Dynamics: Models, Extensions and External Effects. In: Loreto V. et al. (eds) *Participatory Sensing, Opinions and Collective Awareness. Understanding Complex Systems*. Springer, Cham
- [17] Galam S, "Application of statistical physics to politics" *Physica A: Statistical Mechanics and its Applications* 274, 1999, Pages 132-139
- [18] Galam S, "Real space renormalization group and totalitarian paradox of majority rule voting" *Physica A: Statistical Mechanics and its Applications* 285, Issues 1-2, 15 September 2000, Pages 66-76
- [19] Galam, *Physica A* 238, 66 (1997).
- [20] Sznajd-Weron and J. Sznajd, *Int. J. Mod. Phys. C* 11, 1157 (2000)
- [21] Sznajd-Weron, M. Tabiszewski, and A. M. Timpanaro, *Europhys. Lett.* 96, 48002 (2011).
- [22] Hegselmann R and U Krause, "Opinion Dynamics and Bounded Confidence Models, Analysis, and Simulation" *Journal of Artificial Society and Social Simulation* 5 (2002)
- [23] C A Bail, L P Argyle, T W Brown, J P Bumpus, H Chen, M B F Hunzaker, J Lee, M Mann, F Merhout and A Volkovskiy, "Exposure of opposing views on social media can increase political polarization" *PNAS* 1804840115 (2018)
- [24] M Tsubokura, Y Onoue, H A Torii, S Suda, K Mori, Y Nishikawa, A Ozaki and K Uno, "Twitter use in scientific communication revealed by visualization of information spreading by influencers within half a year after Fukushima Daiichi nuclear power plant accident" *PLoS ONE* 13(9): e0203594 (2018)
- [25] Noelle-Neumann, E. (1974). The spiral of silence a theory of public opinion. *Journal of communication*, 24(2), 43-51.

Towards a Robotic Swarm using Deep Neuroevolution: An Experimental Study in Path Formation

Daichi Morimoto [†] Motoaki Hiraga [†] Kazuhiro Ohkura [†]

[†] Hiroshima University, 1-4-1, Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan
{morimoto,hiraga,ohkura}@ohk.hiroshima-u.ac.jp

Abstract: Swarm robotics is the study of a large group of autonomous robots that emerges collective behavior without relying on any centralized control. The evolutionary robotics approach is a promising method to design controllers for a robotic swarm. In this paper, we applied *deep neuroevolution*, which is a technique to design deep neural networks by an evolutionary algorithm, to a path formation of a robotic swarm. The controller designed with the deep neuroevolution succeeded in forming the path. In addition, this paper discusses the performance using the deep neuroevolution comparing with a traditional evolutionary robotics approach.

Keywords: Swarm robotics, Evolutionary robotics, Deep neuroevolution, Path formation

1. Introduction

Swarm robotics is the study of the collective behavior of a large group of autonomous robots [1]. The collective behavior of a robotic swarm emerges from the local interactions among robots and between the robots and the environment. The design methods of controllers for a robotic swarm are classified into two categories [2], i.e., behavior-based design methods and automatic design methods. In behavior-based design methods, the behavior of each robot is designed manually by trial and error. On the other hand, automatic design methods cast the parameters of the controller into an optimization problem and automatically design using an optimization algorithm [3].

A widely used automatic design method is an evolutionary robotics approach [4]. Typically, this approach applies artificial neural networks as the controllers of robots. The parameters of the artificial neural network are optimized by an evolutionary algorithm. This technique to evolve artificial neural networks is known as *neuroevolution* [5]. Evolutionary robotics approaches have succeeded in developing the collective behavior of robotic swarms in tasks such as aggregation [6], flocking [7], and group transportation [8, 9].

In this paper, *deep neuroevolution* [10], which is a technique to design deep neural networks by an evolutionary algorithm, is applied to a path formation of a robotic swarm. In evolutionary robotics approach, there are works that succeeded in generating a collective behavior in complex tasks using the hierarchical controller consisting of sub-controllers [11, 12]. From this fact, artificial neural networks with complex structure might have a possibility to generate a complex collective behavior. This paper applies the deep neuroevolution to a path formation of a robotic swarm and

discusses the performance.

The remainder of this paper is organized as follows. Section 2 briefly explains the related work. Section 3 describes the experiment settings. Section 4 shows the results of the experiments. Finally, conclusions and future work are in Section 5.

2. Related Work

The evolutionary robotics approach is a technique to design robots using an evolutionary algorithm [4]. This approach typically applies *neuroevolution* to designing the controllers of robots. Neuroevolution is a technique to optimize the parameters of artificial neural networks by using an evolutionary algorithm [5, 13]. In general, neuroevolution is divided into two categories; i.e., methods that only evolve the synaptic weights and evolving topologies along with the synaptic weights.

Deep neuroevolution is a technique to apply the neuroevolution to deep neural networks. The deep neural networks are family of artificial neural networks, which composed of multiple layers. There are some related works that evolve deep neural networks using natural evolution strategy [14] and genetic algorithm [10]. These works have done in the benchmarks for reinforcement learning. Therefore, the deep neuroevolution is only applied to problems dealing with a single agent, i.e., none of these problems deal with multi-agent systems. In this paper, we applied the deep neuroevolution to a robotic swarm, which consists of multiple robots.

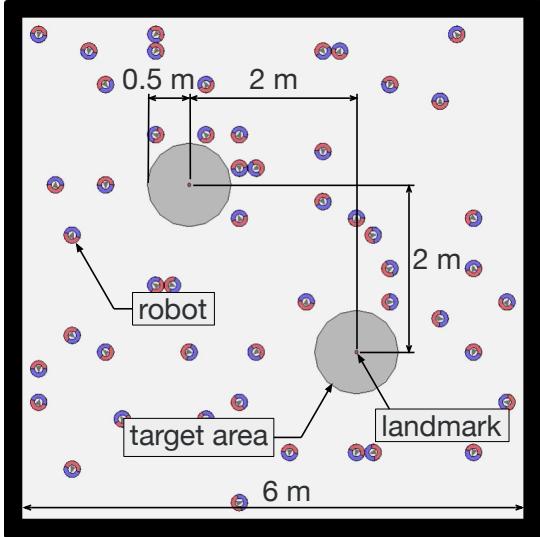


Fig. 1: Screenshot of the environment in the computer simulations.

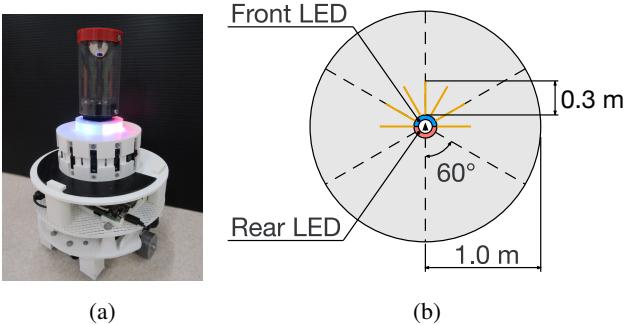


Fig. 2: (a) Physical robot modeled in the computer simulations. (b) Configuration of the omnidirectional camera and distance sensors.

3. Experiment settings

3.1 Path formation task

The path formation task is one of the fundamental tasks addressed in the field of swarm robotics [15, 16]. This task aims to develop a collective path of robots and navigate between two landmarks. Fig. 1 shows the initial state of the simulated environment. Two landmarks, which have LEDs that emit red color, are placed diagonally in the square-shaped environment. Each landmark has a target area with a radius of 0.5 m.

3.2 Robot

Fig. 2(a) shows the physical robot that is implemented as a model of the robot in the computer simulations. The body of the robot is 0.2 m in diameter with two propulsion wheels attached to the left and right side of the robot. Each robot has seven distance sensors, a ground sensor, an omnidirectional

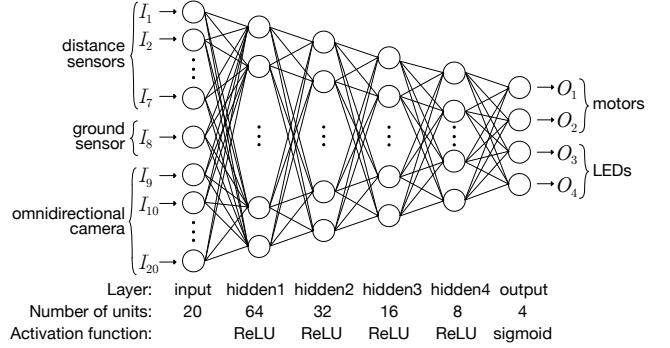


Fig. 3: Structure of the controller of the robot.

camera, and RGB LEDs. Distance sensors are attached to the front side of the robot, as shown in Fig. 2(b). Each value from the distance sensor is normalized in the range of $[0, 1]$. The ground sensor detects if the robot is inside or outside the target area and returns a binary value. The LEDs are attached around the body of the robot and emit blue lights from the front and red lights from the rear of the robot. Both front and rear LEDs can be turned on and off independently. The omnidirectional camera detects LEDs within the sensor range. The visible area of the omnidirectional camera is divided into six sections, as shown in Fig. 2(b). The robot is assumed to detect only the presence of colored LED lights for each section via the omnidirectional camera. The process of the omnidirectional camera is done independently on red and blue color. From each section, it returns a binary value depending on the presence of the corresponding colored LEDs.

3.3 Controller

The controller of the robot is represented by a deep feed-forward neural network, which is a fundamental and widely applicable deep neural network. Fig. 3 shows the structure of the controller. All synaptic weights take real values in the range of $[-1, 1]$. Twenty units in the input layer receive seven real values from distance sensors, a binary value from the ground sensor, and twelve binary values from the omnidirectional camera. All hidden layers are the fully connected layers with the rectified linear unit (ReLU) activation function. The ReLU activation function is described as the following equation:

$$f(x) = \max(0, x). \quad (1)$$

The output layer has four units with a standard sigmoid function, which is represented as the following equation:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2)$$

Two units of the output layer determine the velocities of two wheels attached to each side of the robot. Another two units determine the activation of LEDs in front and rear side of the robots.

Table. 1: Parameter settings for the (μ, λ) evolution strategy.

Parameter	Value
Number of parents μ	30
Number of offspring λ	200
Initial mutation step size	0.05
Mutation step size	$\in [0.00001, 0.15]$

3.4 Deep Neuroevolution

The parameters of the controller are generated using deep neuroevolution. The synaptic weights of the deep feed-forward neural network are optimized by an evolutionary algorithm. The controllers are evaluated via fitness function, which indicates the achievement of the task. The settings of the evolutionary process are based on the previous work [17, 18].

A (μ, λ) evolution strategy [19, 20] is employed for optimizing the controllers. Table 1 shows the settings of (μ, λ) evolution strategy. In the first generation, the controllers are initialized randomly. Each controller is implemented to N robots and evaluated for $M = 3$ independent trials. Each trial lasts 7200 time steps with no evaluation during the first 1200 time steps. In the remaining 6000 time steps, the fitness of i th robot is updated with the following equation:

$$f_i(t) = f_i(t-1) + \begin{cases} 1 & \text{if robot } i \text{ enters} \\ & \text{a new target area,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This equation indicates that the fitness f_i equates to the number of times the i th robot entered a target area that is different from the one visited previously. After $M = 3$ trials, the fitness of the controller F is calculated with the following equations:

$$F = \frac{1}{M} \sum_{m=1}^M F_m \quad F_m = \frac{1}{N} \sum_{i=1}^N f_i, \quad (4)$$

where F_m is the fitness of m th trial, M is the total number of trials, and N is the number of robots. After calculating all of the fitness F for this generation, the controllers of the next generation are produced based on the (μ, λ) evolution strategy. This evolutionary process is repeated until the last generation, that is, the 1000th generation.

4. Results

In this paper, the experiments are conducted using $N = 50$ robots. The controller that scored the best fitness is applied to generate collective behavior in the path formation task. Fig. 4 shows the path formation that is generated using the deep feed-forward neural network. The robots evolved to communicate with LED signals to form a path efficiently. Thus, the robotic swarm was able to accomplish the task with deep neuroevolution.

In addition, the performance using the deep feed-forward neural network is compared with traditional artificial neural networks. A neural network composed of three layers with ten hidden units is employed for comparison. We have prepared two types of this three-layered controller; one with recurrent connections as described in [17, 18] and another with only feed-forward connections with sigmoid activation function. Fig. 4 shows the results using the deep feed-forward neural network (DNN), recurrent neural network with three layers (RNN), and feed-forward neural network with three layers (FNN). All three controllers were able to perform path formation successfully with similar swarm behavior. There was no significant difference in the performance between the deep feed-forward neural network and the other two three-layered neural networks (Mann-Whitney U test, p -value > 0.05). This might be because the task was too easy to be accomplished with the traditional feed-forward neural network with three layers. When the performance is compared with the median and mean values, the recurrent neural network scored the best in the three controllers. However, the deep feed-forward neural network performed slightly better than the traditional feed-forward neural network. From these results, there might be an advantage of increasing the number of layers; however, it appears to be difficult for deep feed-forward neural networks to outperform recurrent neural networks.

5. Conclusions

In this paper, we applied the deep neuroevolution to the path formation of the robotic swarm. The controller with the deep feed-forward neural network succeeded in forming the path. However, the deep feed-forward neural network was not enough to outperform the traditional neural networks in an evolutionary robotics approach of swarm robotics. This might be because the task was too easy for the robotic swarm to discuss the performance.

As a future work, we are planning to investigate the effect of the deep neuroevolution in more complex tasks. At the same time, we plan to examine the effect of the deep neuroevolution on a robotic swarm using the camera input. The deep neural networks have remarkable achievements in extracting features from images. We are planning to use deep neural networks with convolutional neural networks as a future work.

References

- [1] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” in *Swarm Robotics*, ser. Lecture Notes in Computer Science, Vol. 3342. Springer, 2005, pp. 10–20.
- [2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intelligence*, Vol. 7, No. 1, pp. 1–41, 2013.

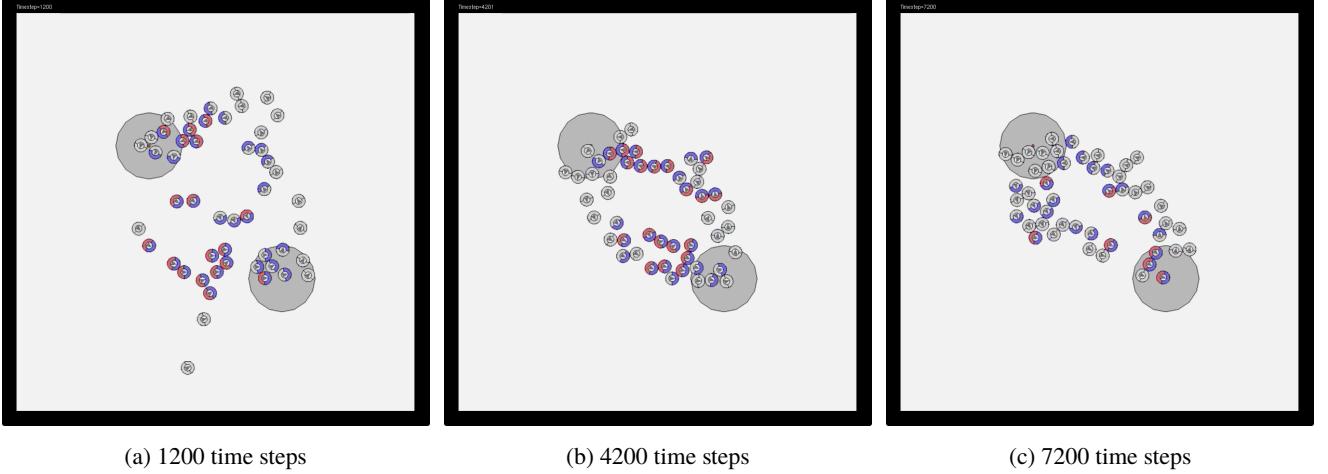


Fig. 4: Path formation of the robotic swarm using the deep feed-forward neural network. The color of the robots describes the activation of the LEDs; blue and red for the activation of the front and rear LEDs respectively, and gray for deactivation of LEDs.

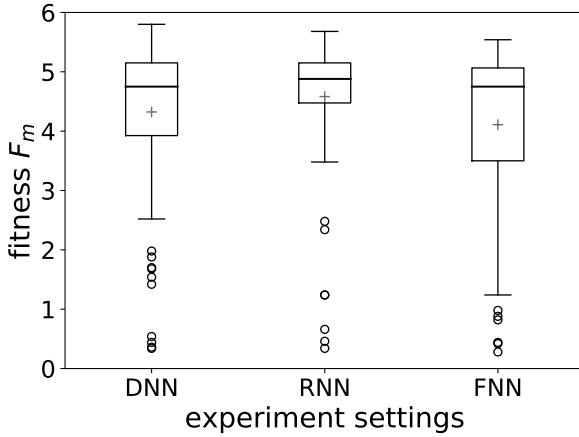


Fig. 5: Box plots of the fitness F_m with different controllers over $M = 100$ trials. Gray crosses indicate the mean value.

- [3] G. Francesca and M. Birattari, “Automatic design of robot swarms: achievements and challenges,” *Frontiers in Robotics and AI*, Vol. 3, p. 29, 2016.
- [4] S. Nolfi and D. Floreano, *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.
- [5] D. Floreano, P. Dürr, and C. Mattiussi, “Neuroevolution: from architectures to learning,” *Evolutionary Intelligence*, Vol. 1, No. 1, pp. 47–62, 2008.
- [6] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo, “Evolving aggregation behaviors in a swarm of robots,” in *European Conference on Artificial Life*. Springer, 2003, pp. 865–874.
- [7] G. Baldassarre, S. Nolfi, and D. Parisi, “Evolving mobile robots able to display collective behaviors,” *Artificial life*, Vol. 9, No. 3, pp. 255–267, 2003.
- [8] R. Groß and M. Dorigo, “Towards group transport by swarms of robots,” *International Journal of Bio-Inspired Computation*, Vol. 1, No. 1-2, pp. 1–13, 2009.
- [9] R. Groß and M. Dorigo, “Evolution of solitary and group transport behaviors for autonomous robots cap-

pable of self-assembling,” *Adaptive Behavior*, Vol. 16, No. 5, pp. 285–305, 2008.

- [10] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning,” *arXiv preprint arXiv:1712.06567*, 2017.
- [11] M. Duarte, S. M. Oliveira, and A. L. Christensen, “Evolution of hierarchical controllers for multirobot systems,” in *Artificial life 14: Proceedings of the international conference on the synthesis and simulation of living systems*, 2014, pp. 657–664.
- [12] Y. Wei, M. Hiraga, K. Ohkura, and Z. Car, “Autonomous task allocation by artificial evolution for robotic swarms in complex tasks,” *Artificial Life and Robotics*, in press.
- [13] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, Vol. 10, No. 2, pp. 99–127, 2002.
- [14] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017.
- [15] V. Sperati, V. Trianni, and S. Nolfi, “Self-organised path formation in a swarm of robots.” *Swarm Intelligence*, Vol. 5, No. 2, pp. 97–119, 2011.
- [16] L. Bayindir, “A review of swarm robotics tasks,” *Neurocomputing*, Vol. 172, pp. 292–321, 2016.
- [17] M. Hiraga, T. Yasuda, and K. Ohkura, “Evolutionary acquisition of autonomous specialization in a path-forming task of a robotic swarm,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 22, No. 5, pp. 621–628, 2018.
- [18] M. Hiraga, Y. Wei, T. Yasuda, and K. Ohkura, “Evolving autonomous specialization in congested path formation task of robotic swarms,” *Artificial Life and Robotics*, in press.
- [19] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies: A comprehensive introduction,” *Natural computing*, Vol. 1, No. 1, pp. 3–52, 2002.
- [20] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2003.

Leading a swarm of mobile robots via a locally informed teleoperated robot

Toshiyuki Yasuda [†] Shunsuke Ota [†] Mitsuru Jindai [†]

[†] University of Toyama, 3190, Gofuku, Toyama, 930-8555, Japan
yasuda@eng.u-toyama.ac.jp

Abstract: This paper investigates how well human operators can control a robot, the task of which is leading a flock of mobile robots to their target area. An operator teleoperates a robot with limited sensing ability to lead as many robots as and to drive as fast as possible toward the goal area. The teleoperated robot has a laser range finder, a camera on the front side, cameras on the front and rear, or an omnidirectional camera as a sensing device and sends the information perceived by either of them to the operator. Flocking robots make decisions using a Boid model. The flocking performance is evaluated in computer simulations.

Keywords: Swarm robotics, human-swarm interaction, Boid, flocking

1. Introduction

The research domain on collective behavior of robot groups, known as Swarm Robotics (SR), has attracted increasing attention in recent years [1]. Swarm robotic systems (SRSs) consist of many homogeneous simple robots that operate without global controllers. The advantages of SRSs over other robotic systems are that they are robust, flexible, and scalable. The collective behavior of robotic swarms emerges from the same self-organization processes that occur in biological swarms.

The shortcomings derived from the simplicity of each individual can be compensated by human intervention. Besides, human-swarm interaction (HSI) is expected in (i) indirectly delivery of the information out of the sensing range of the swarm, and (ii) conveyance of changes in their goals and situations [2]. This paper examines a remote interaction between a human and a swarm of robots based on local information detected through the sensing device on a teleoperated robot.

The remainder of this paper is organized as follows. The design of the teleoperated leader robot is described in Sections 2. Section 3 and 4 present experimental settings and results, respectively. The conclusions and ideas for future work are described in Section 5.

2. HSI via a tele-operated robot

Interactions between humans and robotic swarms are categorized into two forms: proximal and remote interactions. In the proximal interactions, it is assumed that operators and swarms share their workspace. Operators can directly observe a swarm and interact with it. Gesture, face, and speech recognition implemented on a robotic swarm are popular ap-

proaches in this type of interactions.

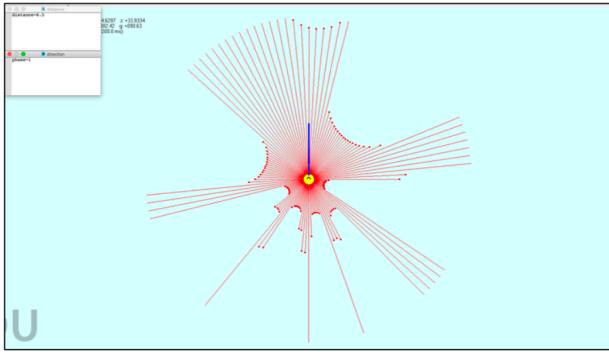
In remote interactions, communication is one of the main issues. These interactions are expected to deploy a swarm in inaccessible or dangerous areas for humans. Sensor networks and mobile adhoc networks for collect global information about swarms and environments are applicable in such situations. We conduct experiments to investigate how the local information from a remote robot affects the behavior of robotic swarms. We are assuming a remote robot has a laser range finder, a camera on the front side, cameras on the front and rear, or an omnidirectional camera for perceiving its peripheral information as shown in Fig. 1. Figure 2 shows a snapshot in which a human operator obtains sensory information from the teleoperated robot and controls the robot using a gamepad.

3. Swarm robot flocking

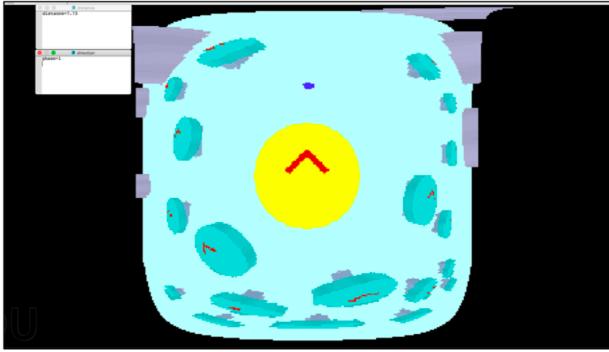
3.1 Experimental settings

Computer simulations are conducted with an open source physical simulator, V-REP. The swarms are evaluated on 19 robots placed in a rectangular experimental field with many obstacles as shown in Fig. 3. The task of the robots is moving from the start area to the goal area via two relay points while avoiding obstacles. The minimum distance to be traveled is 25 m in the simulation unit. The robotic swarm colored in cyan is randomly placed in a grid of the starting area at the beginning of each experimental run while the teleoperated robot colored in yellow is at the center of the rightmost column as shown in Fig. 4.

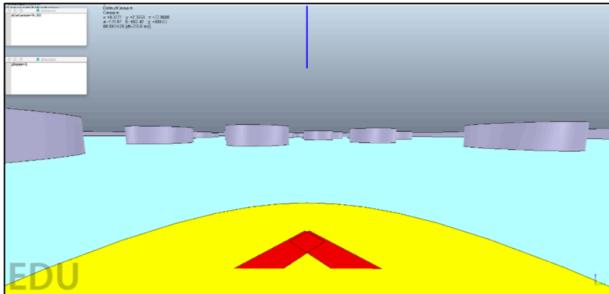
The swarm has no information about the relay and goal points whilst operators can obtain the information about the



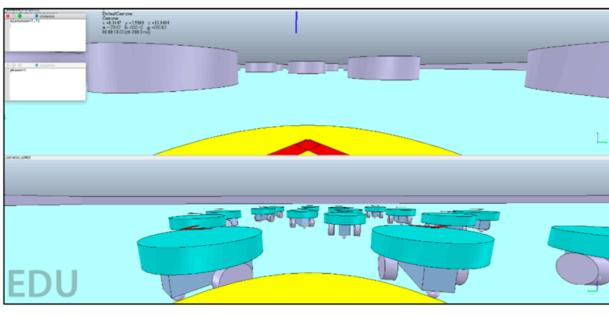
(a) Laser range finder



(b) Omnidirectional camera



(c) Front camera



(d) Front and rear cameras

Fig. 1: Sensory information from the tele-operated robot

direction toward the points. The operators control the moving direction of the teleoperated robot to lead the swarm toward the goal as fast as possible and keep as large aggregation as possible.

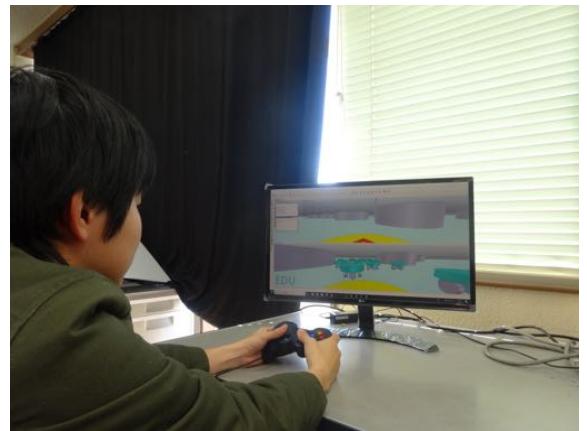


Fig. 2: Experiment overview

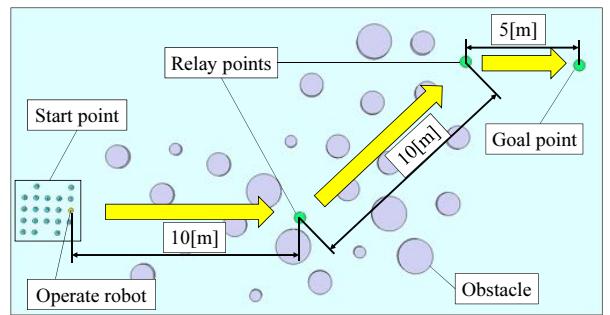


Fig. 3: Experimental environment

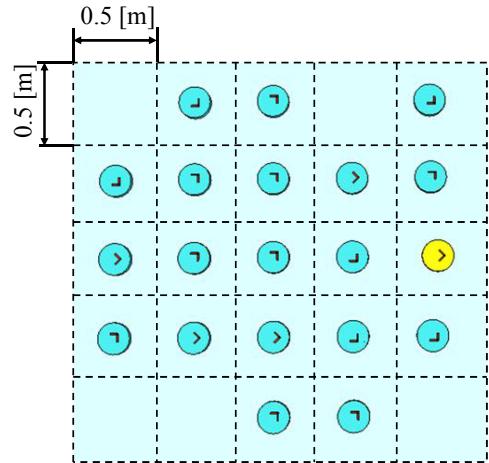


Fig. 4: Initial position of the robots: the teleoperated robot (cyan) and the swarms (yellow)

3.2 Robotic swarms

The motions of SRSs are based on the well-known computer animation model Boid [3]. Reynolds showed that flocking in the Boid model emerges from three simple behaviors: collision avoidance, velocity matching, and flock centering. Figure 5 shows the range of each behavioral rule based on the Boid model. The separation and attraction rules enable the

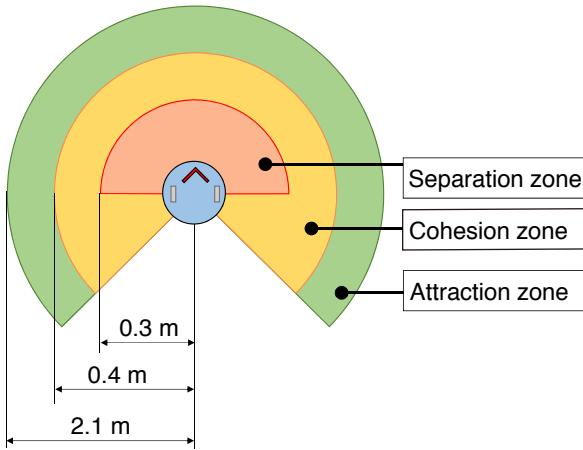


Fig. 5: Boid model

robot to avoid collisions and remain close to its neighbors, respectively. The cohesion rule is used for alignment heading directions \mathbf{h} .

$$\mathbf{h} = \begin{cases} \mathbf{S} & \text{if neighbor(s) in the separation area} \\ \frac{c\mathbf{C}+a\mathbf{A}}{\|c\mathbf{C}+a\mathbf{A}\|} & \text{otherwise} \end{cases} \quad (1)$$

where \mathbf{S} , \mathbf{C} , and \mathbf{A} denote the separation, cohesion, and attraction vectors, respectively. The constants a and c are set within the range [0.0, 1.0]. In the experiments, we set both a and c to 1.0 as these were the recommended values in preliminary experiments.

The vectors are calculated as follows:

$$\mathbf{S} = \mathbf{u}_i \left(\frac{d_i}{r_s} - 1 \right), \mathbf{C} = \frac{1}{n_c} \sum \mathbf{v}_j, \mathbf{A} = \frac{1}{n_a} \sum \mathbf{w}_k, \quad (2)$$

where r_s and d_i denote the radius of the separation area and the distance to the i -th robot in the separation area. \mathbf{u}_i , \mathbf{v}_j , and \mathbf{w}_k are the unit vectors in the direction toward the corresponding neighbor in the sensing range. n_c and n_a are the number of robots in the cohesion and attraction area, respectively.

The heading direction vectors are transformed into signals that actuate the two side-wheels of the mobile robot in Fig. 6. The rotation velocities of the left and right wheels, denoted N_L^s and N_R^s respectively, are calculated as

$$\theta_h = \angle \mathbf{h}_c - \angle \mathbf{h} \quad (3)$$

$$u = \begin{cases} u_{max} \cos \theta_h & \text{if } |\theta_h| \leq \frac{\pi}{2}, \\ u_{min} & \text{otherwise} \end{cases} \quad (4)$$

$$\omega = \theta_h K_p \quad (5)$$

$$N_L^s = \left(u + \frac{\omega l}{2} \right) \frac{60}{2\pi r} \quad (6)$$

$$N_R^s = \left(u - \frac{\omega l}{2} \right) \frac{60}{2\pi r} \quad (7)$$

where, u_{max} and u_{min} respectively denote the maximum and minimum speed of the robot, $K_p = 0.5$ is a proportional gain,

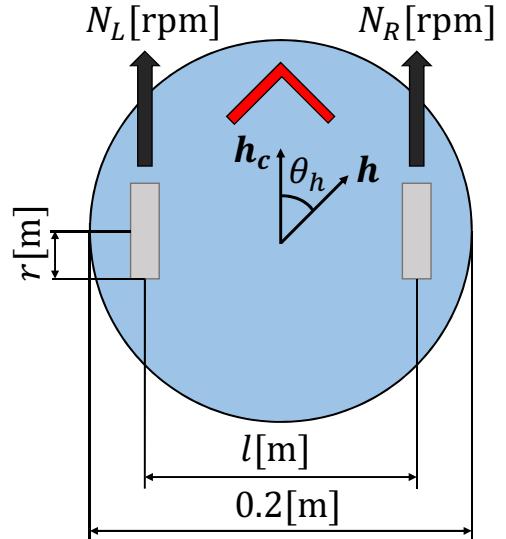


Fig. 6: Mobile robot

l is the distance between the left and right wheels, and r is the wheel radius.

3.3 Teleoperated robot

Operators send motor signals for throttle and steering, denoted *speed* and *steer*, using a gamepad to lead as many robots as and to move as fast as possible toward the goal area. The moving mechanism of the teleoperated robot is as same as that of robots in the swarm. The rotation velocities of the left and right wheels, denoted N_L^t and N_R^t respectively, are calculated as

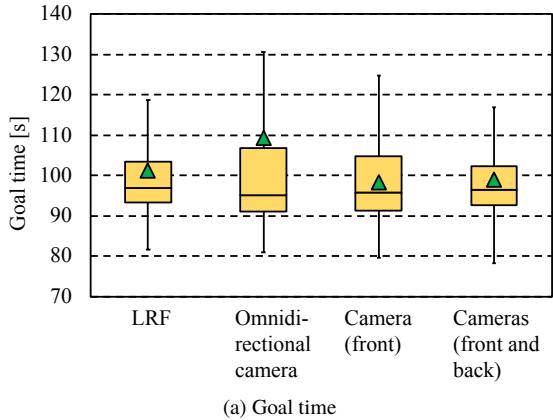
$$N_L^t = \left(speed + \frac{steer}{2} \right) \frac{60}{2\pi} \times 0.01, \quad (8)$$

$$N_R^t = \left(speed - \frac{steer}{2} \right) \frac{60}{2\pi} \times 0.01. \quad (9)$$

4. Results

Twenty operators conduct three trials for each of four sensing devices. We evaluate the flocking performance by using two criteria: speed and coherence. The speed and coherence criteria are based on the goal time and the aggregate size at the goal, respectively. Figure 7 show the experimental results. The significant difference is not obtained in the goal time. The aggregation of a remote robot with an omnidirectional camera, however, finishes the task, organizing a significantly smaller aggregate compared by that with LRF and front, rear cameras.

Table 1 shows the number of trials on aggregation size at the end of the experiments and the relationship between the goal time and aggregate. It is found that as the slower the leader reaches the goal, the coherence gets lower. Such a tendency is clearly observed in the LRF experiments, however, not in the front camera experiments.



(a) Goal time

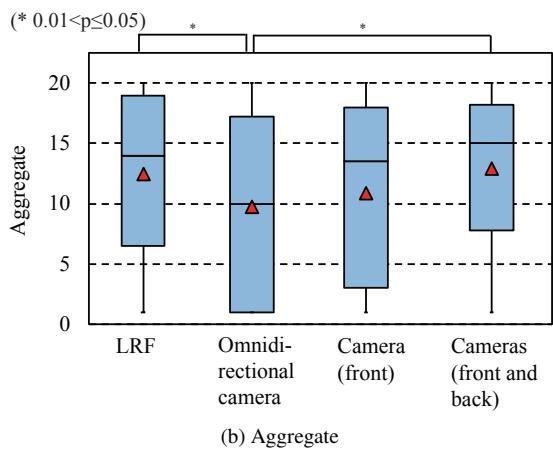


Fig. 7: Experimental results

From these results, it can be said that rich information around the teleoperated robot is not always required and well-simplified information yields better operation. However, all of the operators are not so experienced in the experiments. Operation improvement should be investigated as the evaluation of the information sent to operators.

5. Conclusion

This paper examined the effect of displaying the sensory information perceived by a teleoperated robot for interacting a robotic swarm. Human operators controlled a remote robot that has a limited sensing capability. The flocking performance was evaluated from the viewpoint that which sensing device could be useful for operators. The robot with LRF organized the largest aggregate and got to the goal fastest, compared with that with cameras in the computer simulations.

In the future, experiments using real robots will be conducted. We are planning to take latency in sending/receiving signals to/from the teleoperated robot into consideration. We are interested in HSI with robots that can learn.

Table. 1: Results in each segment of the aggregate size

		(a) Number of trials			
		# robots			
		20–16	15–11	10–6	5–1
LRF		27	9	9	15
Omnidirectional camera		17	10	10	23
Front camera		23	9	10	18
Front and rear cameras		29	11	9	11

		(b) Average time			
		# robots			
		20–16	15–11	10–6	5–1
LRF		98.7	96.0	99.1	101.2
Omnidirectional camera		100.4	106.2	120.0	112.4
Front camera		96.9	100.7	101.4	96.8
Front and rear cameras		97.7	100.7	98.1	101.8

References

- [1] E. Şahin, “Swarm Robotics: From Sources of Inspiration to Domains of Application”, Swarm Robotics WS 2004, LNCS 3342, pp. 10–20, 2005.
- [2] A. Kolling, P. Walker, N. Chakraborty, K. Sycara and M. Lewis, “Human Interaction With Robot Swarms: A Survey”, IEEE Transactions of Human-Machine Systems, Vol. 46, No. 1, pp. 9–26, 2016.
- [3] C. Reynolds, “Flocks, Herds and Schools: a Distributed Behavioral Model”, Computer Graphics, Vol. 21, No. 4 (Proceedings of ACM SIGGRAPH ’87), pp. 25–34, 1987.

Effects of Body Size on Autonomous Specialization and Congestion of Robotic Swarms

Motoaki Hiraga [†] Kazuhiro Ohkura [†]

[†] Hiroshima University, 1-4-1, Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan
{hiraga,ohkura}@ohk.hiroshima-u.ac.jp

Abstract: This paper focuses on the effect of the congestion on the swarm performance by considering the number of robots and the size of the robot. The experiments were conducted in computer simulations and carried out by applying evolutionary robotics approach in a path formation task. The autonomous specialization of the robotic swarm has emerged in situations with moderate congestion. In addition, this paper shows that not only the number of robots but also the size of the robot is the essential feature to discuss the relationship between the swarm performance and the congestion.

Keywords: Swarm robotics, Evolutionary robotics, Congestion, Autonomous specialization

1. Introduction

Swarm robotics is the study of a large group of autonomous robots that emerges collective behavior without relying on any form of centralized control [1, 2]. It takes inspiration from biological swarms, such as flocks of birds, schools of birds, and colonies of ants, which perform a collective behavior that is beyond the capability of a single individual. Robotic swarms are expected to have the following three system-level properties; (i) robustness for operating despite failures in the individuals or disturbances in the environment, (ii) flexibility for generating modularized solutions for different tasks, and (iii) scalability for operating under a wide range of group sizes. These three properties are realized by a robotic swarm when designed with high redundancy.

In situations where many embodied robots gather in a limited area, however, robots tend to interfere with each other, which decreases the performance of the individual robot [3, 4, 5]. This congestion is a crucial issue for a robotic swarm in tasks, where robots are required to move toward the same target, e.g., foraging or navigation tasks [6]. Despite the importance of handling congestion in a robotic swarm, few studies have dealt with this problem. Moreover, these studies only discuss the relationship between swarm performance and the number of robots [3, 4, 5].

This paper focuses on the effect of the congestion on the swarm performance by considering not only the number of robots but also the size of the robot. The controller of robots is designed with an evolutionary robotics approach [7], which automatically develops the controller via an evolutionary algorithm inspired by the Darwinian principle of *survival of the fittest*. This approach has fewer restrictions on the acquirable behavior of the robotic swarm and has a possibility of exhibiting behavior that is difficult to design manually. The swarm

performance is evaluated in a path formation task [8, 9], that aims to develop a path of robots and navigate between two landmarks. The experiments in computer simulations show both the swarm and the robot size affect the behavior of the robotic swarm.

The rest of this paper is organized as follows. Section 2 describes the task addressed in this paper and the simulation models for the experiments. Section 3 presents our evolutionary robotics approach. Section 4 shows the results obtained in the experiments, and Section 5 discusses the results. Finally, conclusions and future work are summarized in Section 6.

2. Problem Settings

The path formation task is one of the fundamental tasks addressed in the study of swarm robotics [8, 9]. In this task, the robotic swarm is to develop a collective path of robots and navigate between two landmarks. The experiments are carried out in computer simulations¹. The rest of this section describes the simulated environment and robots applied in the experiments.

2.1 Environment

The snapshot of the environment is shown in Fig. 1. The environment is a square-shaped arena surrounded by walls. Two landmarks are placed diagonally in the environment. Each landmark has RGB LEDs in the center and a target area with a radius of 0.5 m. A robot is considered to have arrived at the landmark when the robot travels inside the corresponding target area.

¹The experiments are conducted with the *Box2D* physics engine, available at <http://box2d.org> (accessed September 25, 2018).

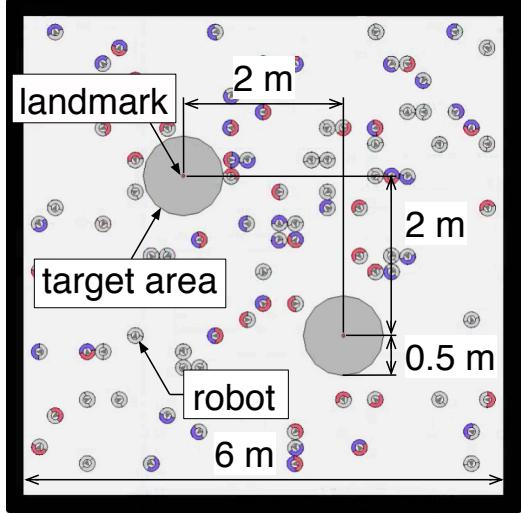


Fig. 1: Snapshot of the environment in the computer simulations. An initial state of the robots positioned randomly in the environment.

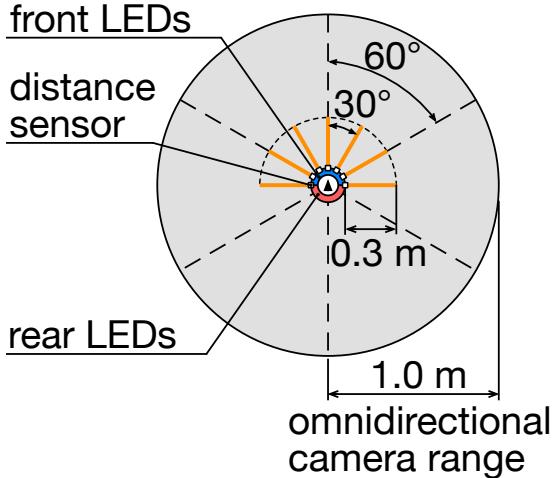


Fig. 2: Configuration of the robot. Distance sensors are attached to the front side of the robot with a 30° interval. The vision of the omnidirectional camera is divided into six circular sectors with a central angle of 60° .

2.2 Robot

The robot modeled in the simulations is illustrated in Fig. 2. The robot is equipped with seven distance sensors, a ground sensor, an omnidirectional camera, and RGB LEDs. Distance sensors are attached to the front side of the robot, as shown in Fig. 2. The distance sensor detects walls and other robots within the sensor range. The ground sensor is attached underneath the robot, which detects whether the robot is inside or outside a target area. The omnidirectional camera allows the robot to detect colored LEDs within the sensor range.

The vision of the omnidirectional camera is coarse-grained; the visual input is divided into six sections as shown in Fig. 2. The omnidirectional camera only detects the pres-

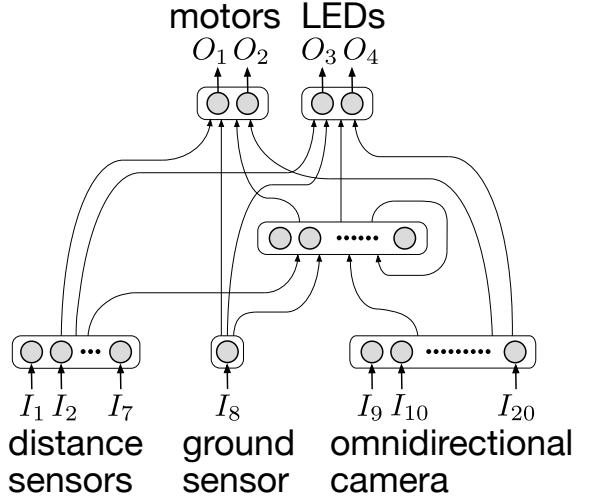


Fig. 3: Structure of the robot's controller. The controller is represented by the recurrent neural network with ten hidden neurons.

ence of colored LED lights for each section. The robots emit blue lights from the front and red lights from the rear. The LEDs can be turned on and off independently according to the outputs of the controller. The LEDs in the center of both landmarks always emit the red color, which is the same color as the rear LED lights of a robot.

3. Methods

The evolutionary robotics approach is a promising method to design controllers for a robotic swarm. Typically, the evolutionary robotics approach applies *evolving artificial neural networks* [10], also known as *neuroevolution* [11], to develop controllers that are represented by artificial neural networks. An evolutionary algorithm evaluates and optimizes the robot controllers based on a predefined fitness function, which indicates the achievement of the task. The following part of this section describes the evolutionary robotics approach applied in this paper.

3.1 Controller

The controller of the robot is represented by a recurrent neural network, as shown in Fig. 3. The input layer is composed of twenty neurons; seven neurons from the distance sensors, one neuron from the ground sensor, and twelve neurons from the omnidirectional camera. The hidden layer is composed of ten neurons with recurrent connections including self-connections. The output layer is composed of four neurons; two neurons for controlling the motors and two neurons for controlling the activation of the LEDs. The output value from the k th neuron in the hidden layer $H_k(\tau)$ is up-

Table. 1: Parameter settings for the (μ, λ) evolution strategy.

Parameter	Value
Number of parents μ	30
Number of offspring λ	200
Initial mutation step size	0.05
Mutation step size	$\in [0.00001, 0.15]$

dated with the following equations:

$$H_k(\tau) = \sigma_1 \left(\sum_i w_{ik}^{IH} I_i(\tau - 1) + \sum_j w_{jk}^{HH} H_j(\tau - 1) \right), \quad (1)$$

$$\sigma_1(x) = \frac{2}{1 + e^{-x}} - 1,$$

where $I_i(\tau - 1)$ is the input value from the i th neuron in the input layer at time $\tau - 1$, $H_j(\tau - 1)$ is the output value from the j th neuron in the hidden layer at time $\tau - 1$, w_{ik}^{IH} is the synaptic weight connected from the i th input neuron to the k th hidden neuron, and w_{jk}^{HH} is the synaptic weight connected from the j th hidden neuron to the k th hidden neuron. The output value from the k th neuron in the output layer $O_k(\tau)$ is updated with the following equations:

$$O_k(\tau) = \sigma_2 \left(\sum_i w_{ik}^{IO} I_i(\tau - 1) + \sum_j w_{jk}^{HO} H_j(\tau - 1) \right), \quad (2)$$

$$\sigma_2(x) = \frac{1}{1 + e^{-x}},$$

where $I_i(\tau - 1)$ is the input value from the i th neuron in the input layer at time $\tau - 1$, $H_j(\tau - 1)$ is the output value from the j th neuron in the hidden layer at time $\tau - 1$, w_{ik}^{IO} is the synaptic weight connected from the i th input neuron to the k th output neuron, and w_{jk}^{HO} is the synaptic weight connected from the j th hidden neuron to the k th output neuron. Two different sigmoid activation function σ_1 and σ_2 are employed to scale the output value of the hidden neuron H_k in the range $[-1, 1]$ and the output value of the output neuron O_k in the range $[0, 1]$. All synaptic weights take real values in the range of $[-1, 1]$. Further details of the robot and the controller can be found in [12, 13].

3.2 Evolutionary algorithm

The (μ, λ) evolution strategy [14, 15] is employed for an evolutionary algorithm. Table 1 shows the parameter settings of the (μ, λ) evolution strategy. The synaptic weights of the controller are optimized via the evolutionary algorithm. The evolutionary process lasts 1000 generations with the zeroth generation of a randomly generated population.

3.3 Fitness function

The controller is evaluated based on the performance of the robotic swarm in the path formation task. A copy of the controller is implemented to N robots and evaluated for $M = 3$ independent trials. Each trial lasts 7200 time steps. Robots can move freely without evaluation during the first 1200 time steps. Subsequently, the individual fitness f_n , which is the fitness of the n th robot, is updated every time step during the remaining 6000 time steps by the following equation:

$$f_n(t) = f_n(t-1) + \begin{cases} 1 & \text{if robot } n \text{ enters} \\ & \text{a new target area,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This equation indicates that the fitness f_n equates to the number of times the n th robot entered a target area that is different from the one previously visited. The comprehensive fitness of the controller F is calculated by the following equations:

$$F = \frac{1}{M} \sum_{m=1}^M F_m, \quad F_m = \frac{1}{N} \sum_{n=1}^N f_n, \quad (4)$$

where M is the total number of trials and F_m is the fitness of the m th trial which equates to the mean value of f_n over the number of robots N .

4. Results

This paper focuses on the effect of the congestion on the swarm performance from both the number of robots and the size of the robot. Therefore, the path formation was performed in the following settings:

- The swarm size $N = 10, 25, 50, 75$, and 100 robots.
- The robot size with 0.2 m and 0.4 m in diameter.

Five independent evolutionary trials were executed for each pair of settings with a different random seed. At the end of the evolutionary trial, the synaptic weights that have obtained the highest fitness among the last 100 generations was selected and re-evaluated for $M = 100$ trials. The best synaptic weights in this re-evaluation was used for the further analysis.

The results of the re-evaluation using the best synaptic weights are shown in Fig. 4. In most cases, the robots with the diameter of 0.2 m perform better than those with the 0.4 m. There was no significant difference between the performance in the robotic swarm with the diameter of 0.2 m and 0.4 m in $N = 10$ robots (Mann-Whitney U test, p -value > 0.05). This is because the swarm size with $N = 10$ robots is insufficient to form a path regardless of the robot size. In cases with $N = 25, 50, 75$, and 100 robots, the performance with the 0.2 m diameter was significantly higher than those of 0.4 m diameter (p -value < 0.001). The congestion is more likely to occur in the larger robot size because fewer robots can enter the target area at the same time.

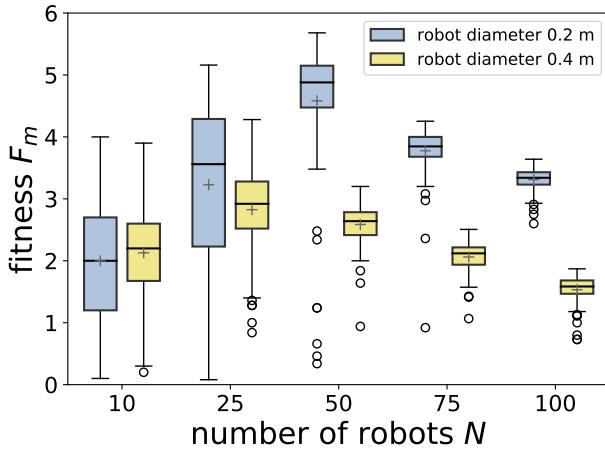


Fig. 4: Box plots of the fitness F_m using the best-evolved weights over $M = 100$ trials for each swarm size. For each number of robot N , the left box plot refers to the robot size with the diameter of 0.2 m and the right box plot refers to the diameter of 0.4 m.

In the robotic swarm with the 0.2 m robot diameter, the highest fitness was scored with $N = 50$ robots. Figs. 5, 6, and 7 show examples of behavior that was observed using $N = 25, 50$, and 100 robots, respectively. The robotic swarm with $N = 25$ robots was enough to generate a path, as shown in Fig. 5; however, it requires more time to aggregate and sometimes fails to form a path. When N is larger than 50 robots, the robotic swarm emerges autonomous specialization; i.e., the robots traveling the inner side of the path activate the LEDs, while those in the outer side deactivate them. This specialization becomes more pronounced when the number of robots increases (see also Figs. 6 and 7). Further discussion on the behavior that was observed using the robots with the 0.2 m diameter can be found in [12, 13].

As for the robots with the diameter of 0.4 m, the highest fitness was scored with $N = 25$ robots. Figs. 8, 9, and 10 show examples of observed behavior with the diameter of 0.4 m using $N = 25, 50$, and 100 robots, respectively. The robotic swarm emerges the autonomous specialization in $N = 50$ robots, which is similar to the one observed in larger swarm sizes with 0.2 m diameter. On the other hand, this specialization will fade away with an increase in the number of robots (see also Fig. 10). Further discussion of the behavior and its analysis is in the next section.

5. Discussion

The activation rates of the LEDs are calculated to discuss the behavior of the robotic swarm. For each robot, the activation rates of the LEDs during 1200 to 7200 time steps is calculated

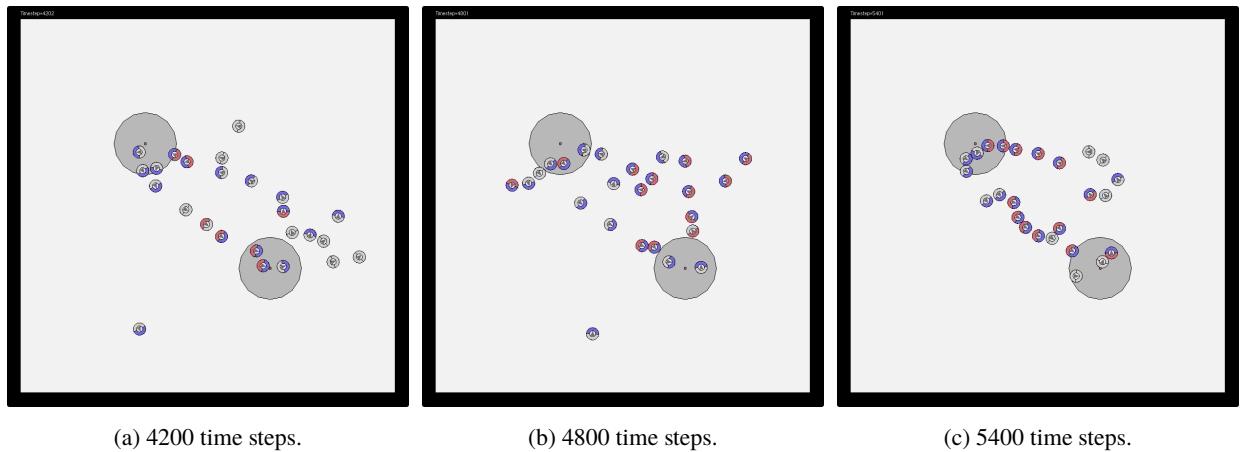
by the following equation:

$$\gamma_{\text{front/rear}} = \frac{\tau_{\text{front/rear}}}{T}, \quad (5)$$

where $\gamma_{\text{front/rear}}$ is the activation rate of the front or rear LEDs, $\tau_{\text{front/rear}}$ is the total time steps of the robot activating the front/rear LEDs during 1200 to 7200 time steps, and $T (= 7200 - 1200 = 6000)$ is the total time steps. The activation rates are calculated based on the behavior observed with $N = 25, 50$, and 100 robots. The scatter plots of the front versus rear LEDs activation rate using the robots with the diameter of 0.2 m and 0.4 m are shown in Fig. 11 and 12, respectively. In addition to the activation rates, the color in Fig. 11 and 12 shows the individual fitness f_n in Eq.(3) during 1200 to 7200 time steps.

The results using robots with the diameter of 0.2 m shows that the specialization emerges as the number of robots increases. As can be seen from Fig. 11(a), the distribution of the activation rates is centered around $\gamma_{\text{front}} = 0.8$ and $\gamma_{\text{rear}} = 0.5$. This implies that all robots have a similar strategy on activating LEDs, i.e., the specialization has not emerged in 25 robots. The distribution of the activation rates spreads with the increase in the number of robots and show a positive correlation between the two activation rates, as can be seen from Figs. 11(b) and (c). Moreover, almost all of the robots obtained relatively high individual fitness f_n in 50 robots, as shown in Fig. 11(b). In 100 robots, however, the robots with lower activation rates have lower individual fitness f_n , as shown in Fig. 11(c). The robots traveling the outer side of the path are more insufficient to perform the task, which leads to lower individual fitness f_n . Therefore, more robots were traveling the outer side of the path and deactivating the LEDs in 100 robots. These results show that the specialization becomes more pronounced as the number of robots increases.

In contrast to the results using robots with the diameter of 0.2 m, the robots with 0.4 m diameter have lower individual fitness f_n because of the congestion, as shown in Fig. 12. The result in 25 robots shows that the distribution of the activation rates with the diameter of 0.4 m is more spread out than that of 0.2 m, as shown in Fig. 12(a). This indicates that the specialization has already started to emerge in 25 robots with the diameter of 0.4 m. There is also a positive correlation in 50 robots, as shown in Fig. 12(b); however, the robots with higher activation rates scored lower individual fitness f_n . This is because of the congestion near the landmarks; i.e., the robots traveling inside the path were pushed much further inward, and consequently, those robots fail to enter the target area. In 100 robots, the positive correlation between the two activation rates has disappeared, as shown in Fig. 12(c). From this result, it can be assumed that the robotic swarm will not emerge the specialization in heavily congested situations. Thus, it can be concluded that the size of the robot is also the essential

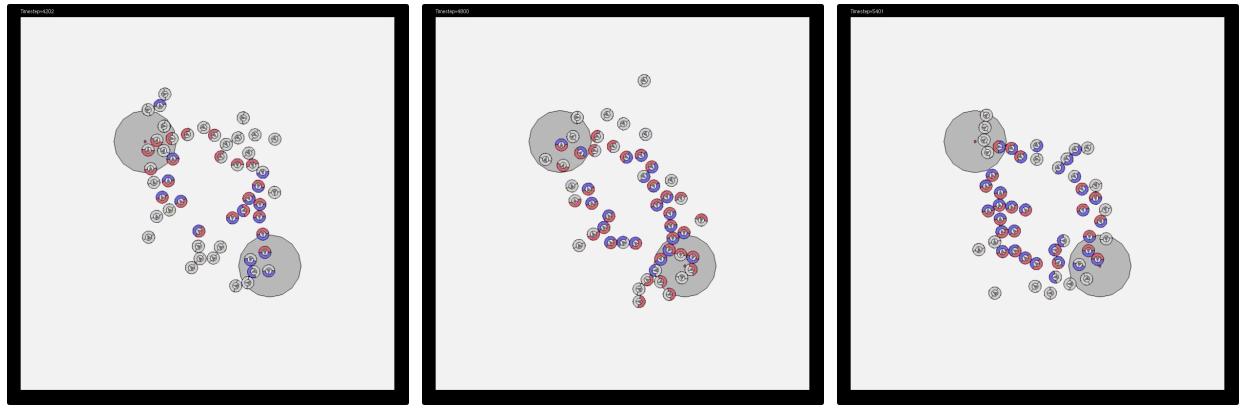


(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 5: Snapshot of behavior observed using 25 robots with the diameter of 0.2 m.

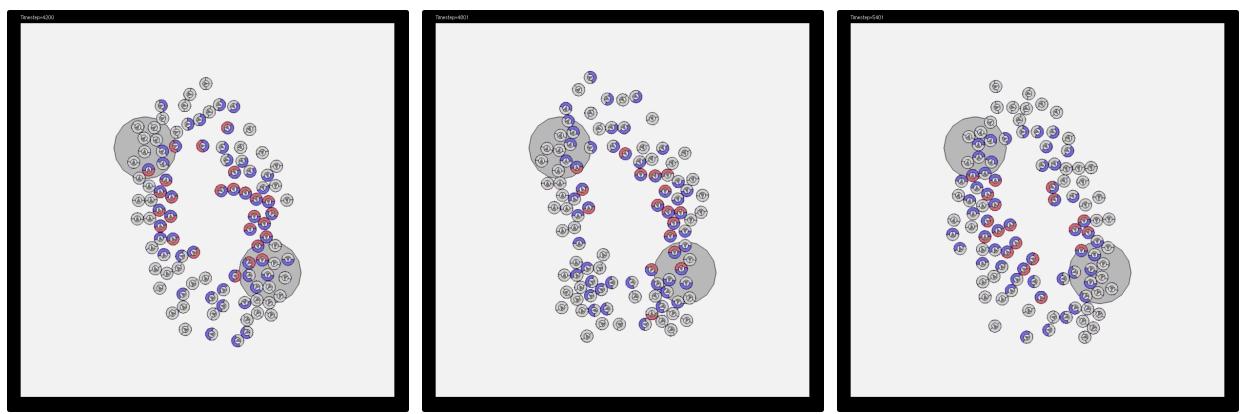


(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 6: Snapshot of behavior observed using 50 robots with the diameter of 0.2 m.

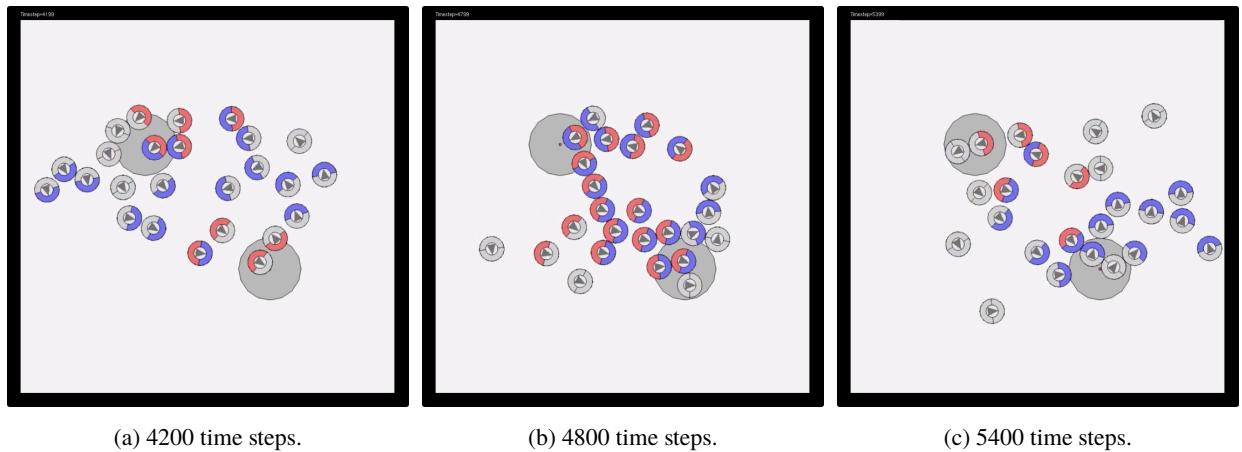


(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 7: Snapshot of behavior observed using 100 robots with the diameter of 0.2 m.

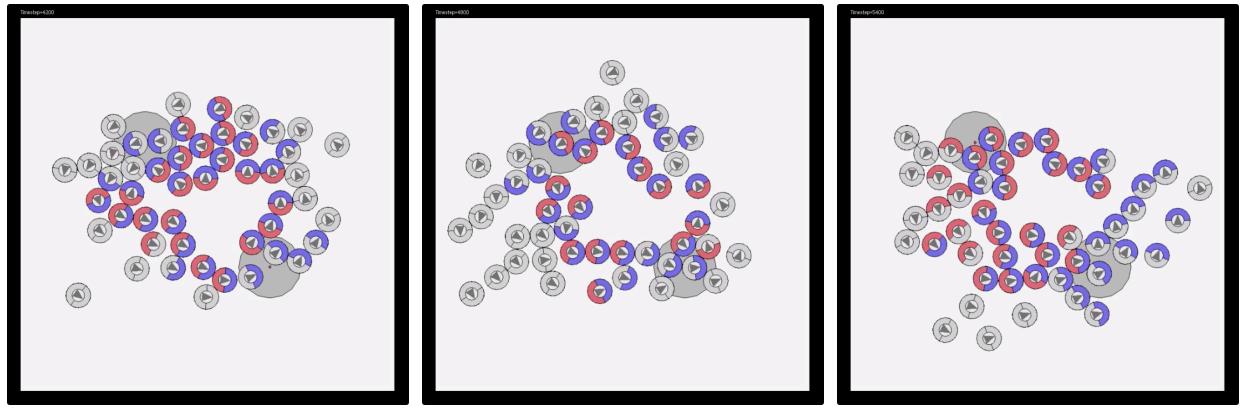


(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 8: Snapshot of behavior observed using 25 robots with the diameter of 0.4 m.

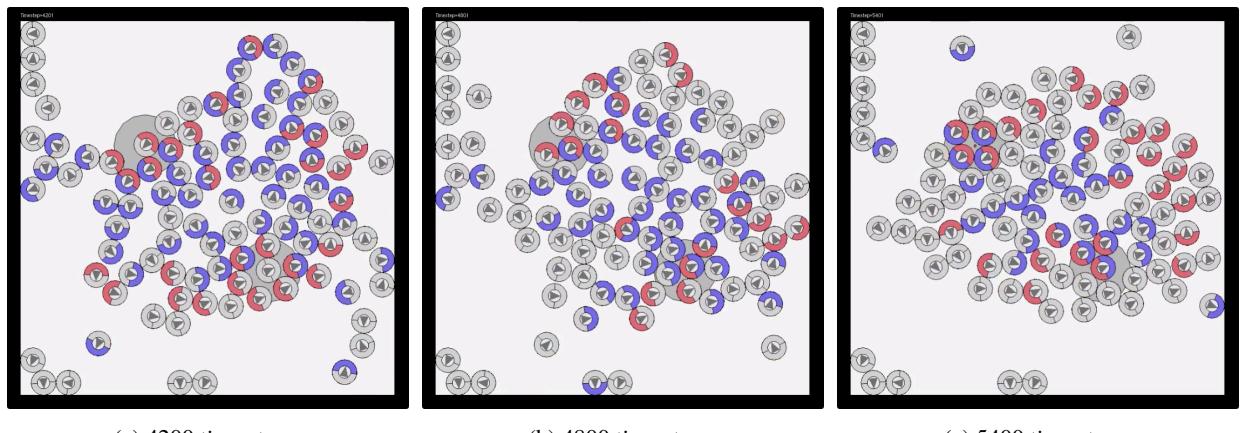


(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 9: Snapshot of behavior observed using 50 robots with the diameter of 0.4 m.



(a) 4200 time steps.

(b) 4800 time steps.

(c) 5400 time steps.

Fig. 10: Snapshot of behavior observed using 100 robots with the diameter of 0.4 m.

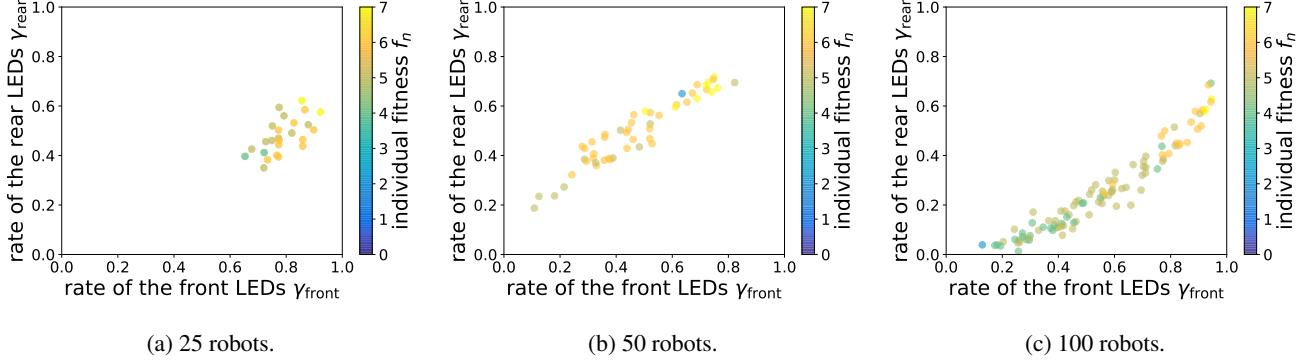


Fig. 11: Scatter plots of the activation rate of the LEDs during 1200–7200 time steps with the robot diameter of 0.2 m. Each point indicates the robot with corresponding activation rates. The color of the point shows the fitness of the individual robot f_n .

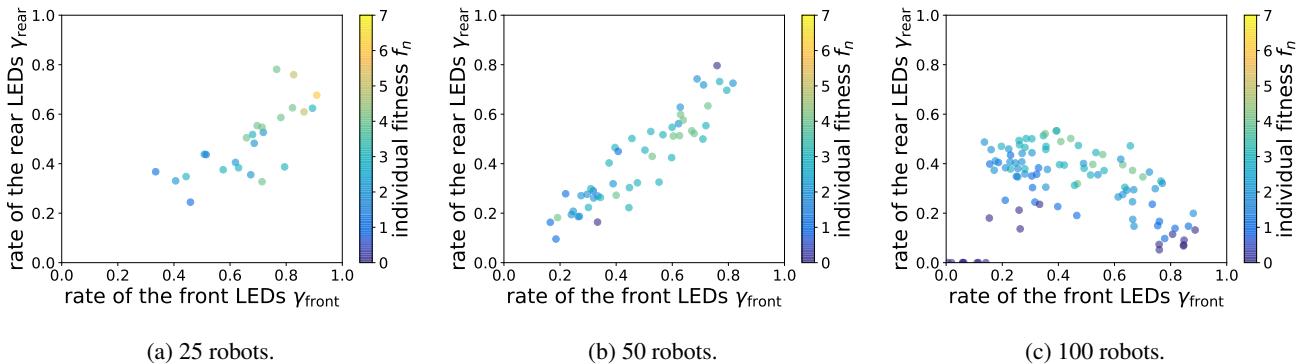


Fig. 12: Scatter plots of the activation rate of the LEDs during 1200–7200 time steps with the robot diameter of 0.4 m.

feature to discuss the relationship between specialization and congestion.

6. Conclusions

The high redundancy is a fundamental characteristic in robotic swarms; however, few studies have discussed the relationship between swarm performance and congestion. This paper focused on the effect of the congestion on the swarm performance by considering the number of robots and the size of the robot. The experiments were conducted in simulations and carried out by applying evolutionary robotics approach in a path formation task. The results show that not only the number of robots but also the size of the robot is the essential feature to discuss the relationship between swarm performance and congestion.

In future work, we are planning to design metrics that define congestion of robotic swarms. As we discussed in this paper, the degree of congestion will affect the behavior of robotic swarms. The metrics for congestion will be a useful tool to design and analyze a robotic swarm. At the same time, we plan to explore the effect of the congestion on swarm performance in different tasks and settings.

References

- [1] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” *Swarm Robotics, Lecture Notes in Computer Science*, Vol. 3342, pp. 10–20, 2005.
- [2] M. Dorigo, M. Birattari, and M. Brambilla, “Swarm robotics,” *Scholarpedia*, Vol. 9, No. 1, p. 1463, 2014.
- [3] K. Lerman and A. Galstyan, “Mathematical model of foraging in a group of robots: Effect of interference,” *Autonomous Robots*, Vol. 13, No. 2, pp. 127–141, 2002.
- [4] H. Hamann, “Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance,” *Swarm Intelligence*, Vol. 7, No. 2-3, pp. 145–172, 2013.
- [5] H. Hamann, *Swarm Robotics: A Formal Approach*. Springer, 2018.
- [6] L. S. Marcolino, Y. T. dos Passos, Á. A. F. de Souza, A. dos Santos Rodrigues, and L. Chaimowicz, “Avoiding target congestion on the navigation of robotic swarms,” *Autonomous Robots*, pp. 1–24, 2016.
- [7] S. Nolfi and D. Floreano, *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.
- [8] V. Sperati, V. Trianni, and S. Nolfi, “Self-organised path formation in a swarm of robots.” *Swarm Intelligence*, Vol. 5, No. 2, pp. 97–119, 2011.
- [9] L. Bayindir, “A review of swarm robotics tasks,” *Neurocomputing*, Vol. 172, pp. 292–321, 2016.

- [10] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, Vol. 87, No. 9, pp. 1423–1447, 1999.
- [11] D. Floreano, P. Dürr, and C. Mattiussi, “Neuroevolution: from architectures to learning,” *Evolutionary Intelligence*, Vol. 1, No. 1, pp. 47–62, 2008.
- [12] M. Hiraga, T. Yasuda, and K. Ohkura, “Evolutionary acquisition of autonomous specialization in a path-formation task of a robotic swarm,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 22, No. 5, pp. 621–628, 2018.
- [13] M. Hiraga, Y. Wei, T. Yasuda, and K. Ohkura, “Evolving autonomous specialization in congested path formation task of robotic swarms,” *Artificial Life and Robotics*, in press.
- [14] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies: A comprehensive introduction,” *Natural computing*, Vol. 1, No. 1, pp. 3–52, 2002.
- [15] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2003.

The Novel Interactive Picture Book System: IOΞK

Saya Fujino Naoki Mori

[‡] Osaka Prefecture University, Osaka, Japan
fujino@ss.cs.osakafu-u.ac.jp

Abstract: Understanding human emotion and “Kansei” is one of the most interesting and important issues in the artificial intelligence field. In this study, we proposed the novel interactive picture book utilizing deep learning technique called Interactive digital picture book with Observing \exists Kansei \in User: IOΞK). The effectiveness of IOΞK is confirmed by computer simulations taking an original picture book written by the author as an example. We also show the analysis results of distributed expression of existing picture books to improve the usability of IOΞK.

Keywords: Neural Network, Picture book, Machine Learning, Kansei Engineering

1. Introduction

Recently, artificial intelligence (AI) has attracted attention as an important fundamental technology in society, because of its remarkably high performance. For example, Alpha GO [1] which has won one of the world’s best player of Go, and Watson [2] which solves quizzes like human beings, AI have also been reported that it is superior than human beings in a several research field. However, it is extremely difficult for AI to understand human emotion or “Kansei”. Given the fact, we considered about auto-generation of picture book by AI which can touch human’s heart and proposed the novel interactive picture book called Interactive digital picture book with Observing \exists Kansei \in User (IOΞK).

IOΞK is the system which focus on “Kansei” in users and make from some images and stories based on the “Kansei”. In this research, we showed its effectiveness by the example of original picture book generation by author and the numerical experiments of analysis of published picture books. Furthermore, the Japanese national railway 103 series that authors have a strong interest for, has inspired the name of IOΞK. Fig.1 shows a picture of the 103 series taken by author. The name of the proposed system is associated by the Japanese people with trains for which they have a deep appreciation because of their usefulness in the society, this is what I want to achieve with the developed system. So, we call proposed system IOΞK as “hyaku san kei”.

Picture books are consist of pictures and sentence. In machine learning fields such as image recognition, deep learning has been one of the key techniques in recognizing images because of its remarkably high performance. One of the most powerful methods to solve image recognition problems is using a deep convolutional neural network (DCNN)[3]. We utilize the DCNN for understanding pictures in IOΞK. Besides, to understand sentences, we apply doc2vec[4] to IOΞK.



Fig. 1: the 103 series: Japanese Hanwa line

To obtain the effective DCNN structure, we also proposed a novel evolutionary approach called evoDL[5].

To show the effectiveness of IOΞK, computer simulations are carried out taking an original picture book written by the first author as an example. We also show the analysis results of distributed expression of existing picture books to improve the usability of IOΞK.

2. Related works

In this section, we state that the elemental technology of AI about image processing and natural language processing, and previous studies on generating picture books and stories.

2.1 Deep Convolutional Neural Network

In this section, we introduce the deep convolutional neural network (DCNN). DCNN is one of the deep learning methods and widely used in the image recognition field. Fig. 2 shows the architecture of AlexNet which one of the most famous

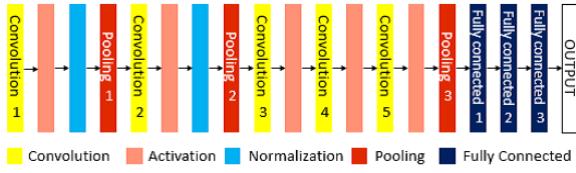


Fig. 2: The architecture of AlexNet

DCNN. The structure of AlexNet is simpler than other DCNNs, however the performance is as strong as others. DCNN is composed of one or more convolutional layers, pooling layers, and fully connected layers. Fully connected layers follow convolutional and pooling layers. DCNN uses fewer learnable parameters. Convolutional layers are the core building blocks and its parameters consist of a set of learnable filters. Input image is passed through convolutional layers and dot products are computed between the entries of the filter and the image at any given position.

2.2 Doc2Vec

Doc2vec[4] is document vector generation method expanding the concept of word2vec to document. There are also two types of learning models in doc2vec, namely distributed bag-of-words (DBOW) model and distributed memory (DM) model. DBOW model is the learning model that corresponding to skip-gram model in word2vec. In DBOW model, doc2vec learns to predict word vector of each word contained in document from document vector. On the other hand, DM model is the learning model that corresponding to CBOW model in word2vec. In DM model, it learns to predict the word vector of focus word in document from word vectors of context words and document vector.

2.3 Tweet2Vec

Tweet2vec[6] is one of the character composition model which finds vectorspace representations of whole tweets. Fig. 3 shows Tweet2vec model. It uses a similar structure to the C2W[7] model, with Long Short Term Memory (LSTM)[8] units replaced with Bi-directional Gated Recurrent Unit (Bi-GRU)[9].

The input to the network is defined by an alphabet of characters. The input tweet is broken into a stream of characters.

3. IOEK

IOEK is the system which focus on “Kansei” in users and make from some images and stories based on the Kansei. It is shown that the useful parts of making picture book which analyses the illustration taste and literary style as user’s Kan-

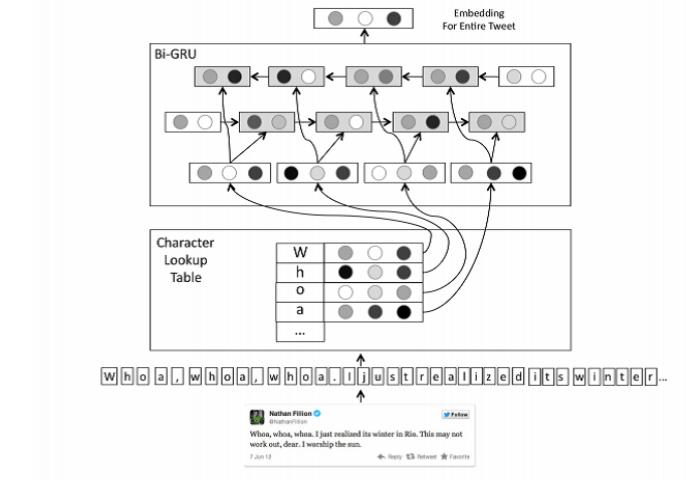


Fig. 3: Outline of Tweet2Vec[6]

sei. The proposed method IOEK has image recognition module, literally style correction module and display module.

3.1 Image recognition module

This module is to recognize illustration taste/touch in a picture book. When we make the picture book, we have to draw and write everything by hand and it involves considerable effort. Especially, drawing pictures requires time and effort to draw characters, in addition to backgrounds and foregrounds. When user draws the main characters, the system made by this research can select the closest illustrations to their touch and they don’t have to draw other objects. The illustrations are gathered from free illustration dataset. To select the illustrations, we adopted DCNN acquired by evolutionary method in the section 4.

3.2 Literally style correction module

The module can choose the user’s most favorite expression in the same meaning words. Because the module is specialized in Japanese, we skip the rest details.

3.3 Display module

The display module will be able to added objects by user and interaction about the picture book with the system. In this paper, we implemented the function which displays the sentence and illustrations made by the system.

4. Evolutionary approach for DCNNs

In this section, we shows a novel evolutionary approach for finding a good deep learning model evoDL[5]. It is needed by image recognition module of IOEK.

The DCNN is one of the deep learning methods and is widely used in the image recognition field. In this study, we

Table. 1: Gene of individuals in GA

Design Variables	Allele
The number of filters (NF)	16, 32, 64
Filter size (FS)	3, 5, 7, 9, 11
Pooling size (PS)	3, 5, 7
NL1	128, 256, 512, 1024
NL2	16, 32, 64, 128
Batch size (BS)	3, 10, 30
Activation function with ReLU (Re)	1 (use), 0 (not use)

N11: The number of nodes in fully connected layer 1

N12: The number of nodes in fully connected layer 2

used the architecture of AlexNet[3] as the DCNN. AlexNet is one of the most popular architectures of a DCNN. It contains eight learned layers: the first five are convolutional and the remaining three are fully connected. Using a softmax function, the last fully connected layer produces an output consisting of class labels. Although the DCNN is a powerful method, tuning hyperparameters involves considerable effort. To solve this problem, we propose a novel evolutionary approach for finding a good deep learning model evoDL[5]. We adopted the GA as the optimization technique for evoDL.

4.1 Setting of evoDL

In this study, we focused only on hyperparameters represented by an integer or a boolean value, such as the number of filters, the filter size, and the max pooling size, using an activation function. We used the conventional GA framework except for the representation of the genotype. The uniform crossover, uniformly distributed mutation, and tournament selection are applied in this respective order to the population of the GA. The number of training data is 160; the total number of positive data is 80 drawn the characters and background illustrations by author, the number of negative data is 80 chosen some illustrations which author don't like from the free dataset. Then, we prepared the each number of 1440 by using data augmentation. Table 1 shows the details of alleles of the genotype in GA. Fig. 4 shows the example of the chromosome of an individual. Each string in the locus is related to strings in Table 1. The basic structure of the DCNN is fixed according to AlexNet. In Fig. 4, the seven genes “Re” represent the flag of ReLU functions, which are set just after three convolutional layers and two fully connected layers. If gene Re is 1, the corresponding ReLU function is set after the target layer.

4.2 Fitness function

In this study, the fitness function $F(s)$ of evoDL is the negative value of the sum of loss in k -fold cross-validation using training data, where s is an individual related to a certain

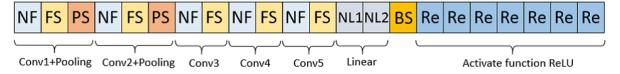


Fig. 4: Example of genotype

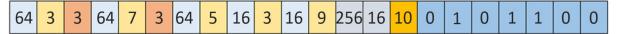


Fig. 5: Genotype of best individuals

DCNN.

In evoDL, fitness function $F(s)$ is represented as follows:

$$F(s) = - \sum_{i=1}^k f_{loss_i}(s) \quad (1)$$

where s denotes an individual and $f_{loss_i}(s)$ is the loss of k -fold cross-validation after i epochs are calculated by cross entropy. In this case, the maximum fitness value is 0. Other types of fitness functions such as the sum of accuracy or the function of considering complexity of network are available in the proposed evoDL.

5. Experiments and Results

Here, we show the experimental results of IOEK.

5.1 Experiment1: the network evolution

To make picture book scene by IOEK, the DCNN which can understand the touch of user pictures. We obtained that DCNN by evoDL. Fig. 5 shows the elite individual genotype obtained by evoDL shown in section 4.

ReLU functions are always used except the second convolutional layer. This results shows that the ReLU function is very important to obtain effective DCNN. In obtained network, the number of filters and the number of nodes in fully connected layer is smaller and simpler than the ones for AlexNet. DCNN learned user's illustrations and classified some objects whether that are similar to the user's ones or not from free illustration dataset. The objects were classified as user's style. Table 2 shows accuracy and Fig. 6 shows the illustrations selected by DCNN.

5.2 Experiment2: the decoration of picture book scene by IOEK

In experiment2, we used the method described in the section 3.1. We regard the author as the one user in this section.

We define two states of picture book illustration; A: the scene drawn by user before decorated, B: the scene decorated by the selected illustration objects. Fig. 7-8 and Fig. 9-10 show scene1 and scene2, respectively. Fig. 11 shows a scene

Table. 2: The accuracy of the selects of picture book objects

Training accuracy	1.00
Test accuracy	0.90
Base line	0.62

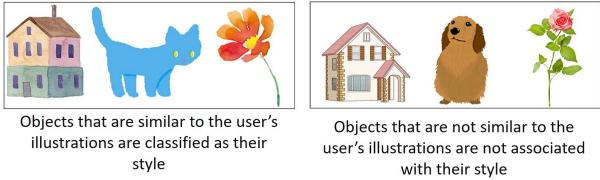


Fig. 6: The example of the selects of picture book objects

of the picture book. we displays a part of them because of space limitations. Fig. 11 shows that IOEK is able to decorate the picture with a natural feeling.



Fig. 7: scene 1-A before decorated

5.3 Experiment3 : a literary style collection by IOEK

To correct literally style to user preferences, we apply IOEK to user written document S . First, we selects 11 words in user document S and adding two different words to each position. In this paper, positions and words are decided by user. IOEK selects the document which has the most similar distributed representation of user preferred picture books in 2^{11} documents. The distributed representation is obtained word2vec with using data of 3229 Japanese novels, 1730 cooking recipes, 9941 trip reviews and two user preferred picture books. Cosine similarity was used for measurement,



Fig. 8: scene 1-B after decorated

Table. 3: The cosine similarity between the best document and user preferred picture books

Picture book A	Picture book B	Sum
0.450	0.515	0.965

and sum of cosine similarity between the target document and user preferred picture book A and B are calculated. Table 3 show the result of experiment 3. As a result of confirmation by the user, the obtained sentence appropriately reflects the preference of the user.

6. The distributed representation of picture book

Although picture book is consist of images and sentences, the distributed representation is obtained those data. We proposed the distributed representation of picture book to make a user preferred picture book by IOEK.

In the first step, we consider the accuracy of classifying pages to their corresponding picture books. We tried the identification of title of picture books by using features obtained by DCNN and Tweet2Vec. We prepared seven published picture books and one picture book by author, so the total number of pages is 102. Because Japanese picture books are written in Hiragana, word-based morphological analysis is difficult to apply. Therefore, characters-based Tweet2vec is suitable to analyze picture book.

First, we consider the following two types of feature vector on each page of picture book.

Image feature vector The vector made from the output of 3744 dimensions of last convolutional layer

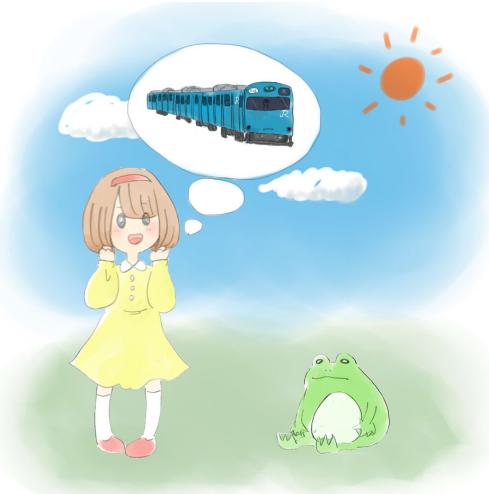


Fig. 9: scene 2-A before decorated



Fig. 10: scene 2-B after decorated

Image and sentence feature vector The Vector which con-

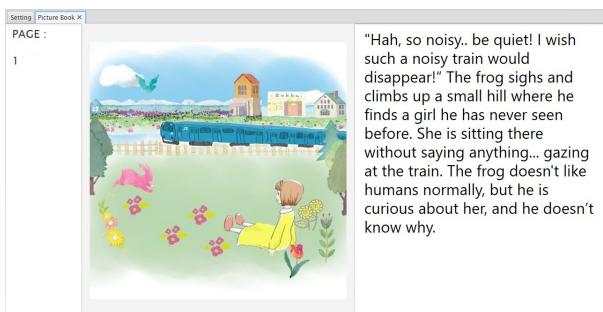


Fig. 11: scene 1

Table. 4: the accuracy of classifying pages of picture books in each class

Image feature vector	0.925
Image and sentence feature vector	1.0

catenates Image feature vector and the output of 100 dimensions of Tweet2Vec

We mapped the two types of feature space on the two dimension space by t-SNE[10] and confirmed that the distance between picture books. Fig. 12 and 13 show the two spatial dimensions made from image feature vector and image + sentence feature vector, respectively. The vertical and horizontal axes on each Fig. express the elements of vectors mapped by t-SNE[11]. Each point shows one page of picture book and each color shows the title of picture book. Fig. 13 shows that the vector made from image and sentence is better than the vector made from only image for clustering. Particularly, the gray points which show the picture book made by author can form an appropriate cluster that does not mix with other clusters. Fig. 4 shows the average accuracy of 5-fold cross validation which classifies the pages of picture books in each class by using support vector machine(SVM). We used the image feature vectors and the image + sentence feature vectors as a feature. When we use only the image feature vectors, we can confirm the high accuracy, 92.5 %. However in the case of using the image + sentence feature vectors, we can gain the highest accuracy, 100 %. The result shows that each picture book has some unique features, and the one made by IOEK has originality.

7. Conclusion and future works

In this study, we considered the interactive picture book system by using AI and proposed the novel interactive picture book utilizing deep learning technique called Interactive digital picture book with Observing \exists Kansei \in User: IOEK). The evoDL is utilized to obtained DCNN which can select user preferred image objects. The literally correction ability of IOEK is also estimated. We also proposed the distributed representation of picture book by DCNN and tweet2vec. Computer simulations show the effectiveness of IOEK.

Important future works are as follows:

- Auto-decorating picture books by selected objects
- Quantitative evaluation of emotion and feeling on users.
- Extend evoDL genotype to be able to represent complex network topology.
- The landscape analysis of search space based on various fitness functions.

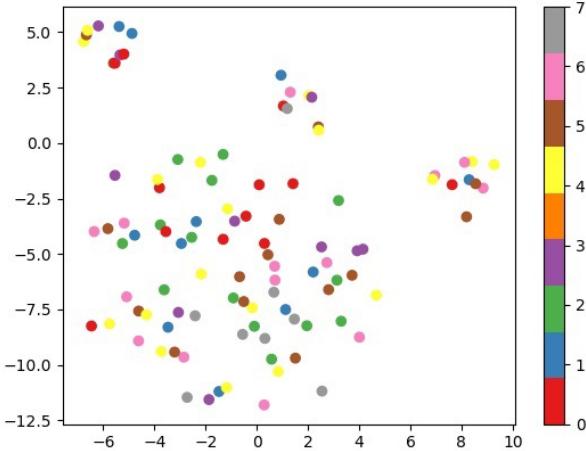


Fig. 12: Image feature map

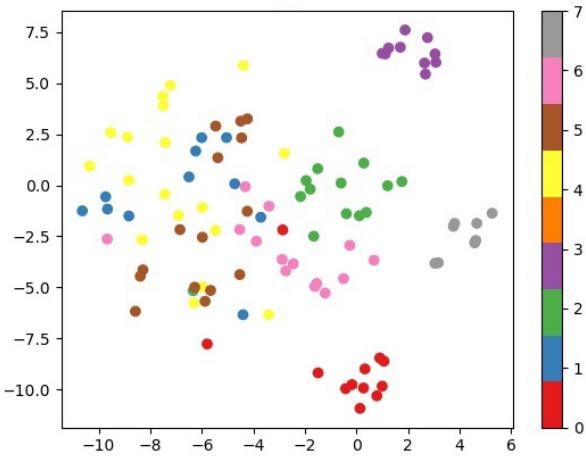


Fig. 13: Image and sentence feature map

- Built the GUI part of IOEK for releasing user application.

Acknowledgment

A part of this work was supported by JSPS KAKENHI Grant, Grant-in-Aid for Scientific Research(C), 26330282. A part of this work was supported by LEAVE A NEST CO.,LTD. A part of this work was supported by a research granted from the Murata Science Foundation.

References

- [1] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, and I. Antonoglou

et al. Mastering the game of go with deep neural networks and tree search. *Nature*, Vol. 529, pp. 484–503, 2016.

- [2] Hiroshi Kanayama and David Ferrucci. Building watson : Can a computer compete against the best human contestants in the tv quiz show? *J. IEICE, Natural Language Understanding and Models of Communication*, Vol. 110, No. 356, p. 89, dec 2010.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pp. 1106–1114. 2012.
- [4] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, Vol. abs/1405.4053, , 2014.
- [5] Saya Fujino, Taichi Hatanaka, Naoki Mori, and Keinosuke Matsumoto. The evolutionary deep learning based on deep convolutional neural network for the anime storyboard recognition. In *Distributed Computing and Artificial Intelligence, 14th International Conference, DCAI 2017, Porto, Portugal, 21-23 June, 2017*, pp. 278–285, 2017.
- [6] Bhuvan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W. Cohen. Tweet2vec: Character-based distributed representations for social media. *CoRR*, Vol. abs/1605.03481, , 2016.
- [7] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. Context-and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 383–392. ACM, 2015.
- [8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2605, 2008.
- [11] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

Prediction of Race Results and Generation of Predicted Articles in Keirin

Takumi Yoshida [†]

Soichiro Yokoyama [†]

Tomohisa Yamashita [†]

Hidenori Kawamura [†]

[†] Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan
yoshida@complex.ist.hokudai.ac.jp

Abstract: Proactive information dissemination is necessary to revitalize the Japanese Keirin business. An article describing the prediction of the race result is effective for revitalizing the Keirin business. However, writing articles manually is a heavy burden. This is because, on average, 60 races are held daily, which is considerable. Another reason is that participating cyclists will be decided according to the previous day's race results. Therefore, there is a strong demand for automatic generation of content, including articles. The purpose of this research is to generate articles automatically with regard to the Japanese Keirin. To automatically generate articles, we predict race results using machine learning. Moreover, we generate an article explaining the predicted results using a template.

Keywords: machine learning, neural network, prediction

Introduction

Studies on the automatic generation of articles are actively being conducted. In sports, research has been done to generate abstractive summaries from Japanese tweets [1] and research to generate inning summary based on encoder-decoder model and transfer learning [2]. Moreover, even in fields other than sports, research has been done to generate summary text from time-series numerical data [3]. All above studies have been aiming at reducing the burden of manually generating text.

Japanese Keirin hosts approximately 60 races daily, and articles are manually written for each race. The term "article" here refers to an article describing the prediction of the race result. Moreover, the participating cyclists for the next race are decided according to the race result of the previous day; thus, the burden of generating articles is great. On the contrary, the acquisition of new users is a problem in the Keirin industry, and proactive information dissemination is required. Therefore, this research aims toward the automatic generation of articles in Keirin. To automatically generate articles, we predict the race results using machine learning. In addition, we generate an article explaining the predicted results using the template.

Keirin

Keirin basically involves cycling around a track on a bicycle. It was developed in Japan around 1948 for gambling purposes. In a Keirin race, the cyclists form a column called a "line". This is unique to Japanese Keirin. In many cases,

cyclists belonging to the same area form a line. The order in a line is decided by strength of relationship between cyclists. A cyclist who does not have a partner runs alone. A cyclist (leading cyclist) running at the start of the line can freely decide the speed and the timing of acceleration. However, leading cyclists consume considerable amount of energy because they receive the most wind resistance in the line. A cyclist (following cyclist) who runs behind the leading cyclist can use the front cyclist as a wind shield so as to save his energy. They have the role of helping the leading cyclists by blocking the any cyclists attempting to move up from the rear. The cyclists run with the team until the final goal line; however, toward the end, each cyclist aims at finishing first. The race flow created by a line is unique to Keirin.

Keirin is one of public gambling in Japan. The betting rules are almost the same as that in horse racing, which is a famous mode of public gambling. With regard to horse racing, there are several researches that discuss the prediction of race results using various methods such as neural networks [4], support vector machines [5], logistic regression [6] and fuzzy logic [7]. The difference between Keirin and horse racing is whether it is a Keirin bicycle, the number of participants, the frequency of races; however, the biggest distinguishing factor is the line. Considering the interaction between the cyclists caused by the line is important for predicting the race results in Keirin.

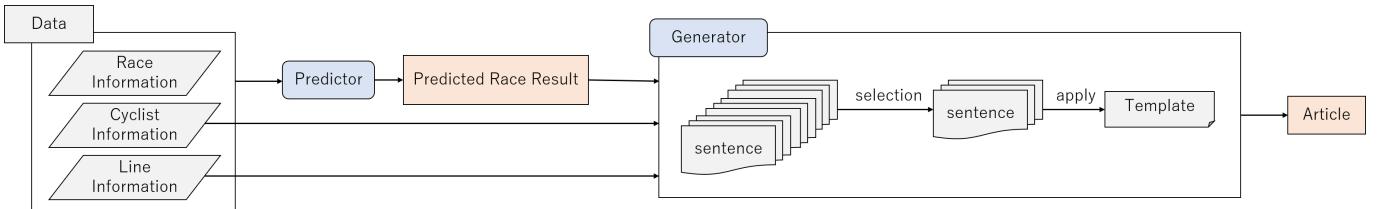


Fig. 1: Article Generation System

Article on Keirin

Existing Articles

Articles are manually written for all races and delivered to users at newspapers and voting sites in Keirin. We demonstrate examples of existing articles on voting sites [8].

- ・ "磯島が駆けて番手の野木が本命 . 鋭さ光るのは丸山だ . 逆転の捲りに一考 . 目標の高鍋次第で大久保、攻め多彩な吉田も怖い . "

(Nogi is the favorite because Isojima, who is Nogi's leading cyclist, is strong. Maruyama is in good condition. There is the possibility that he achieves victory from behind. Okubo and Yoshida are also strong.)

- ・ "持ち前のスピードを全開なら⑤松本 ②蓮井でワンツー . 自在③田中 ⑨米原の捲りや同県⑥山田にマークする④竹内や⑦内村も単級"

(⑤Matsui- ②Hasui are a one–two finish. There is the possibility that ③Tanaka-⑨Yonehara achieve victory from behind. ④Takeuchi who marks ⑥Yamada and ⑦Uchimura are also strong.)

- ・ "藤縄一中澤の近畿勢に期待 . 隅田一高津 , 中部勢の抵抗も見ものだろう . "

(I expect that Fujinawa and Nakazawa, who are Kinki cyclists, win. Sumida and Takatsu, who are Chubu cyclists, may win.)

The existing article contains the following four pieces of information.

- Players expected to be ranked in the top three
- Information on cyclists expected to be ranked in the top three
- Information on line
- Information on race flow

Moreover, existing articles have the following two features.

- Prediction of settlement by line
- Prediction of multiple race flows

It is considered that the prediction of settlement by line is a feature of existing articles because the interaction among

cyclists caused by the line has a considerable influence on the race result. Due to the influence of the line, it is assumed that the race result will differ considerably depending on the race flow; thus, the prediction of multiple race flows is stated in existing articles. From the above statements, it is understood that the line is important for Keirin.

Article Generation Approach

In Figure 1, we show the article generation system. The goal is to generate the article containing the above four pieces of information. However, as there is no data on the race flow currently, in this paper, we aim to generate articles containing the following three pieces of information.

- Players expected to be ranked in the top three
- Information on cyclists expected to be ranked in the top three
- Information on the line

First, we predict the race results using machine learning. The target of prediction are the cyclists who finish first, second, and third. Then, we generate the article using the predicted race result, the cyclist's past record, and line information. We use templates to generate articles containing the above information.

Prediction

To generate articles, we predict the race results using machine learning. The target of prediction are the cyclists who finish first, second, and third. To predict the race result, we propose a two-cyclist model and a nine-cyclist model. We select a neural network as a machine learning method. In this paper, we experimented to compare a two-cyclist model with a nine-cyclist model.

Two-cyclist model

The two-cyclist model is a pair-wise method that predict race results by comparing two cyclists. The flow by which this model predicts the race result is as follows.

- (1) Data of two cyclists are input, and their output values are aggregated for each race.
- (2) Approximately calculate the order probability of the top three cyclists from the aggregated values.
- (3) Select the one with the highest order probability as the prediction result.

input/output

The input of the two-cyclist model is the data of the two cyclists (number i , number j). We denote this input as $x_{i,j}$. This model outputs the probability that the cyclist of number i wins the cyclist of number j . The output layer of this model is a sigmoid function. We define the teacher label $t_{i,j}$ of $x_{i,j}$ in Eq. (1).

$$t_{i,j} = \begin{cases} 0 & place_i > place_j \\ 1 & place_i < place_j \end{cases} \quad (1)$$

where $place_i$ is the place of number i . We correct $y_{i,j}$ with Eq. (2) because $y_{i,j} \neq 1 - y_{j,i}$.

$$y_{i,j}(\text{correction}) = \frac{y_{i,j} + (1 - y_{j,i})}{2} \quad (2)$$

Predict race result

We calculate the order probability of the top three cyclists from aggregated outputs. We approximate the order probability by the product of the following three probabilities.

- Probability that number a takes the first place.
- Conditional probability that number b takes the second place given number a takes the first place.
- Conditional probability that number c takes the third place given number a takes the first place and number b takes the second place.

We show the calculation formula for each probability in Eq (3), (4), (5).

$$p(1\text{st} = a) = \frac{\prod_{j \neq a} y_{a,j}}{\sum_i (\prod_{j \neq i} y_{i,j})} \quad (3)$$

$$p(2\text{nd} = b | 1\text{st} = a) = \frac{\prod_{j \neq a,b} y_{a,j}}{\sum_i (\prod_{j \neq a,i} y_{i,j})} \quad (4)$$

$$p(3\text{rd} = c | 1\text{st} = a, 2\text{nd} = b) = \frac{\prod_{j \neq a,b,c} y_{a,j}}{\sum_i (\prod_{j \neq a,b,i} y_{i,j})} \quad (5)$$

Nine-cyclist model

The nine-cyclist model is a list-wise method that predicts the race results by comparing the nine cyclists. This model learns the order of the top three cyclists directly without incorporating the workings of the two-cyclist model.

input/output

The input of the nine-cyclist model is the data of nine cyclists (number 1, number 2, ..., number 9). The input value without a corresponding cyclist is 0. This model outputs the order probability of the top three cyclists. The output layer of this model is the softmax function. The teacher label t is a one-hot vector representing the order of the top three cyclists. We define t ($t[\text{index}] = 1$) in Eq. (6).

$$\begin{aligned} \text{index} &= (n_1 - 1) \times 56 + (n_2 - 1) \times 7 + (n_3 - 1) \\ &\quad - d_1 - d_2 - d_3 \\ d_1 &= \begin{cases} 0 & n_1 > n_2 \\ 7 & n_1 < n_2 \end{cases}, d_2 = \begin{cases} 0 & n_1 > n_3 \\ 1 & n_1 < n_3 \end{cases}, \\ d_3 &= \begin{cases} 0 & n_2 > n_3 \\ 1 & n_2 < n_3 \end{cases} \end{aligned} \quad (6)$$

where n_i is the cyclist number of i -th place. t is made of 504 dimensions. If race result is 1 – 2 – 3, $t = [1, 0, 0, \dots, 0]$.

Predict race result

This model selects the one with the highest output values as the prediction result.

Experiment

We experimented to compare the two-cyclist model with the nine-cyclist model.

Data

We use the race results for the data held in Japan between 2014 and 2016. The data of years 2014 and 2015 is used as training data and the data of 2016 is used as test data. The training data is for 40,230 races and the test data is for 15,731 races. In this paper, we use the data that satisfies the following conditions.

- Men's race
- No scrambling line
- No dead heat
- No missing data

Features

We use the race feature, the cyclist feature, and the line feature for the neural network input. We show details of each feature in Table 1. Racer features other than cyclist number and age are data for the last four months. The line feature is a feature for predicting the influence of the line. Each model uses different line features that are appropriate for that model. The two-cyclist model uses the cyclist features of the leading cyclist as the line feature. It also uses a feature that indicate

whether the two cyclists to be compared are part of the same line. The nine-cyclist model uses the line vector that indicates the cyclist's line. If the line is 1-9-3, the line vector is $[1, 0, 1, 0, 0, 0, 0, 0, 1]$.

Table. 1: input features

race feature	race grade track length number of cyclists
cyclist feature	cyclist number age race score number of the first place number of the second place number of the third place percentage of the first place percentage of the first or second place number of backstretch leads number of winning tricks
line feature	leading cyclist or not position in line leading cyclist (two-cyclist model) same line or not (two-cyclist model) line vector (nine-cyclist model)

Hyperparameter

We set 20% of training data as validation data and determine the hyperparameter of the neural network via grid search. We demonstrate the search range of hyperparameters in Table 2. We demonstrate the hyperparameters of the two-cyclist model in Table 3 and hyperparameters of the nine-cyclist model in Table 4. In Table 2, 3 and 4, $[2^n, 2^n]$ means that the neural network has two hidden layers and each layer has 2^n units.

Table. 2: search range of hyperparameters (neural network)

hyperparameter	search range
activate function	ReLU
mini batch size	256
drop out	0.5
optimizer	Adam [9]
learning rate	0.001, 0.0001, 0.00001
hidden layer	$[2^n], [2^n, 2^n], [2^n, 2^n, 2^n]$ ($n=7,8,9$)

Table. 3: hyperparameters (two-cyclist model)

hyperparameter	selected
activate function	ReLU
mini batch size	256
drop out	0.5
optimizer	Adam
learning rate	0.0001
hidden layer	$[2^n]$ ($n=8$)
epoch	210

Table. 4: hyperparameters (nine-cyclist model)

hyperparameter	selected
activate function	ReLU
mini batch size	256
drop out	0.5
optimizer	Adam
learning rate	0.0001
hidden layer	$[2^n, 2^n]$ ($n=8$)
epoch	200

Result

We use the accuracy of the order of the top three cyclists and top-K-accuracy as evaluation items. The top-K-accuracy is the proportion of the race that the model predicted a cyclist to attain a certain place within the top K places. We show test results in Table 5. Top 3 (1st, 2nd) denotes the proportion of races wherein the model predicted first and second within the top three.

Table. 5: Test results

order of top 3			
two-cyclist model	4.9 %		
nine-cyclist model	5.6 %		
	top 1 (1st)	top 2 (1st)	top 3 (1st)
two-cyclist model	40.0 %	60.5 %	72.7 %
nine-cyclist model	34.9 %	51.5 %	60.0 %
	top 2 (1st,2nd)	top 3 (1st,2nd)	
two-cyclist model	24.2 %	42.2 %	
nine-cyclist model	23.8 %	36.3 %	
	top 3 (1st,2nd,3rd)		
two-cyclist model	13.2 %		
nine-cyclist model	15.2 %		

Discussion

The nine-cyclist model has a higher prediction accuracy for the top three than the two-cyclist model , the two-cyclist model has higher prediction accuracy for the first and second than the nine-cyclist model. This indicates that it is effective to predict the influence of the line using all cyclists for input. On the contrary, the simple model comparing two cyclists has better accuracy for the prediction of the first and the second place. This indicates that the prediction accuracy for the first place can be increased to 40.0% using these features. In future, we aim to design a network structure that can predict the influence of the line without reducing the high prediction accuracy for the first place.

Article Generation

We generate articles based on the predicted race results. The generator selects sentences to be applied to the template according to the predicted race result and the cyclist's information. The generator generates articles by applying the selected sentences to the template. In the following, we explain the template and the selection of the sentences that apply to the template.

Template

We show the template below.

- <sentence1><cyclist name> が <sentence2>
- <predicted place> 着 <sentence3>

The generator can generate articles by applying arbitrary sentences to this template. If we generate an article by actually applying some sentences to the template, it becomes as follows.

- 前日は 9 着だった②大山が自力で決めて 1 着。
(②Oyama who attained ninth place yesterday is the first place on his own power.)
 - <sentence1> : 前日は 9 着だった
(who was 9th place yesterday)
 - <cyclist name> : ②大山
(②Oyama)
 - <sentence2> : 自力で決めて
(on his own power)
 - <predicted place> : 1
 - <sentence3> : .

In this manner, by selecting a sentence according to the situation, it is possible to generate an appropriate article.

Selection of sentences according to race results

By selecting a sentence according to the line composition of the predicted race result, the generator generates articles containing information about the line. In the case where the predicted first and second place are players of the same line, among others, the aim is to generate articles to explain it.

Selection of sentences according to cyclist information

By selecting a sentence according to cyclist information, the generator generates articles containing cyclist information. We use the following data as cyclist information.

- Number of the first place for the last four months
- Percentage of the first place for the last four months
- Percentage of the first or second place for the last four months
- Age

We set conditions on cyclist information and select sentences according to that condition. The set conditions are shown as below.

- Number of first place for the last four months ≥ 10
- Percentage of first place for the last four months ≥ 0.2
- Percentage of first or second place for the last four months ≤ 0.4
- Age ≤ 25
- Age ≥ 40

Results and discussion of article generation

Some examples of generated articles are summarized in Table 6, 7 and 8. Existing articles cited from Odds Park [8]. The articles containing the above three pieces of information have been generated.

- Players expected to be ranked in the top three
- Information on cyclists expected to be ranked in the top three
- Information on the line

Generated articles can explain the line composition. Example 1 explains that the cyclists in the first and second place belong to the same line. Similarly, Example 2 explains that all the cyclists from the first to third places belong to the same line. A generated article can also describe the information on cyclists. In Example 1, it is described that the percentage of first place for the last four months of the cyclist predicted to be as the first place is 22.2%.

Compares the generated articles with the existing ones, the generated articles often contain the same cyclists as the existing articles. In Example 3, three people—Takimoto, Tobeuse, and Matsuo—listed in the generated article, are also listed as pre-eminent cyclists in existing articles. The above facts indicate that the content of the generated article is acceptable. However, it is a problem that generated articles have a poor vocabulary compared to existing articles. Moreover, it is a problem that the prediction of multiple race flows cannot be described.

Table. 6: example 1

2018-5-23 Muko-machi 6th race	
Existing article	"意地見せる地元高久保，器用な月森相手だが，松川が一枚上手か。" (Takakubo who is a local cyclist will show determination. Tsukimori is skillful. Matsukawa may be one step higher.)
Generated article	勝率 22.2%の③松川が自力で 1 着. 同ラインの⑤田中が③松川に続いて 2 着. ③松川ラインで連携してトップ 2 独占. ②月森ラインから④三宅が 3 着と予測. (③Matsukawa whose percentage of the first places is 22.2% wins on his own power. ⑤Tanaka belonging to the same line as ③Matsukawa takes the second place after him. The ③Matsukawa Line cooperates and monopolizes the top two. ④Miyake belonging to the ②Tsukimori takes the third place in the line.)

Table. 7: example 2

2018-5-23 Gihu 4th race	
Existing article	"二日目余裕で逃げ切った⑤山本のパワーが断トツ。ここも力任せの先行で豪快に逃げ切る。近畿勢⑦辻本が喰い下がると見るも、③安藤や①山中も気力次第か。" (⑤Yamamoto who easily won the race on the second day has the greatest power. He easily wins also in this race. ⑦Tsujimoto, who is a Kinki cyclist, takes the second place. ③Ando and ①Yamanaka may also win depending on their vitality.)
Generated article	勝率 33.3%の⑤山本が自分の脚で 1 着. 同ラインの⑦辻本が⑤山本に続いて 2 着. ②谷が 3 着. ⑤山本ラインが別線を抑えて上位独占と予測. (⑤Yamamoto whose percentage of the first place finishes is 33.3% wins on his own power. ⑦Tsujimoto belonging to the same line as ⑤Yamamoto takes the second place after him. ②Tani takes the third place. ⑤Yamamoto line monopolizes the top three.)

Table. 8: example 3

2018-5-23 Sasebo 7th race	
Existing article	"互角戦だが⑦滝本の自力に期待. ①戸伏の逆転も. ④内藤の連入十分. ②飯塚も自力で単独. 穴は⑥松尾の抜け出しから。" (Although the power of all cyclists are equal, I expect Tanaka. There is the possibility that Kobuse attains victory from behind. Naito may take the second place. Iizuka may win on his own power. There is a small possibility that Matsuo comes out on the top.)
Generated article	二連対率 45.8%の⑦滝本が自力で 1 着. 同ラインの番手①戸伏が⑦滝本に続いて 2 着. ⑦滝本ラインでトップ 2 独占. 別線の⑥松尾が 3 着と予測. (⑦Takimoto whose percentage of the first or second place finishes is 45.8% wins on his own power. Bante cyclist ①Tobuse belonging to the same line as ⑦Takimoto takes the second place after him. The ⑦Takimoto Line monopolizes the top two. ⑥Matsuo takes the third place.)

Conclusion

In this paper, we predicted the race result for Japanese Keirin races through a neural network and generated the article based on the predicted result. We compared the two-cyclist model of the pair-wise method with the nine-cyclist model of the list-wise method and demonstrated the advantages of each model. We generated articles explaining the prediction result using the template.

In future, we aim at designing a network structure that can predict the influence of the line without reducing the high prediction accuracy for the first place that the two-cyclist model shows. Another future task is the generation of articles that have abundant vocabulary and the generation of articles describing the prediction of multiple race flows.

Acknowledgments

This research was performed with the support of Charloto Co., Ltd. We express our gratitude here.

References

- [1] Yuuki Tagawa and Kazutaka Shimada. Generating abstractive summaries of sports games from Japanese tweets. *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 82–87, 2016.

- [2] Yuuki Tagawa and Kazutaka Shimada. Simple and sophisticated inning summary generation based on encoder-decoder model and transfer learning. *2017 International Conference on Asian Language Processing (IALP)*, pp. 252–255, 2017.
- [3] Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1374–1384. Association for Computational Linguistics, 2017.
- [4] Elnaz Davoodi and Ali Reza Khanteymoori. Horse racing prediction using artificial neural networks. In *Proceedings of the 11th WSEAS International Conference on Neural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems, NN'10/EC'10/FS'10*, pp. 155–160, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).
- [5] David Edelman. Adapting support vector machine methods for horserace odds prediction. *Annals of Operations Research*, Vol. 151, No. 1, p. 325, Nov 2006.
- [6] Noah Silverman and Marc Suchard. Predicting horse race winners through a regularized conditional logistic regression with frailty. *Journal of Prediction Markets*, Vol. 7, No. 1, pp. 43–52, 2013.
- [7] Manish Jogeah, Akshay Kumar Chandoo, Selukoto Pau-piah, and Sameerchand Pudaruth. Using fuzzy logic to predict winners in horseraces at the champ de mars. 01 2015.
- [8] Odds Park Corp. Odds park. <http://www.oddspark.com/keirin/>. (Accessed on 05/29/2018).
- [9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980, , 2014.

Vehicle Turn Signal Recognition using Internal features on Convolutional Neural Network

Keisuke Yoneda[†] Ryota Hagi[†] Naoki Suganuma[†]

[†] Kanazawa University, Kanazawa, Ishikawa, Japan

k.yoneda@staff.kanazawa-u.ac.jp

Abstract: Automated vehicle technology is rapidly developed as next-generation mobility. It recognizes surrounding circumstances using onboard sensors and predicts future behaviors for surrounding moving objects. This behavior prediction is conventionally based on moving and posture of surrounding vehicles. To help behavior prediction in advance, adding information of turn signal to the conventional approach is effective. This paper describes two networks created for turn signal recognition using Convolutional Neural Network (CNN) and Recurrent neural network (RNN). Both networks use different types of internal features from CNN. Some evaluations show that the proposed network obtains about 90% of recognition accuracy.

Keywords: Image Processing, Automated Driving, Convolutional Neural Network, Recurrent Neural Network

1. Introduction

Automated vehicle technology is rapidly growing. Many companies and research organizations have conducted public road experiments to develop self-driving cars. The automated vehicle recognizes surrounding circumstances, conducts its decision making and controls its steering and acceleration autonomously. It is generally equipped with various types of on-board sensors such as LiDAR, Millimeter-Wave Radar, Camera, GNSS/INS. Then, the following processes are operated in real-time.

- Self-Localization: Vehicle position is estimated by introducing GNSS/INS and applying map matching.
- Environmental Recognition: Static obstacles and dynamic objects such as vehicle, cyclist and pedestrian are recognized using range sensors. Dynamic road features such as traffic lights are recognized using vision sensors.
- Motion Prediction: Future object state is computed by using the digital map and dynamic object information.
- Decision-Making: The driving route is extracted from the predefined digital map by integrating recognition results.
- Trajectory Generation: Smooth and safety trajectory are optimized taking into account the driving lane, static and dynamic objects.
- Control: Steering and acceleration/deceleration are controlled along the obtained trajectory.

In the recognition procedure, the observations of the range sensors such as LiDAR and Radar make it possible to esti-

mate the position, velocity, orientation, size, and category of dynamic objects by time-series processing. The prediction step is then computed future object states by integrating velocity and yaw rate as a motion-based prediction. However, in this approach, there are situations that the automated vehicle cannot predict appropriate behaviors for surrounding vehicles. For example, at intersections, the automated vehicle cannot distinguish whether the stopped vehicle is parking or waiting to make a turn. Likewise, the intention of lane changes by surrounding vehicles cannot be recognized until they begin changing their lane. On the other hand, human drivers generally rely on turn signals to convey their intention to the surrounding drivers. By sharing this intention, the drivers can make their turn safely. From the point view of the read-ahead driving, a turn signal recognition method was developed using vision sensor in order to predict motions in advance for the multiple lanes and the intersection driving.

A turn signal detection method has been proposed using stereo camera [1]. It detects frontal vehicles and extracts turn signals for the detected vehicles using time-series image differencing. A flashing frequency is then calculated by applying Fast Fourier Transform (FFT). The adaboost algorithm then classifies the blinking state for the obtained power spectrum. Unfortunately, since there is no public dataset for the turn signal recognition, it is necessary to prepare it by yourself. Therefore, they improved accuracy by using the randomly generated frequency spectrum for learning data. Then, they achieved 86% recognition accuracy for expressway driving data.

In our previous works, the convolutional neural network (CNN) based recognition method was investigated [2]. It is

similar to the method of [1]. It recognizes the lighting state by applying CNN, and FFT computes the flashing frequency for the time series information to identify the blinking state. Although the dataset used for the evaluation was different, it reported a recognition rate of 82% for dataset on the Japanese public road. This method does not utilize the internal features of CNN, and uses only the classification results about the lighting state. Then, there is a problem that leads to false recognition. Therefore, this paper investigates the method of turn signal recognition by using CNN and Recurrent neural network (RNN) in order to improve the performance.

2. Proposed Method

2.1 Overview

The proposed method consists of the following procedures.

- (1) Extracting a bounding box of the surrounding vehicles.
- (2) Tracking the same vehicle from the time series images.
- (3) Classifying the lighting state for the single-frame image.
- (4) Classifying the blinking state for the multi-frames image.

The preceding vehicle is extracted from captured image from the onboard frontal camera. Because the vehicle image of the bounding box is almost a symmetric image, the classifier recognizes the turn signal for the half region of the image to reduce the complexity of the classification problem. The bounding box of the vehicle is then split into left/right images as shown in Fig. 1(a). The left half of image is turned over because of which, both left/right classifiers share the same CNN. In order to recognize turn signals, two categories (Turn ON or Turn OFF) are defined for both left and right images regardless of the lighting of the brake lamp. Figure 1(a) illustrates the CNN network model which was implemented in the previous work [2]. It was implemented using CNN with the AlexNet architecture [3]. For object classification, some network structures with higher recognition performance than AlexNet have been proposed [4]. They have many layers which extract object features to deal with the classification problem with large numbers of classes. However, in the turn signal recognition, it is necessary to pay attention to the lighting state of the lamp which is a small region with respect to the vehicle image. Therefore, the proposed method uses AlexNet which has a shallow network structure to focus on small regions.

In the previous approach [2], the CNN classifies the lighting state for both left and right images. FFT then computes the frequency of the blinking from the time-series data of classification results. Generally the turn signal blinks with a frequency of $1.5 \pm 0.5\text{Hz}$. Therefore, the number of frames for FFT can be computed based on the frequency of image capturing. Here, Δt denotes the image capturing interval.

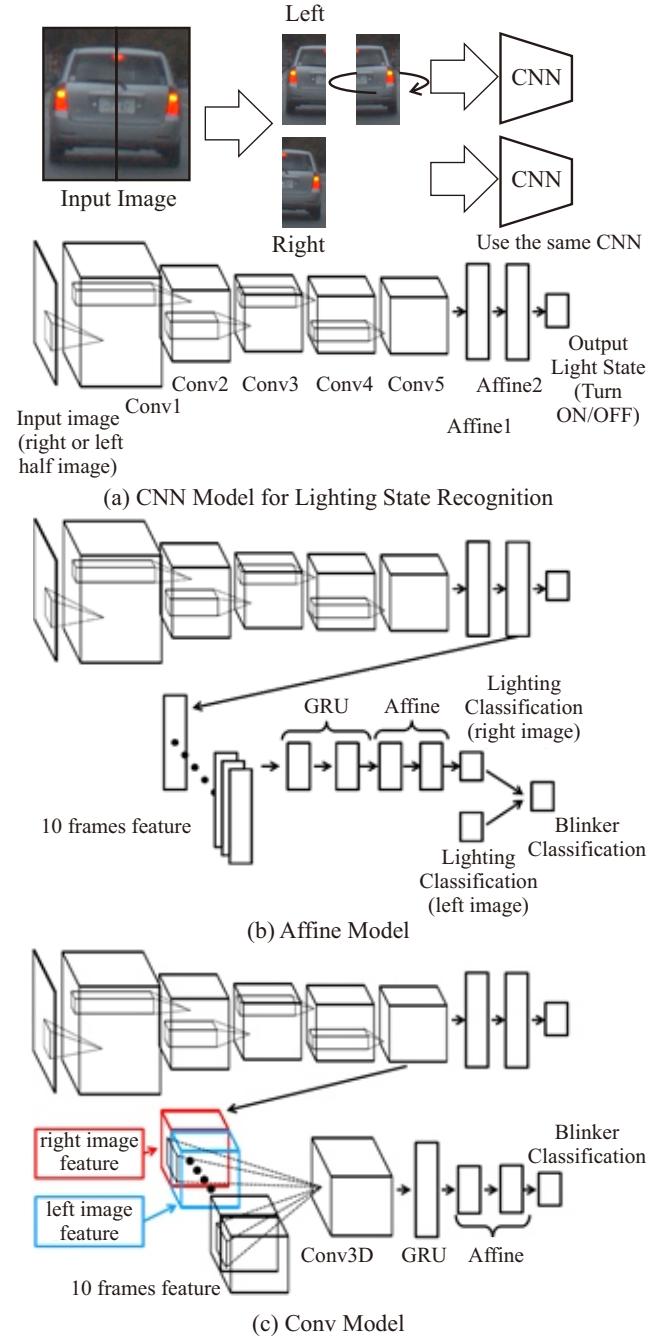


Fig. 1: Neural Network Model

The frequency resolution Δf of FFT is calculated using the following equations.

$$\Delta f = \frac{1}{N\Delta t} \quad (1)$$

$$\omega_n = \Delta f n \quad (0 < n < N) \quad (2)$$

where, ω_n is the obtained frequency and N is the number of input frames for FFT. In our research, the shooting frequency of the camera used for the experiment is set to 7.5 Hz. Therefore, the number of input frames is set to 10 in order to compute the blinking frequency within the range between

1.0Hz and 2.0Hz.

2.2 Vehicle Tracking

In order to extract surrounding vehicles, there are two strategies, “Sensor-Fusion based approach” and “Image-only based approach”. In the former method, surrounding dynamic objects are recognized using a range sensor such as LiDAR mounted on the automated vehicle [5][6]. In other words, the information about position, velocity, acceleration, orientation, size, and category for each object have already been estimated in real time using a time series processing such as the Kalman filter. Therefore, by transforming such object into image coordinates, bounding boxes can be obtained.

In the latter method, the state-of-the-art object detection method such as SSD (Single Shot multi-box Detector) [7] is applied to extract bounding boxes of the vehicle. The vehicle tracking is simply implemented by searching the maximum overlapped bounding box using IoU (Intersection over Unit). By using multiple methods in this manner, surrounding vehicles can be stably recognized. In this paper, recognition rate in vehicle tracking is not discussed.

2.3 Recognition Model

Figure 1(b) and (c) indicate the proposed network models. Both model is implemented using the CNN model (Fig. 1(a)) and the RNN with GRU (Gated Recurrent Unit) [8] architecture. Additionally, both model directly recognizes the blinking state using the internal feature from the CNN instead of applying FFT for the classification results in the previous work. Figure 1(b) is “Affine model” which inputs the feature from the affine layer of the CNN. Figure 1(c) is “Conv model” which inputs the feature from the convolutional layer of the CNN. Both model can extract intermediate features from the CNN network, but “Conv model” makes it possible to extract two dimensional features for the input image.

This paper evaluates recognition accuracies for these “Affine model” and “Conv model”.

2.4 Data Augmentation

Data augmentation is carried out to improve recognition performance. Generally, modified training images are generated by applying an affine transformation and brightness/contrast modifications to increase image diversity. For the purpose of turn signal recognition, this paper introduces two types of data augmentations concerning pseudo sunshine images and pseudo sequential images. In the situations with strong sunshine, there is a case that reflected light is wrongly recognized as a lighting region. Therefore, pseudo sunshine images are randomly generated by partially brightening it like a spotlight as shown in Fig. 2(a).

On the other hand, in the original continuous image, only blinking images or lights-out images are included. In order

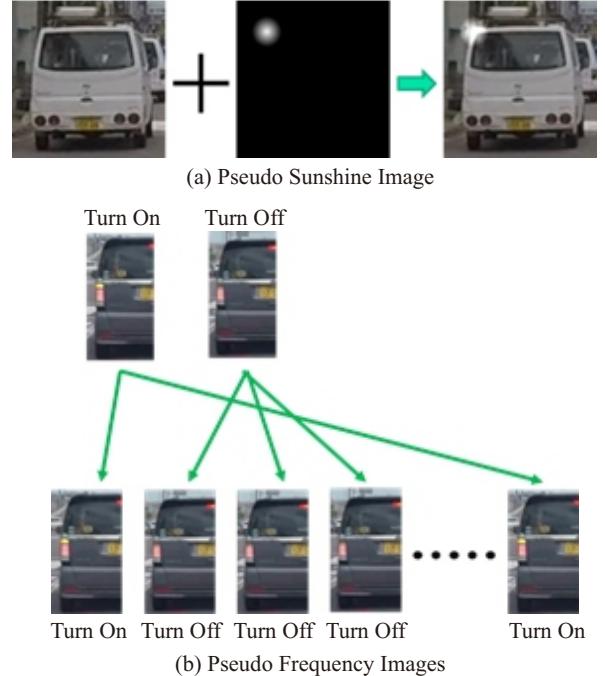


Fig. 2: Specialized Data Augmentations

to make it possible to recognize blinking with an appropriate frequency, a pseudo frequency image is created. As shown in Fig. 2(b), the pseudo sequential images are created by randomly arranging the turn-on and the turn-off images.

3. Evaluation

3.1 Data Collection

Training and testing datasets were then prepared by picking onboard camera images for daytime driving data. 1,405 training sequential images (No signal: 587, Right: 457, Left: 361) were prepared. A sequential image consists of 10 frames of images in the training data. As a testing data, 11 driving movies with 1,334 frames were prepared which includes 2,132 frames of vehicle images (No signal: 924, Right: 783, Left: 425).

3.2 Conditions

In order to verify the performance of the proposed method, two types of following evaluations are carried out for the testing dataset.

- Performance evaluation by adding specialized data reinforcement method
- Performance evaluation for different architecture of networks

In the evaluations, recognition accuracy and class average accuracy are compared for each condition.

Table. 1: Evaluated Results for Data Augmentations

	Basic	Basic + Pseudo Sunshine	Basic + Pseudo Freq.	All
No signal	0.879	0.940	0.885	0.904
Right	0.810	0.736	0.839	0.831
Left	0.741	0.769	0.830	0.825
Average	0.810	0.815	0.851	0.852

Table. 2: Evaluated Results for Recognition Models

	FFT Model	Affine Model	Conv Model
No signal	0.989	0.929	0.904
Right	0.659	0.890	0.831
Left	0.500	0.864	0.825
Average	0.716	0.894	0.852

3.3 Results

Table 1 shows recognition accuracies by adding specialized data augmentations. It summarizes the performance for Affine model in different data augmentations. It can be confirmed that the class average accuracy was improved by 4% by introducing the specialized data augmentations. It was shown that it was effective to increase the variation of the data by introducing the pseudo frequency images.

Table 2 shows recognition accuracies for different types of networks. In Table 2, FFT Model is a baseline described in [2]. It recognizes blinking states by directly applying FFT for lighting states. The results show that the proposed models have higher class average accuracy than the baseline. Although the accuracy of Affine model was the highest in this experiment, further analysis compared with Conv model is necessary.

4. Conclusion

The CNN and RNN based turn signal recognition method was developed. The effective network structure was clarified for the blinking state recognition through evaluations of time series processing with different input features. The results show that the obtained network can recognize turn signals with an accuracy of around 90%. It was shown that introducing the pseudo sequential images is an effective data augmentation. Further evaluations and performance improvements for night-time data are necessary as future works.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP17H03201.

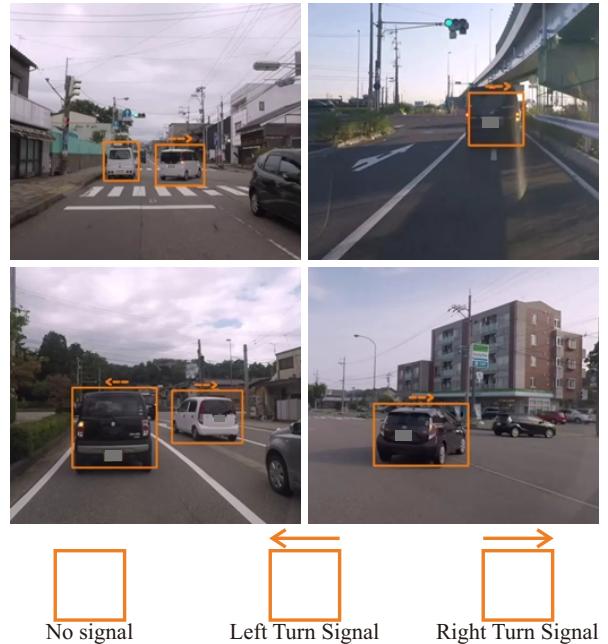


Fig. 3: Typical Recognition Results

References

- [1] B. Frohlich, M. Enzweiler and U. Franke, "Will this car change the lane?-Turn signal recognition in the frequency domain ", Proceedings of 2014 IEEE Intelligent Vehicles Symposium, 37-42, 2014.
- [2] K. Yoneda, R. Hagi, A. Kuramoto, M. Aldibaja, R. Yanase and N. Suganuma, "Convolutional Neural Network Based Vehicle Turn Signal Recognition ", Journal of Advanced Control, Automation and Robotics, Vol. 3, No. 2, pp. 102-106, 2017.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", In Advances in neural information processing systems(NIPS), pp. 1097-1105, 2012.
- [4] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [5] A. Teichman, J. Levinson, S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," IEEE International Conference on Robotics and Automation, pp. 4034-4041, 2011.
- [6] N. Suganuma, M. Yoshioka, K. Yoneda and M. Aldibaja, "LIDAR-based Object Classification for Autonomous Driving on Urban Roads", Journal of Advanced Control, Automation and Robotics, Vol. 3, No.=2, pp. 92-95, 2017.
- [7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single Shot Multi-Box Detector", Proceedings of European conference on computer vision, pp. 21-37, 2016.
- [8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", Proceedings of Empirical Methods on Natural Language Processing, pp. 1724-1734, 2014.

Combined Method of YOLO and IOU Tracker for Tracking and Identifying Chimpanzees

Yuichiro Ikeda [†] Hiroyuki Iizuka [†] Masahito Yamamoto [†]

[†] Hokkaido University, Hokkaido, Japan
ikeda@complex.ist.hokudai.ac.jp

Abstract: The development of information science in recent years has greatly contributed to studies of animal behavior. We are collaborating with Sapporo Maruyama Zoo to reduce the load on zookeepers taking care of animals using artificial intelligence. One of our targets is to automatically create chimpanzees' action histories for health management and maintenance of breeding environments. In order to create action histories, it is necessary to track individuals. However, tracking with identification is a challenging problem. In this research, we propose a system which combines the conventional image recognition, YOLO, and tracking, IOU Tracker, methods to realize tracking and identifying chimpanzees at the same time. Our experimental results show that it is possible for our system to improve detection results.

Keywords: Tracking, Identification, Detection, Chimpanzee

1. Introduction

The development of information science in recent years has greatly contributed to the field of animal behavior. Especially, it promotes the new research area called Computational Ethology [1], which automatically measures and analyzes animal behavior with information science technology. In Computational Ethology, it is important that individual identification and behavior classification of animals by analyzing images and videos. The development of this area is linked with the development of the information science, and it demands the application of image recognition technology based on deep learning such as convolutional neural network (CNN) [2] which has been making remarkable achievements in recent years.

Computational Ethology and image recognition technology are required for real-world applications. We are collaborating with Sapporo Maruyama Zoo to reduce the load on zookeepers taking care of chimpanzees using artificial intelligence. One of our targets is to automatically create chimpanzees' behavior history for health management and maintenance of breeding environments. Traditionally, caretakers have spent huge time and made a lot of efforts to check whether there was any abnormality in the animal by observing recorded fast-forward videos. Furthermore, analyzing the health condition of animals and relationships among animals depends on the experiences of skilled caretakers. Automatic creation of behavior history by artificial intelligence can change and improve the current situation of caretakers in Zoo. This automatically collects when, where, and what the animals did, it is assumed that development of an automatic

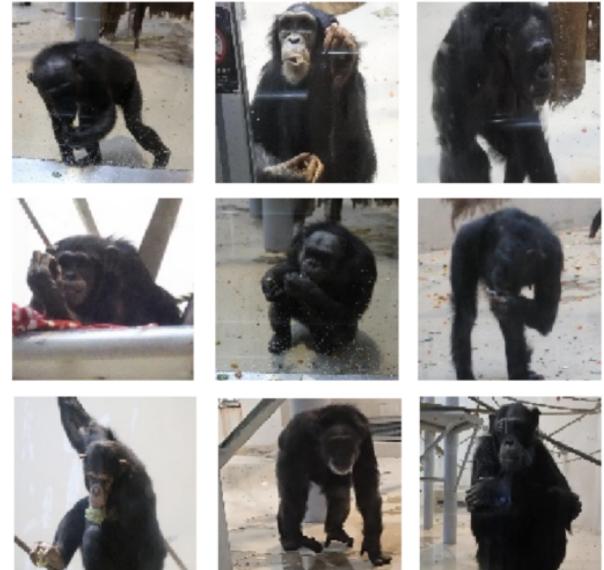


Fig. 1: Chimpanzees breeding at Sapporo Maruyama Zoo.

health management system of animals and provision of the most suitable environmental enrichment [3].

As a previous study of automatic behavior history creation, there was an attempt to automatically classify the behavior of wild birds by attaching an acceleration sensor on their body [4]. Although those sensor approaches can obtain direct information about animal behaviors, they disturb the animal behaviors and might hurt animals. On the other hand, the camera approaches can release animals and do not give any stresses to animals. For animals, it is better to use a cam-

era although it requires high image recognition technology. In fact, it becomes possible to realize such highly accurate image recognition thanks to the recent advance of artificial intelligence.

For the automatic creation of animal action histories, identification, tracking, and action classification of individuals are necessary. Our previous work shows that it is possible to identify chimpanzees from one-shot images that include only a single chimpanzee figure (Fig.1) [5]. However, animals are bred in an environment where multiple individuals can move in complicated ways. Therefore, we need a method to track and identify each individual from videos showing multiple individuals.

Since animals move freely in a breeding facility, individuals are occluded by the others or instruments of the facility in many cases. It might be able to solve the problem by putting multiple cameras but the locations for cameras are usually restricted in the case of zoos. Therefore, we propose a method for tracking and identifying individuals from videos captured by a single camera.

2. Data set

In this study, we apply our proposed method to nine chimpanzees bred at Sapporo Maruyama Zoo. We installed a camera outside the window at the inside breeding facility and shot a video in cooperation with Sapporo Maruyama Zoo. The indoor breeding facility is 100 square meters wide and 8 meters high. The captured video in the size of 1920×1080 was 5,002 frames at 5 fps, annotated the bounding box and individual name for each individual in each frame. At the time of learning, 4702 frames (5 fps, 15 min 40 sec) were used as training data and 300 frames (5 fps, 1 min) were used as test data. Additionally, training data images which were flipped in the horizontal direction was added as a training data set. From these data, we must correctly locate each individual's location and identify each individual every frame.

3. YOLO

Now, we verify how much individuals can be identified using YOLO [8] which can recognize multiple objects at the same time with high accuracy.

3.1 YOLOv3

Recently, in the field of Multi-Object Detection(MOD), there are a lot of models based on CNN such as Faster R-CNN [6], SSD [7], YOLO etc. In general, those detection speed and performance are in a trade-off relationship, but YOLOv3 shows high performances for both of them. YOLOv3 outputs the positions and sizes of the bounding boxes $[x, y, w, h]$ and the confidences of the predicted classes. Moreover, YOLOv3

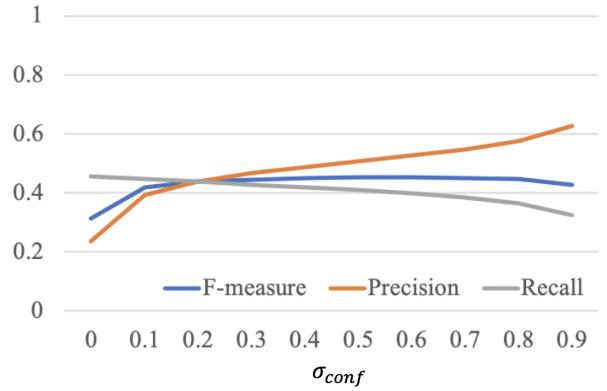


Fig. 2: Precision, recall and F-measure of the result by Individual Detector.

has the structure of Fully Convolutional Networks, and it can keep the positional relation of objects in the image up to the output. Moreover, since it has a structure combining Residual Blocks and Feature Pyramid Networks, detection performance to objects of various sizes is high, and chimpanzee photographed far away in a video can be detected.

3.2 Identifying results by YOLOv3

For training, we use CNN weights called Darknet 53 that are pre-trained on Imagenet. YOLO learns to correctly output the position and size of the bounding box surrounding each individual and its individual name. For each individual, only the predicted box with the maximum confidence is taken as the output result. This prevents the multiple detections of the same individual in each frame. In addition, boxes whose confidence exceeds the threshold σ_{conf} among those boxes were adopted as the detection result. We use precision, recall, and F-measure to evaluate the performance of detection.

$$Precision = \frac{\sum_t TP_t}{\sum_t TP_t + FP_t} \quad (1)$$

$$Recall = \frac{\sum_t TP_t}{\sum_t TP_t + FN_t} \quad (2)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

The learning result is shown in Fig. 2. As a result, it can be confirmed that the larger the σ_{conf} , the larger the precision and the smaller the recall. This is caused by increasing the detection number of boxes with low confidence as the σ_{conf} is increased. In addition, the F-measure was 0.453 at the maximum with $\sigma_{conf} = 0.6$. This result is insufficient to capture all the behavior of chimpanzees, and further improvement of the F-measure is required.

Figure 3 shows the precision, recall and F-measure at $\sigma_{conf} = 0.6$ at each frame of the test video. As a result, F-measure = 0, that is, there were several time periods in which

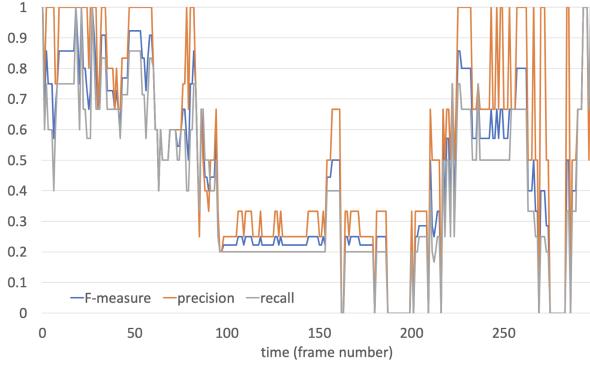


Fig. 3: Trends in precision, recall and F-measure at test data by Individual Detector.



Fig. 4: the picture of 282th frame in test data.

no individual could be detected. Figure 4 shows the example of an image where no chimpanzee was detected correctly. As one of the reasons, it is conceivable that learning of the detector is insufficient, but it is difficult to create a data set that covers the distribution of all attitudes in training because chimpanzee performs complicated actions. Also, since it is difficult to detect while identifying individuals, it may happen that no one is detected correctly.

On the other hand, there was also a time zone in which the $F\text{-measure} = 1.0$, that is, a time zone in which all individuals could be correctly detected. Therefore, even if there is a frame that is difficult to detect an individual, it is considered that it can be specified by tracking that individual and referring to the information of the detectable frame.

4. Proposed method

Detection and identifying chimpanzees in all frames is a difficult task. Detection of YOLO is independent in each frame and has no relevance in the time direction, so it is impossible to correct a wrong target with information of another frame with high confidence. Therefore, by associating each frame in the time direction by a tracking method, it becomes possible to use information of other frames.

Hence, our proposed system combines a method of MOD

into Tracking as shown in Fig. 5. First, The Individual Detector performs detection with identification, and the Chimpanzee Detector performs detection of chimpanzees without identification. Next, the IOU Tracker generates trajectories based on the input from the Chimpanzee Detector. Finally, their trajectories are identified based on the detection results of the Individual Detector.

4.1 Previous study of identifying trajectory

Lu et al. [10] proposed a method of tracking while playing basketball game videos with each player's identification. In this study, they used features such as MSER, SIFT and RGB color histogram of tracking players. However, unlike players wearing uniforms, it is difficult to identify chimpanzees with similar features and colors in their method. Therefore, it is considered effective to learn these feature quantities automatically by YOLO based on CNN.

In addition, Lu et al. [11] proposed a method to classify players' teams with a CNN by cutting out players' detection result in football game videos. However, since the chimpanzees are captured in various postures, the aspect ratio of the bounding boxes does not become constant. Also, unlike broadcast videos of group sports, chimpanzees do not show up in a certain size, so the quality of the input image to the Classification Network is significantly different. Therefore, we need a high detection performance to objects of various sizes. Then, YOLOv3 is robust against the detection of multiple aspect ratio chimpanzees.

In this way, several methods have been proposed for identifying tracked trajectories, but YOLOv3 is considered to be good for identifying chimpanzees that perform complex behaviors.

4.2 Combining IOU Tracker with Individual Detector

As a tracking method, we use IOU Tracker [9] which is a simple Tracking-by-Detection method but show high precision in the MOT tasks. IOU Tracker is an algorithm that allocates the same ID to the tracking target whose IOU between the current and previous detection boxes is larger than the threshold. For two bounding boxes a and b , the IOU is calculated by the following Eq.4.

$$IOU(a, b) = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \quad (4)$$

For input of IOU Tracker, Chimpanzee Detector by YOLOv3 which detects chimpanzees without individual identification is used. If individual identification is not involved, detection of chimpanzees is easier. Therefore, it is considered that the trajectory is also generated in the frame in which no chimpanzee is detected by the Individual Detector. By IOU Tracker, target identification is not performed, thus it is necessary to identify by another method.

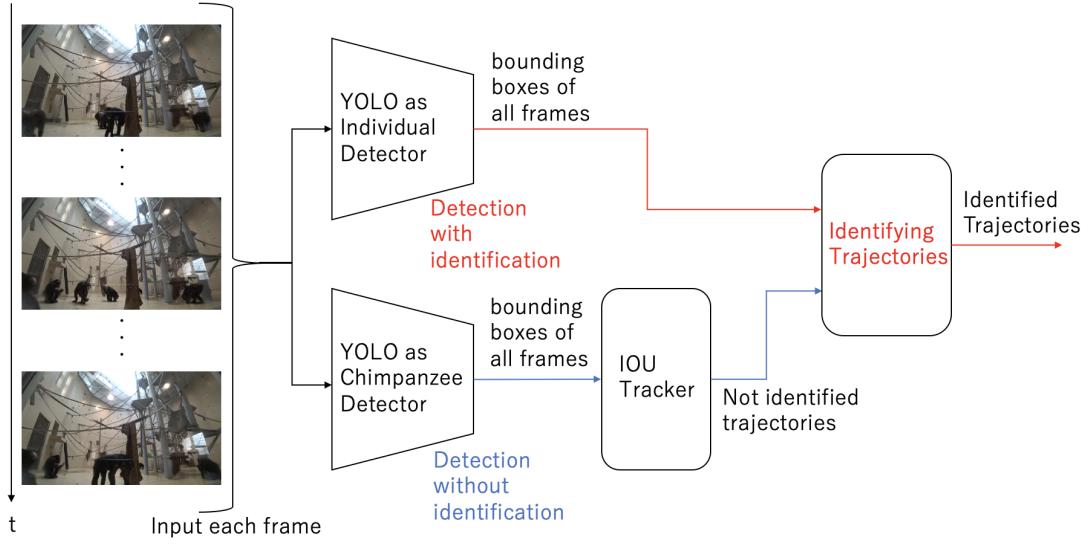


Fig. 5: Outline of our proposed model. The sequence of images is input to two detectors. From the sequence of bounding boxes obtained by Chimpanzee Detector, we obtain some trajectories by IOU Tracker. The trajectories are identified by the sequence of bounding boxes obtained by Individual Detector.

Algorithm 1 Identify Trajectories

```

Input:  $\sigma_{conf}$                                 ▷ threshold of confidence
Input:  $\sigma_{iou}$                                 ▷ threshold of IOU
Input:  $M$                                     ▷ the number of chimpanzees
Input:  $N$                                     ▷ the number of frames
Input:  $\hat{B}$                                 ▷ boxes found by Individual Detector
Input:  $\hat{T}$                                 ▷ Unidentified trajectories
Output:  $\tilde{T}$                                 ▷ Identified trajectories

Calculate sums of confidence :
Initialize  $S[\text{len}(\hat{T})][M] = 0$  ▷ S is sums of confidences
for  $t = 0$  to  $N$  do
    for all  $box \in \hat{B}[t]$  do
        for all  $trajectory \in \hat{T}$  do
            if confidence of  $box > \sigma_{conf}$  then
                if  $IOU(box, trajectory[t]) > \sigma_{iou}$  then
                     $i \leftarrow$  the id of trajectory
                     $j \leftarrow$  individual id of box
                     $S[i][j] \leftarrow S[i][j] + \text{confidence of } box$ 

Allocate individual id to  $\tilde{T}$  :
 $L = []$                                          ▷ assigned trajectory id list
while  $\max(S) \neq 0$  do
     $(i, j) \leftarrow \text{argmax}(S)$  ▷ i=trajectory id, j=individual id
     $S[i][j] \leftarrow 0$ 
    if  $L.\text{find}(i)$  then
        continue
    if not  $\tilde{T}.\text{find}(\text{time and } j \text{ competing one})$  then
        allocate  $j$  as individual id to  $\tilde{T}[i]$ 
        Add  $i$  to  $L$ 
return  $\tilde{T}$ 

```

Hence, our proposed method identifies the trajectories which IOU Tracker generates by Individual Detector shown in Algorithm 1. First, we obtain the IOU above the threshold σ_{iou} between the trajectory's bounding box of each frame and the identified box of the Individual Detector, and calculate the sum of the confidence over the threshold σ_{conf} of each individual. Next, for the largest sum of the confidence of individuals in each trajectory, the individual name is taken as the individual name of the trajectory. Then, one by one check in order of the sum total of confidence, and assigns the individual name to the trajectory, but skips when the same individual name as that trajectory and time zone overlap already existed. The reason is that it is unnatural that a plurality of specific individuals exists in the same time zone, and The trajectories with the higher values of the confidence are more reliable for identification. Finally, when individual names are allocated to all trajectories or all of the confidence sums are checked, identification is terminated.

5. Results

5.1 Tracking without identification result

5.1.1 Chimpanzee Detector result

First, we verify whether chimpanzees can be detected without identification with YOLOv3. In order to perform tracking with the IOU Tracker, the target position must be correctly detected in almost all frames. As a result, it was possible to detect "chimpanzee" with very high performance because the F-measure was 0.899 at the maximum with $\sigma_{conf} = 0.9$.

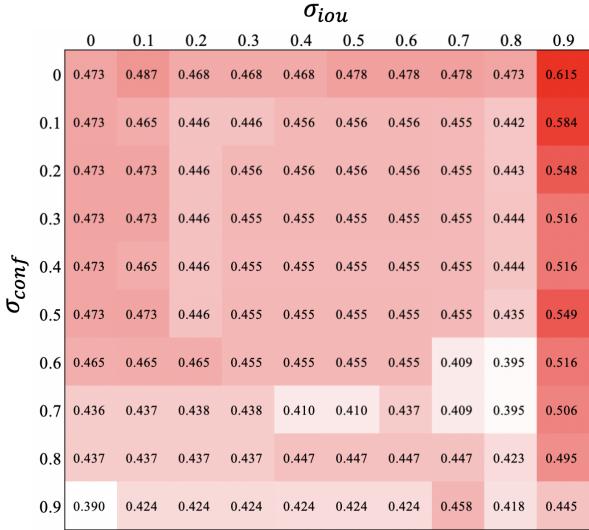


Fig. 6: Heatmaps of F-measure when σ_{conf} and σ_{iou} are changed.

5.1.2 IOU Tracker result

We allocate IDs to the outputs of the Chimpanzee Detector by IOU Tracker. IOU Tracker's parameters are the default of the sample codes provided by the authors¹. $IOU \geq 0.5$ as the criterion that the current and previous target are same. To evaluate the performance of the tracking result, we measure two ratios of between trajectories and annotated boxes.

$$R_{lost} = \frac{\sum_t lost_t}{\sum_t g_t} \quad (5)$$

$$R_{mismatch} = \frac{\sum_t mismatch_t}{\sum_t g_t} \quad (6)$$

where, $lost_t$ and $mismatch_t$ are the number of times losing and mismatching the tracking targets at time t , respectively. When IOU becomes less than 0.5, the tracking is cut off, which means the different IDs are assigned between previous and current steps. g_t is the number of annotated boxes. As a result, we obtained a R_{lost} of 0.136 and a $R_{mismatch}$ of 0.146 for the test data set. It means that targets were sometimes lost or replaced, but tracking were roughly correct.

5.2 Correction experiment result

Figure 6 shows the F-measure by a heat map as a result of correcting the trajectories. The lower the σ_{conf} was, the higher the F-measure tended to be. Since this is close to recall's tendency in Individual Detection showed in Fig. 2, it is considered that our proposed method effectively works by increasing the number of boxes that will help correction even if the confidence to the correct individual is low. Also, when $\sigma_{iou}=0.9$, the F-measure was clearly high. In the situations where several individuals are densely populated, there

¹<https://github.com/bochinski/iou-tracker>

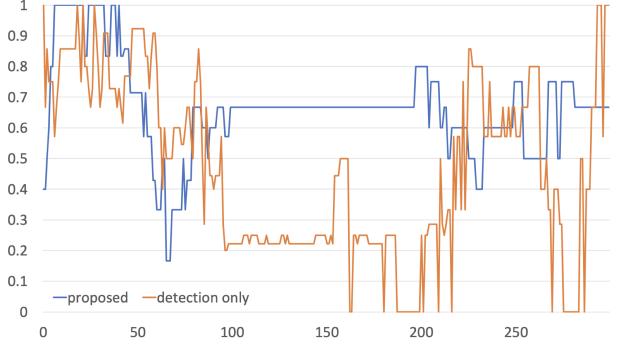


Fig. 7: Comparing trends in precision, recall and F-measure at test data by our proposed method and detection only.

is a possibility that the predicted box of an individual is erroneously used to correct the trajectory of other individuals. By setting the σ_{iou} higher, it is considered to be able to prevent it.

Figure 7 shows the results of F-measure under $\sigma_{conf}=0.0$ and $\sigma_{iou}=0.9$ at each frame of the test video by our proposed method. The result is compared with the result of the only detection method shown in Fig. 3. Focusing on the 100th to 200th frames, we can see that our proposed method could distinguish individuals that could not be identified by detection only. In each 100th to 200th frame, most individuals were not detected as shown in Fig. 8 (c), and the confidences were very small. Therefore, the sums of the confidences of each trajectory including the 100th to 200th frames tended to depend on the detection results of other frames with high confidence. Hence, since the F-measures and confidences of results by the Individual Detector were high except for the 100th to 200th frames, it can be considered that the proposed method was able to correct the result at the 100th to 200th frames. Each result of the 197th frame of the test data is shown in Fig. 8. Detection alone cannot detect all chimpanzees, but it is almost correctly detected by our proposed method. This is because it was corrected by correctly detected results in other frames.

Figure 9 shows the changes in the x coordinates of the trajectories according to time t . Some trajectories were disconnected or identified as other individuals. For some disconnected trajectories, like Susie's one, it can be identified as the same individual again. This means that our proposed method can re-identify the trajectories in which different IDs were assigned by the IOU Tracker, and it can be confirmed that our proposed method worked effectively.

On the other hand, our proposed method largely depends on the results of the IOU tracker. First of all, it is difficult to obtain a detector that can be identified individually in every situation. Therefore, if the individual can be properly tracked, even if there are unidentifiable frames, it can be considered that correction can be made from the information on other

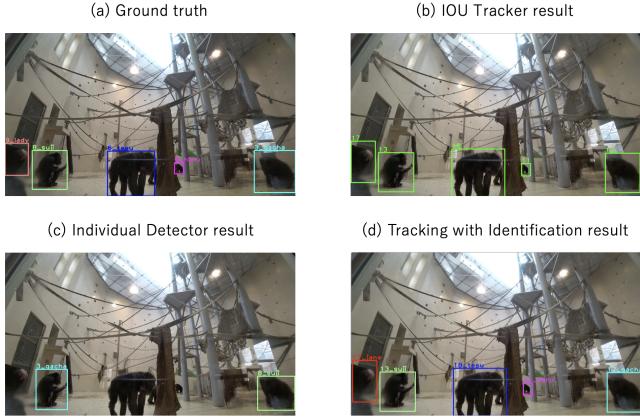


Fig. 8: Comparison of actual (a)ground truth, (b)tracking result, (c)detection result and (d)result of our proposed method in the 197th frame of test data.

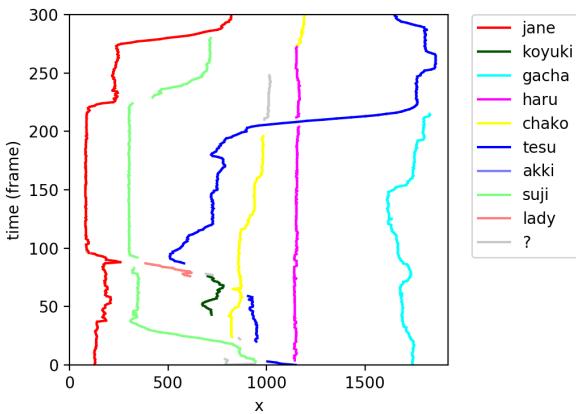


Fig. 9: Trajectories for test data by our proposed method.

frames. This is the basic idea of our proposed method. Nevertheless, if the tracking is incorrect, it is possible that all the results of the trajectory will be erroneously identified. In our proposed method, a simple method called IOU Tracker was used, but this method uses only past frames for determining the ID of the trajectory of the current frame. However, since it is possible to use future frames in this research, even if tracking is used, the accuracy may be improved if trajectories are acquired using all frames.

6. Conclusion

In this paper, we proposed a novel method to identify and track chimpanzees from only videos taken with a single fixed camera. Our proposed method identifies the trajectories by another detector for individual identification of chimpanzees. We also investigated valid parameters in our proposed method. By comparing the results discriminated only by the detector and the results of our proposed method, significant improvement in F-measure was shown. This result

shows that our proposed method is effective.

ACKNOWLEDGMENTS

This work is supported by Sapporo City Economic and Tourism Bureau, Sapporo City Environment Bureau Sapporo Maruyama Zoo and Sapporo Electronics and Industries Cultivation Foundation. The authors would like to thank all those involved.

References

- [1] Anderson D. J. and Perona P. "Toward a science of computational ethology", *Neuron*, Vol. 84, pp.18-31, 2014.
- [2] Y. LeCun, L. Bottou, Y. Ben-gio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 86(11), pp. 2278-2324, 1998.
- [3] Maple Terry and Perdue Bonnie, "Environmental Enrichment", *Zoo Animal Welfare*, 95-117, 2013.
- [4] Sakamoto KQ, Sato K, Ishizuka M, Watanuki Y, Takahashi A, et al, "Can Ethograms Be Automatically Generated Using Body acceleration data from Free-Ranging Birds?", *PLOS ONE*, 4(4), e5379, 2009.
- [5] Yuichiro Ikeda, Hiroyuki Iizuka, and Masahito Yamamoto, "Identifying Individual Chimpanzee with Convolution Neural Networks," *JSAI2018* proceedings, pp. 93–95, 2018.
- [6] Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, Jian, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *Advances in Neural Information Processing Systems*, 28, pp. 91–99, 2015.
- [7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C. Berg, "SSD: Single Shot MultiBox Detector", *ECCV* 2016, pp21–37, 2016.
- [8] Redmon, Joseph and Farhadi, Ali "YOLOv3: An Incremental Improvement", *arXiv*, 2018.
- [9] E. Bochinski, V. Eiselein and T. Sikora, "High-Speed tracking-by-detection without using image information", 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, pp. 1-6, 2017.
- [10] W. Lu, J. Ting, J. J. Little, and K. P. Murphy, "Learning to Track and Identify Players from Broadcast Sports Videos", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1704-1716, July 2013.
- [11] Keyu Lu, Jianhui Chen, James J. Little, Hangen He, "Lightweight convolutional neural networks for player detection and classification", *Computer Vision and Image Understanding*, 2018.

A Novel Bare-bones Particle Swarm Optimization Algorithm with Co-evaluation

Jia Guo [†] Yuji Sato [‡]

[†] Hosei University, Tokyo, Japan

[‡] Hosei University, Tokyo, Japan

guojia@ieee.org

Abstract: A novel bare bones particle swarm optimization algorithm with co-evaluation (BBPSO-C) is proposed in this work. A shadow particle swarm is used to enhance the global search ability of the proposed algorithm. A dynamic grouping method is used to disperse both the shadow particle swarm and the original particle swarm. After the dispersion, an exchanging process will be held between the two swarms. The original swarm will be more concentrated and the shadow swarm will be more scattered. In addition, particles in different swarms are not static. The grouping and exchanging process will be implemented every iteration. With the moving of particles between the two particle swarms, the BBPSO-C gains the ability on both the global search and the local search. To verify the search ability of the proposed method, a set of comprehensive benchmark functions are used in the experiments. Finally, the experimental results confirmed the optimization ability of the BBPSO-C.

Keywords: Particle swarm, bare bones, co-evaluation

1. Introduction

The particle swarm optimization (PSO) algorithm is a global optimization method first proposed by Kennedy and Eberhart[1] in 1995. It is a population-based algorithm that inspired by the team behaviors like birds flocking and fish schooling. Particles are designed to have memories to record the best position of themselves and the swarm is designed to have a memory to record the best position among all particles. In each evaluation, particles change their positions according to their personal best positions and the global best position of the swarm. In this pattern, particles will move toward to the global best particle. When the evaluation time reaches the max or all particles gather to a certain extent, the algorithm can output the final result. These features make the PSO a powerful method in different areas. However, the evaluation pattern has a shortcoming at premature convergence. Plenty of researcher try to solve this problem and improve the performance of the PSO [2].

The bare bones particle swarm optimization algorithm (BBPSO) is a simple version of the PSO, which is proposed by Kennedy [3] in 2003. Particles in the BBPSO are no longer moving in the search space. With the canceling of the velocity term, the BBPSO becomes parameter free and easy applying. A Gaussian distribution is used to select the next position of each single particle. However, the BBPSO still has some defect like the premature convergence. To cross the shortcomings of the original BBPSO, Guo [4] proposed a pair-wise bare bones particle swarm optimization algorithm

(PBBPSO). A separate iteration strategy is used in the pair-wise operator to enhance the diversity of the swarm. Particles are classified into two groups, the leader group and the follower group. Different iteration strategies are used to different groups to slow down the speed of diversity losing.

2. Related Works

Yang and Sato [5] proposed a modified bare bones particle swarm optimization with differential evolution (BBPSO-DE) for large-scale problem. A ring neighborhood topology is used in the proposed method. Also, the mutation and crossover operators of DE strategy is adopted in the BBPSO-DE. To verify the performance of the proposed method, a set of large-scale benchmark functions are used in the experiments. The experimental confirmed the optimization ability of the BBPSO-DE.

Guo and Sato [6] proposed a dynamic allocation bare bones particle swarm (DABBPSO) optimization algorithm. The DABBPSO is inspired from the social structure of humane society. Particles with different positions are considered to have different talents. Using this logic a dynamic allocation method is proposed in their work. Particles are allocated to different groups according to their characteristics. Different evaluation rules are implemented in different groups.

Shen [7] proposed a hierarchical learning bare-bones particle swarm optimization (HLBBPSO) algorithm. The HLBBPSO adopts a hierarchical learning strategy to maintain the population diversity. An archive is used to store the

accepted infeasible solutions and auxiliary operations are introduced to help accepted infeasible solutions to enter into the feasible region. Guo [8] proposed a bare bones particle swarm optimization algorithm with dynamic local search (DLS-BBPSO). Particles are groups without human intervention.

However, the population-based algorithms still face the premature convergence and maybe trapped in some local minimal when dealing with multimodal functions. To solve this dilemma, the bare bones particle swarm optimization algorithm with co-evaluation is proposed in this work. The details of the proposed method is shown in the next section.

3. The bare bones particle swarm optimization algorithm with co-evaluation

The bare bones particle swarm optimization algorithm with co-evaluation (BBPSO-C) is proposed in this section. A shadow swarm is used in the BBPSO-C to enhance the search ability. The shadow swarm will assist the original swarm during iterations. Particles are flowing between the two swarms. Generalized speaking, particles which are more dispersed will move to the shadow swarm and the particles which are more central will move to the original swarm. The BBPSO-C is consisted of two major steps: the dynamic classification and the particle exchanging. They will be introduced in the following subsection.

3.1 Dynamic classification

A dynamic classification method is used to classify both the original particle swarm and the shadow particle swarm. The classification is implemented inside a particle swarm to keep it individual. Each time two random particles are selected from the swarm. A comparison will hold between the two particles. The particle with better personal best position will be assigned to the main group (MG) while the other one will be assigned to the assistant group (AG). The next position of the particle in the AG will be selection from the following equation:

$$\begin{aligned}\alpha &= \frac{(pbest(MG) + pbest(AG))}{2} \\ \beta &= |pbest(MG) - pbest(AG)| \\ x(AG) &= N(\alpha, \beta)\end{aligned}\quad (1)$$

where the $Pbest = (pbest(1), pbest(2), pbest(n))$ is the personal best position of each particle; the $N(\alpha, \beta)$ is a Gaussian distribution; the $x(i)$ is the next position of the particle. On the other hand, the next position of a particle in the MG is selected from the following equation:

$$\begin{aligned}\gamma &= \frac{(pbest(MG) + gbest)}{2} \\ \delta &= |pbest(MG) - gbest| \\ x(MG) &= N(\gamma, \delta)\end{aligned}\quad (2)$$

where the $Pbest = (pbest(1), pbest(2), pbest(n))$ is the personal best position of each particle; the $N(\gamma, \delta)$ is a Gaussian distribution; the $x(i)$ is the next position of the particle; the $gbest$ is the global best position of the swarm. The pseudo-code of this process in the original swarm is shown in **Algorithm**. The same method will be used in the shadow swarm.

Algorithm BBPSO-C: grouping, original swarm

```

Require: Max iteration time,  $T$ 
Require: Particle swarm  $X = x_1, x_2, \dots, x_n$ 
Require: Personal best position  $Pbest = p_1, p_2, \dots, p_n$ 
Require: Global best position  $Gbest$ 

1: while  $t < T$  do
2:   while  $X \neq \emptyset$  do
3:     Randomly select two particles,  $x_i$  and  $x_j$ , from  $X$ 
4:     if  $f(x_i) < f(x_j)$  then
5:       Update  $x_i$  with equation (2)
6:       Put  $x_i$  into the MG
7:       Update  $x_j$  with equation (1)
8:       Put  $x_j$  into the AG
9:     else
10:      Update  $x_i$  with equation (1)
11:      Put  $x_i$  into the AG
12:      Update  $x_j$  with equation (2)
13:      Put  $x_j$  into the MG
14:    end if
15:   end while
16:   Update  $Pbest, Gbest$ 
17:    $t=t+1$ 
18: end while
```

Since the selection of the two particles is random, it is uncertain that whether the two particles are very near or very far. If the two particles are close enough, the iterative pattern between the two particles should be a local search. Otherwise, if the two particles are wide apart, the iterative pattern should be a global search. The possible iterative pattern of the dynamic classification is shown in Fig. 1.

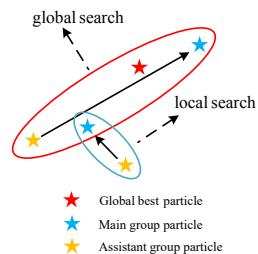


Fig. 1: Swarm construction of the elite selection

3.2 Particle exchanging

After the dynamic classification, each swarm has one MG and one AG. The AG of the original swarm and the MG of the shadow swarm will change their positions in the next iteration. The exchanging pattern is shown in Fig. 2.

After the exchanging, the original swarm gains stronger ability on locating the local minimum with the two MGs. Meanwhile, the shadow swarm gains stronger ability on escaping from a local minimum with the two AGs.

Also, the dynamic classification will be implemented in the next iteration. The construction of current AG and MG will be destroyed and reconstructed every iteration. The fluxion of the particles between the two swarms increases both the local search and the global search ability of the proposed algorithm.

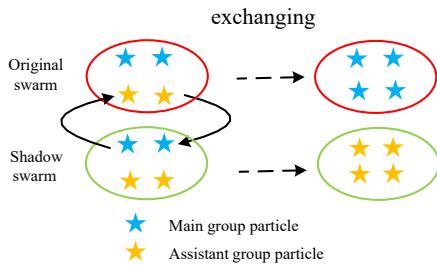


Fig. 2: Swarm construction of the elite selection

4. Experiments and discussion

4.1 Experimental methods

To verify the performance of the BBPSO-C, both unimodal and multimodal benchmark functions are used in the experiment. The details of the benchmark functions are shown in Table 1. All the test functions are in two major groups, the unimodal group ($f_1 - f_4$) and the multimodal group ($f_5 - f_8$). In the unimodal group, the f_1 is a simple unimodal function. The f_2 to f_4 are collected by Suganthan [9]. The f_2 is a shifted unimodal function; the f_3 is a shifted and rotated unimodal function; the f_4 is a scalable and noise in fitness unimodal function. In the multimodal group, the f_5 is a wildly used test function, the f_6 to f_8 are collected by Suganthan [9]. The f_6 is a shifted separable function; the f_7 is a shifted and rotated scalable function; the f_8 is a shifted, rotated function which continuous but differentiable on a set of points.

Also, the BBPSO [1], the PBBPSO [4] and the DLS-BBPSO [8] are used as the control group. The dimension of all functions is 30. The population of each algorithm is 30. The iteration time of each test is 1500. Also, to reduce the accidental error, all results are the average value from 30 independent runs.

4.2 Experimental results and discussion

The experimental results are shown in Table 2. The empirical error is defined as $|result - TOS|$, where the *result* is the global best value given by an algorithm after its final iteration, the *TOS* is the theoretical optimal solution (TOS) of the function. To make a clear competition, we rank the result form the best to the worst. The best result gets one point and the worst one get 4. The average rank result is shown at the bottom of Table 2. In the unimodal group, the BBPSO-C gets two first and two second rank. On f_1 , the result of the BBPSO is more than 1.00E+3 times larger than the BBPSO. On f_2 , the BBPSO-C gets the first rank. The result is 53.73% smaller than the result of the PBBPSO. On f_3 , the result of the BBPSO-C is 83.62% smaller than the result of the DLS-BBPSO. On f_4 , the result of the BBPSO-C is 18.92% times larger than the first rank, the PBBPSO. In the multimodal group, the BBPSO-C gets two first and two second rank. On f_5 , the result of the BBPSO-C is 2.91E-10 while other results are around 2.00E-02. It means the BBPSO-S performances much better than other algorithms. On f_6 , the BBPSO-C is the second rank. The result is 17.48% larger than the result of the DLS-BBPSO. On f_7 , the BBPSO-C gets the first. The result is 17.65% smaller than the result of the PBBPSO. On f_8 , the result of the BBPSO-C is 4.93% larger than the result of the BBPSO. To sum up, the performance of the BBPSO-C is best among all the algorithms. The average rank is 1.5. It means that the proposed co-evolution method can improve the search ability of the original BBPSO on both unimodal and multimodal functions.

5. Conclusion

A bare-bone particle swarm optimization algorithm with co-evaluation (BBPSO-C) is proposed in this work. A shadow particle swarm is used in the BBPSO-C to keep the diversity of the original swarm. Particles in each swarm are divided into two groups: the main group and the assistant group. Particles in the main group are more concentrated and particles in the assistant group are more scattered. After that, the assistant group of the original swarm and the main group of the shadow swarm will change their positions. The original swarm with two main groups will have a stronger ability in locating the local optimum. The shadow swarm will have a stronger ability on escaping from a local optimal. In the next iteration, the particle grouping and exchanging process will be implemented again. The two swarms work together to find the global optimal position in the search area. To verify the performance of the proposed method, a set of comprehensive benchmark functions are used in the experiments. Finally, the experimental results confirmed the optimizing ability of the BBPSO-C.

Table. 1: Experiment Functions

Function	Search Range	Minimum	Dimension
f_1 = Sphere Function	(-100, 100)	0	30
f_2 = Shifted Schwefel's Problem 1.2	(-100, 100)	-450	30
f_3 = Shifted Rotated High Conditioned Elliptic Function	(-100, 100)	-450	30
f_4 = Shifted Schwefel's Problem 1.2 with Noise in Fitness	(-100, 100)	-450	30
f_5 = Ackley Function	(-32.768, 32.768)	0	30
f_6 = Shifted Rastrigin's Function	(-5, 5)	-330	30
f_7 = Shifted Rotated Rastrigin's Function	(-5, 5)	-330	30
f_8 = Shifted Rotated Weierstrass Function	(-0.5, 0.5)	90	30

Table. 2: Comparisons of empirical error between BBPSO, PBBPSO, DLS-BBPSO and BBPSO-C

Function	BBPSO[3]		PBBPSO[4]		DLS-BBPSO[8]		BBPSO-C	
	Mean	Std. Dev.						
f_1	2.04E-21	3.70E-21	1.67E-15	3.67E-15	8.64E-15	1.98E-14	3.14E-18	5.26E-18
f_2	1.97E+04	1.29E+04	1.10E+03	7.08E+02	5.45E+03	4.69E+03	5.09E+02	2.25E+02
f_3	2.94E+08	1.13E+08	1.07E+07	8.91E+06	4.47E+07	3.66E+07	7.32E+06	3.48E+06
f_4	2.13E+06	9.25E+05	1.11E+04	4.12E+03	1.54E+04	5.96E+03	1.32E+04	7.470E+03
f_5	3.27E-02	7.89E-02	1.54E-02	4.74E-02	8.86E-02	3.40E-02	2.91E-10	3.47E-10
f_6	7.12E+01	2.21E+01	5.29E+01	1.61E+01	3.89E+01	7.56E+00	4.57E+01	1.21E+01
f_7	2.07E+02	1.90E+02	1.70E+02	6.21E+01	2.15E+02	5.40E+01	1.40E+02	6.69E+01
f_8	3.85E+01	4.89E+00	4.06E+01	2.06E+00	4.11E+01	1.32E+00	4.04E+01	1.20E+00
Average Rank	3		2.25		3.25		1.5	

References

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [2] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Computing*, vol. 19, no. 2, pp. 431–448, 2015. [Online]. Available: <http://link.springer.com/10.1007/s00500-014-1262-4>
- [3] J. Kennedy, "Bare bones particle swarms," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 2003, pp. 80–87.
- [4] J. Guo and Y. Sato, "A pair-wise bare bones particle swarm optimization algorithm," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, no. 1. IEEE, may 2017, pp. 353–358. [Online]. Available: <http://ieeexplore.ieee.org/document/7960018/>
- [5] S. Yang, "Modified Bare Bones Particle Swarm Optimization with Differential Evolution for Large Scale Problem," in *2016 IEEE Congress on Evolutionary Computation*, 2016, pp. 2760–2767.
- [6] J. Guo and Y. Sato, "A dynamic allocation bare bones particle swarm optimization algorithm and its application," *Artificial Life and Robotics*, vol. 23, no. 3, pp. 353–358, 2018. [Online]. Available: <http://dx.doi.org/10.1007/s10015-018-0440-3>
- [7] Y. Shen, J. Chen, C. Zeng, and B. Ji, "A novel constrained bare-bones particle swarm optimization," in *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, 2016, pp. 2511–2517.
- [8] J. Guo and Y. Sato, "A Bare Bones Particle Swarm Optimization Algorithm with Dynamic Local Search," in *Studies in Computational Intelligence*, 2017, vol. 248, pp. 158–165.
- [9] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *Report*, no. May, pp. 1–50, 2005.

Evolutionary Learning of Multiple TTSP Graph Structured Patterns from Positive and Negative TTSP Graph Data: Its Graph Pattern Display System and Performance Evaluation

Yuuki Yamagata[†], Fumiya Tokuhara[†], Yuito Inaba[†], Kouki Muratani[†]

Tetsuhiro Miyahara[†], Yusuke Suzuki[†], Tomoyuki Uchida[†] and Tetsuji Kuboyama[‡]

[†]Faculty of Information Sciences, Graduate School of Information Sciences

Hiroshima City University, Hiroshima 731-3194, Japan

[‡]Computer Centre, Gakushuin University, Tokyo 171-8588, Japan

{w20223@e., mb67015@e., y20029@e., y20210@e., miyares18@, y-suzuki@, uchida@}hiroshima-cu.ac.jp,
ori-ies18@tk.cc.gakushuin.ac.jp

Abstract: We examine knowledge acquisition from graph structured data, which is an important task in machine learning and data mining. Two-terminal series parallel (TTSP) graphs are used as data models for electric networks and scheduling. We propose an improved evolutionary learning system of multiple TTSP graph structured patterns from positive and negative TTSP graph data. A newly implemented graph pattern display system of our learning method improves the comprehensibility of an evolutionary learning process. As its performance evaluation we report extensive experimental results on various parameter settings of our evolutionary learning system.

Keywords: evolutionary learning, genetic programming, sets of graph structured patterns

1. Introduction

The quantity of graph structured data has recently been increasing, and attention is being drawn to machine learning and data mining methods that can process graph structured data. Two-terminal series parallel (TTSP) graphs (Fig.1) are used as data models for electric networks and scheduling. TTSP graph patterns [1] (Fig.2) are special types of TTSP graphs some edges of which are replaced with special edges called variables, and represent graph structured patterns having structured variables.

We focus on the graph classification problem [2] specialized for TTSP graph data and consider the following learning problem (Section 3). Given positive and negative TTSP graph data w.r.t. a specific phenomenon, the problem is to obtain a characteristic graph pattern, which matches many positive TTSP graph data and few negative TTSP graph data. In our previous work [3, 4] on this problem, we reported a learning method for acquiring characteristic multiple graph structured patterns by evolutionary computation using sets of TTSP graph patterns as individuals, from positive and negative TTSP graph data. However, the comprehensibility of an evolutionary learning process by the previous learning system is low since it has no function for displaying the graph representations of individuals manipulated by evolutionary operators. The performance evaluation of the previous learning system using parameter settings was also insufficient.

In this paper, we propose an improved evolutionary learning system of multiple TTSP graph structured patterns from positive and negative TTSP graph data. A newly implemented graph pattern display system (Section 4) of our learning method considerably improves the comprehensibility of an evolutionary learning process. As its performance evaluation we report extensive experimental results (Section 5) on our evolutionary learning system and show that some features of this system are effective in obtaining characteristic multiple TTSP graph patterns.

In the proposed method, we use as a subprocedure our previous learning method [5, 6], which is a Genetic Programming (GP) based method for acquiring characteristic single TTSP graph patterns from positive and negative TTSP graph data. Genetic Programming is an evolutionary method which is an extension of Genetic Algorithm and deals with tree structured objects such as formulas and programs [7, 8].

As a multiple graph structured pattern, we use a set of TTSP graph patterns, in order to deal with the graph data that a single graph pattern is inappropriate to represent. If at least one of a set of TTSP graph patterns matches a TTSP graph, we say that the set of TTSP graph patterns matches the TTSP graph. Our learning method, as a main procedure for acquiring characteristic sets of TTSP graph patterns from positive and negative TTSP graph data, is a two-stage evolutionary learning method.

We discuss related work. Our evolutionary learning framework is different from the approaches for learning TTSP graph patterns from positive TTSP graph data [1, 9] in that it learns from both positive and negative TTSP graph data. Genetic Network Programming (GNP) [10] and Graph Structured Program Evolution (GRAPE) [11] are known for evolutionary learning from graph structured data. A related combinatorial problem is also known [12].

This paper is organized as follows. In Section 2, we introduce sets of TTSP graph patterns as multiple graph structured patterns. In Section 3, we review our data mining problem and our method for solving the problem, that is, an evolutionary learning method for acquiring characteristic sets of TTSP graph patterns as individuals, from positive and negative TTSP graph data. In Section 4, we report that a newly implemented graph pattern display system of our learning method considerably improves the comprehensibility of the evolutionary learning processes. In Section 5, we report extensive experimental results on our evolutionary learning system to show that some features of our system are effective in obtaining characteristic multiple TTSP graph patterns. In Section 6, we give concluding remarks.

2. Sets of TTSP Graph Patterns as Multiple Graph Structured Patterns

In this section, we first review TTSP graph patterns (also called TTSP term graphs) [1], which are used for representing characteristic graph structured patterns, according to [1, 5]. We then introduce sets of TTSP graph patterns as multiple graph structured patterns. A *multidag* is a directed connected graph that allows multiple edges and does not contain any cycle. A multidag is said to be *two-terminal* if it has exactly one source and one sink. A *TTSP graph* is a two-terminal multidag by recursively applying “series” and “parallel” operations [13, 14]. Examples of TTSP graphs and series and parallel operations are illustrated in Fig.1.

TTSP graph patterns are graph patterns obtained by replacing some edges of TTSP graphs with variables, where variables are special lists of two vertices. A variable in a TTSP graph pattern can be replaced with any TTSP graph. Every edge and variable in a TTSP graph pattern is labeled with some words from specified languages. We assume that all variable labels in a TTSP graph pattern are different.

For a TTSP graph pattern π and a TTSP graph G , we say that π matches G if G can be obtained from π by replacing all variables in π with appropriate TTSP graphs. A *parse tree* of a TTSP graph pattern or a TTSP graph is defined as a tree representation of operations that define the TTSP graph pattern or the TTSP graph. In a parse tree of a TTSP graph pattern or a TTSP graph, an edge labeled with “p” represents an application of parallel operation and an edge labeled with “s” represents an application of series operation. It has been

shown that there is a one-to-one correspondence between a TTSP graph pattern and its parse tree [1]. For a TTSP graph pattern π , we denote by $t(\pi)$ the corresponding parse tree of π . A parse tree of a TTSP graph or a TTSP graph pattern has the structure of a rooted tree with ordered or unordered children.

GP is an evolutionary method that deals with tree structured objects. Therefore, GP approach is suited for learning the structural features of TTSP graph data by using parse trees of TTSP graph patterns as individuals [5, 6] (Fig.3).

We give examples of a TTSP graph pattern in Fig.2 and a parse tree in Fig.3. In the figures of a TTSP graph pattern or a parse tree, variables are represented by boxes with lines to its elements. In Fig.2, the TTSP graph G_3 is obtained from the TTSP graph pattern g by replacing variables labeled with x and y in g with TTSP graphs G_1 and G_2 , respectively. Then we see that the TTSP graph pattern g matches the TTSP graph G_3 . In Fig.3, for example, we describe the parse trees $t(\pi_1)$ and $t(\pi_2)$ of TTSP graph patterns π_1 and π_2 , respectively.

In order to represent sets of TTSP graphs more precisely, we consider a set of TTSP graph patterns to be a multiple graph pattern. A set of TTSP graph patterns $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ matches a TTSP graph G if π_i matches G for some i ($1 \leq i \leq n$).

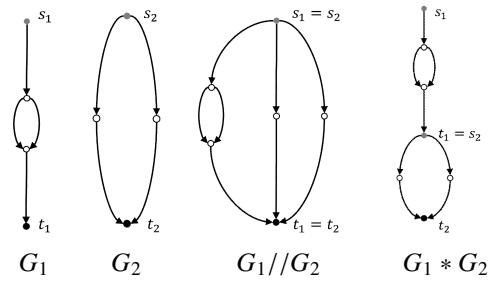


Fig. 1: TTSP graphs $G_1, G_2, G_1//G_2$ (obtained by parallel operation on G_1, G_2) and G_1*G_2 (obtained by series operation on G_1, G_2).

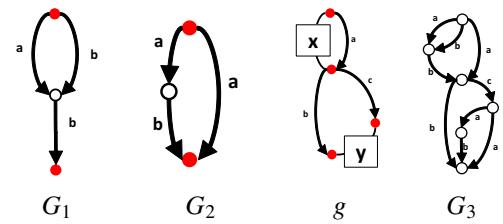


Fig. 2: TTSP graphs G_1, G_2, G_3 and TTSP graph pattern g .

3. Acquisition of Characteristic Sets of TTSP Graph Patterns

In this section, we review a learning method [3, 4] for acquiring characteristic sets of TTSP graph patterns from positive and negative TTSP graph data by using an evolutionary method using sets of TTSP graph patterns as individuals, in order to explain the parameter settings on which we report the experimental results in Section 5. Our learning method for acquiring characteristic sets of TTSP graph patterns uses as a subprocedure the previous learning method [5] for acquiring characteristic single TTSP graph patterns as individuals from positive and negative TTSP graph data.

First, we describe Problem of Acquiring Characteristic Single TTSP Graph Patterns and the method for solving the problem. Let D be a finite set of positive and negative TTSP graph examples. The *fitness* of a TTSP graph pattern π w.r.t. D , denoted by $\text{fitness}_D(\pi)$, is defined as the balanced accuracy of π w.r.t. D , that is, $\text{fitness}_D(\pi) = (\text{the ratio of positive examples in } D \text{ that } \pi \text{ matches} + \text{the ratio of negative examples in } D \text{ that } \pi \text{ does not match}) / 2$. A TTSP graph pattern as an individual is a binary classifier of TTSP graph data. Therefore, a TTSP graph pattern of high fitness is considered a characteristic single TTSP graph pattern, which matches many positive and few negative TTSP graph data. We define the problem of finding a TTSP graph pattern of high fitness w.r.t. a given set of TTSP graph examples as follows.

Problem of Acquiring Characteristic TTSP Graph Patterns[5, 6]

Input: A finite set D of positive and negative TTSP graph examples.

Problem: Find a TTSP graph pattern of high fitness w.r.t. D .

We proposed the GP method for solving Problem of Acquiring Characteristic TTSP Graph Patterns [5, 6]. The genetic operator crossover of the GP method is shown in Fig.3. This figure shows the application of crossover, as an example of the genetic operation, to two TTSP graph patterns π_1 and π_2 and the two corresponding parse trees $t(\pi_1)$ and $t(\pi_2)$.

Second, we describe Problem of Acquiring Characteristic Sets of TTSP Graph Patterns and the method for solving the problem. For a set A , we denote by $|A|$ the number of elements in A . In a manner similar to the setting of single TTSP graph patterns, the *fitness* of a set Π of TTSP graph patterns w.r.t. D , denoted by $\text{fitness}_D(\Pi)$, is defined as the balanced accuracy of Π w.r.t. D , that is, $\text{fitness}_D(\Pi) = (\text{the ratio of positive examples in } D \text{ that } \Pi \text{ matches} + \text{the ratio of negative examples in } D \text{ that } \Pi \text{ does not match}) / 2$. A set of TTSP graph patterns as an individual is a binary classifier of TTSP graph data. So a set of TTSP graph patterns of high fitness is considered a characteristic set of TTSP graph patterns, which matches many positive and few negative TTSP graph data. We define the problem of finding a set of TTSP graph patterns of high

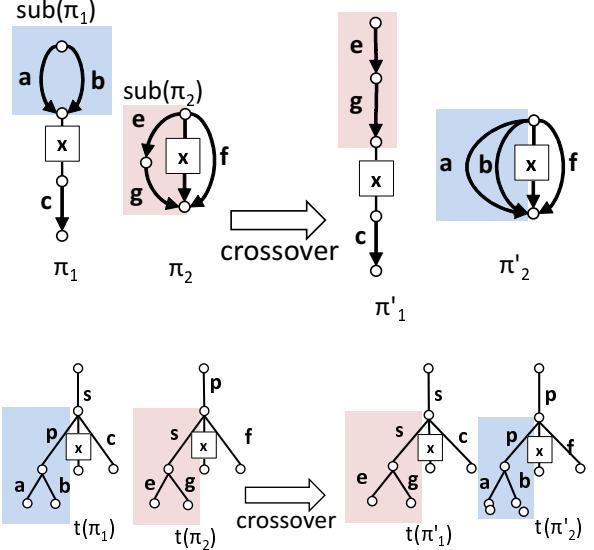


Fig. 3: Genetic operator crossover on TTSP graph patterns π_1 and π_2 . Genetic operator crossover on corresponding parse trees $t(\pi_1)$ and $t(\pi_2)$.

fitness w.r.t. a given set of TTSP graph examples as follows.

Problem of Acquiring Characteristic Sets of TTSP Graph Patterns:[3, 4]

Input: A finite set D that consists of a set P of positive TTSP graph examples and a set N of negative TTSP graph examples, and a positive integer c ($1 \leq c < |P|$).

Problem: Find a set Π ($1 \leq |\Pi| \leq c$) of TTSP graph patterns of high fitness w.r.t. D .

We proposed an evolutionary method [3, 4] for solving Problem of Acquiring Characteristic Sets of TTSP Graph Patterns that uses a set (represented as a tuple) of TTSP graph patterns as an individual. The evolutionary learning method as a main procedure for acquiring characteristic sets of TTSP graph patterns from positive and negative TTSP graph data is based on the two-stage evolutionary methods for obtaining characteristic sets of other tree or graph patterns [15, 16, 17].

We explain an outline of our method for Problem of Acquiring Characteristic Sets of TTSP Graph Patterns. If the positive data have a diversity, sets of TTSP graph patterns are more appropriate for explaining data than single TTSP graph patterns. Let Pos_j be the j -th cluster of the set P of positive TTSP graph examples obtained by performing a clustering algorithm, and D_j be the union of Pos_j and the set N of negative TTSP graph examples. The method performs the j -th GP learning process GPL_j for each j ($1 \leq j \leq c$), as a sub-process, for obtaining single TTSP graph patterns π with the fitness w.r.t. D_j ($\text{fitness}_{D_j}(\pi)$) given the input set D_j . The fitness of a TTSP graph pattern π after the current generation in GPL_j is called the *basic fitness* of π . Let $\{\pi_j^1, \pi_j^2, \dots, \pi_j^k\}$

be a set of TTSP graph patterns with k highest basic fitness w.r.t. D_j in GPL_j after a current generation for each j ($1 \leq j \leq c$), where k is a positive integer. An individual in the main procedure is a tuple (a representation of a TTSP graph pattern set) $[\pi_1^{\ell_1}, \pi_2^{\ell_2}, \dots, \pi_c^{\ell_c}]$ of TTSP graph patterns such that $1 \leq \ell_j \leq k$ for each j ($1 \leq j \leq c$). For each TTSP graph pattern π_j in the GP learning process GPL_j , let $n(\pi_j)$ be the number of occurrences of an isomorphic TTSP graph pattern to π_j as the j -th element of a TTSP graph pattern tuple in a current population of the main procedure. The *added fitness* of π_j is defined as $C_{add} \times \frac{n(\pi_j)}{b}$, where C_{add} is a positive real constant (the maximum of added fitness) and b is the population size of the main procedure. We define the fitness of the TTSP graph pattern π_j in the GP learning process GPL_j before the next generation as the sum of the basic fitness of π_j and the added fitness of π_j . A TTSP graph pattern π_j with larger value of $\frac{n(\pi_j)}{b}$ is a better component in a TTSP graph pattern set of high fitness. Therefore, we add $C_{add} \times \frac{n(\pi_j)}{b}$ to the fitness w.r.t. D_j ($fitness_{D_j}(\pi)$) of the TTSP graph pattern π_j before the next generation of the GP subprocess GPL_j , in order to increase the chance of survival of the TTSP graph pattern π_j in GPL_j and obtain a better set of TTSP graph patterns in the whole learning process.

4. The Graph Pattern Display System of an Evolutionary Learning Method of Multiple TTSP Graph Structured Patterns

In this section, we report a newly implemented graph pattern display system of our evolutionary learning method of multiple TTSP graph patterns from positive and negative TTSP graph data. In order to improve the comprehensibility of an evolutionary learning process and revise evolutionary operators and parameter settings, we need to check the graph representations of individuals manipulated by evolutionary operators in a whole evolutionary learning process. However, our previous learning system [3, 4] has no function for displaying graph patterns as individuals manipulated by evolutionary operators. Therefore, we implemented a graph pattern display system of our evolutionary learning method.

The evolutionary learning system stores a log file of a whole process of an evolutionary run. After the system completes an evolutionary learning process, the graph pattern display system shows graphical representations of multiple TTSP graph patterns (TTSP graph pattern sets) as individuals of the main procedure, TTSP graph patterns as individuals of the subprocedures (subroutines) and the whole evolutionary learning process. The implementation is in Java and uses Swing and Graphviz.

We show windows of the graph pattern display system in Figs.4 to 11. The window in Fig. 4 displays parse trees of TTSP graph patterns as individuals in the subprocedures, and

the window in Fig. 5 displays the TTSP graph patterns represented by the parse trees. The window in Fig. 6 displays individuals in the main procedure, which are sets of parse trees of TTSP graph patterns, and the window in Fig. 7 displays the corresponding individuals in the main procedure, which are sets of TTSP graph patterns represented by the parse trees. If we choose (Fig. 8) a TTSP graph pattern or a parse tree of a TTSP graph pattern in windows of the graph pattern display system, the graph pattern display system shows a window displaying the large graphical representation of the chosen TTSP graph pattern or the chosen parse tree of a TTSP graph pattern and the evolutionary process (Fig. 9). The graph pattern display systems also shows enlarged representations of a parse tree of TTSP graph pattern (Fig. 10) and the corresponding TTSP graph pattern (Fig. 11) represented by the parse tree.

As Figs.4 to 11 show, the graph pattern display system considerably improves the comprehensibility of an evolutionary learning process of obtaining characteristic multiple TTSP graph patterns from positive and negative TTSP graph data.

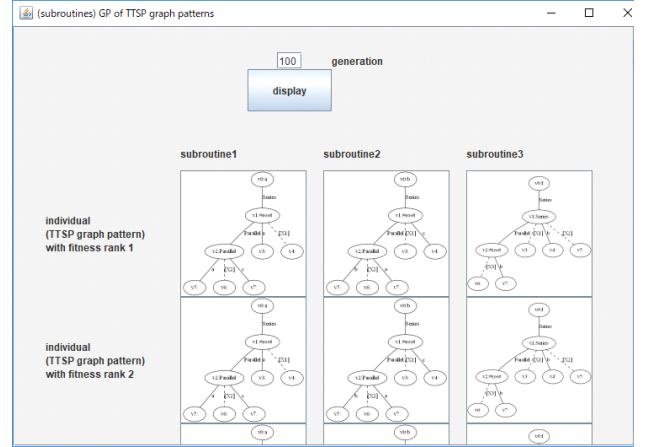


Fig. 4: Window displaying parse trees of TTSP graph patterns as individuals in subprocedures.

5. Experimental Results

In this section, we report extensive experimental results on our evolutionary learning system of multiple TTSP graph patterns from positive and negative TTSP graph data, in order to show the effectiveness of our learning system and its parameter settings. In our previous work [3, 4], we reported some experimental results with a fixed setting of parameters of our learning system. Our implementation of the learning system is in Java on a Windows 10 Pro 64bit system.

In order to perform experiments with synthetic data, we considered a target set Π_0 of TTSP graph patterns. We regarded randomly generated TTSP graphs that Π_0 matches as positive examples and randomly generated TTSP graphs that

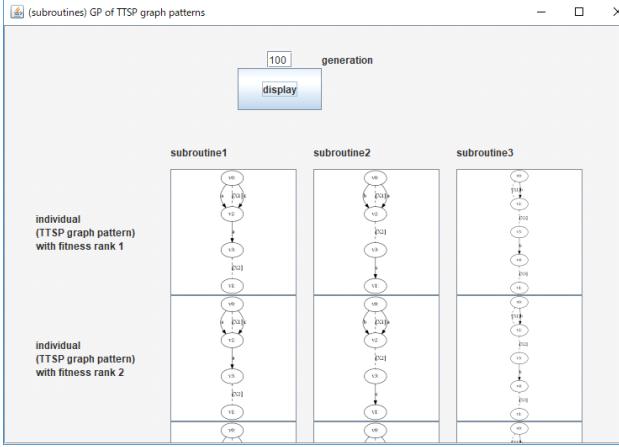


Fig. 5: Window displaying TTSP graph patterns as individuals in subprocedures.

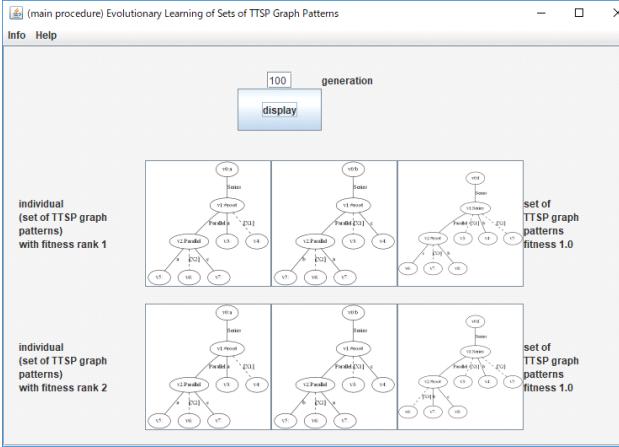


Fig. 6: Window displaying sets of parse trees of TTSP graph patterns as individuals in main procedure.

Π_0 does not match as negative examples. In the experiments reported here, we considered the target set $\Pi_0 = \{\pi_1, \pi_2, \pi_3\}$ of TTSP graph patterns, which is illustrated in Fig. 12. Then we had a set D consisting of 1000 positive examples and 1000 negative examples. Some positive and negative TTSP graphs are shown in Fig. 13.

Using the data set D , we performed 10 experimental runs. To calculate the fitness of single or multiple TTSP graph patterns as individuals, we used a matching algorithm [1] for TTSP graph patterns and TTSP graphs. The parameters of our evolutionary learning and GP settings are shown in Table 1. We performed experiments by changing the following 4 important parameters I_{red} , C_{add} , k , c and report the run time and the fitness of individuals in each parameter setting. The parameter I_{red} shows whether a selected set of individuals from a subprocedure is redundant ($I_{red} = red$) or not ($I_{red} = non-red$), that is, $I_{red} = red$ means that two or

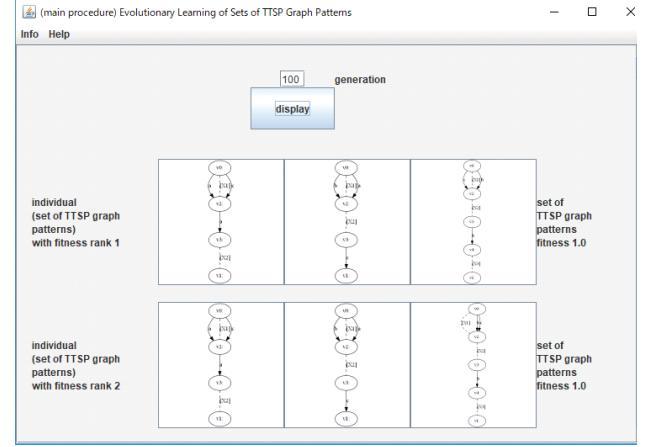


Fig. 7: Window displaying TTSP graph pattern sets as individuals in main procedure.

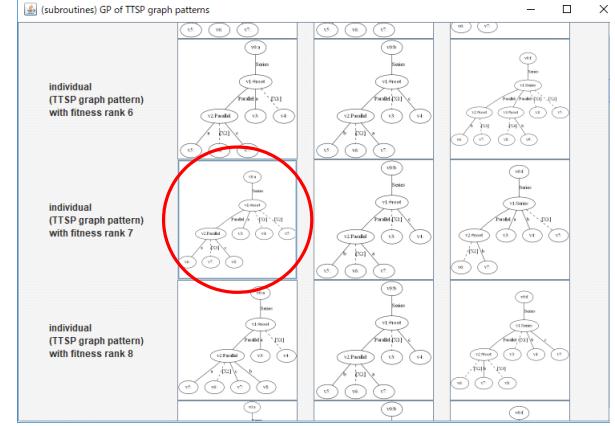


Fig. 8: Window displaying chosen parse tree (indicated by red circle) of TTSP graph pattern in subprocedure.

more isomorphic TTSP graph patterns may be included in each selected set of individuals from a subprocedure. The parameter C_{add} denotes the maximum value of added fitness. The parameter k denotes the number of selected single TTSP graph patterns with high fitness from a subprocedure. The parameter c denotes the number of clusters, which is an argument of the problem of acquiring characteristic sets of TTSP graph patterns.

In the experiments reported here, we used the following parameter settings: $I_{red} \in \{red, non-red\}$, $C_{add} \in \{0.1, 0.2, 0.3\}$, $k \in \{5, 8, 10\}$ and $c = 3$. Fig. 15 shows the average values of the 10 runs of the fitness of the individual with the highest fitness in each generation with $(I_{red}, C_{add}, k, c) = (non-red, 0.1, 5, 3)$ and $(I_{red}, C_{add}, k, c) = (red, 0.1, 5, 3)$. Table 2 shows the average values of fitness of the individuals with the highest fitness in the final generation (called the *best individuals*), and the average run time for each parameter setting. Fig. 14 shows

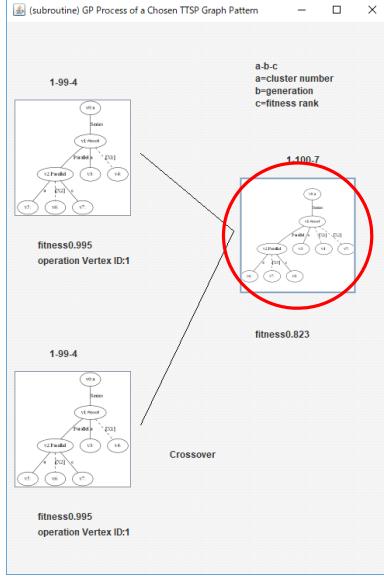


Fig. 9: Window displaying GP process of chosen parse tree (indicated by red circle) of TTSP graph pattern in subprocedure.

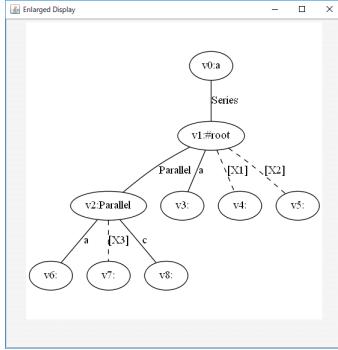


Fig. 10: Window displaying enlarged parse tree of TTSP graph pattern in subprocedure.

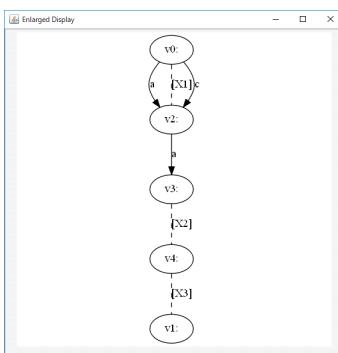


Fig. 11: Window displaying enlarged TTSP graph pattern in subprocedure.

Table. 1: Fixed Parameters of evolutionary learning and GP in this experiment

Parameters of evolutionary method as main procedure	
Population size	50
Maximum number of generations	200
Parameters of GP as subprocedures	
Population size	50
Reproduction probability	0.05
Inversion probability	0.05
Crossover probability	0.45
Mutation probability	0.45
Selection method	tournament size 4
Roulette wheel selection	elite size 3
Maximum number of generations	200

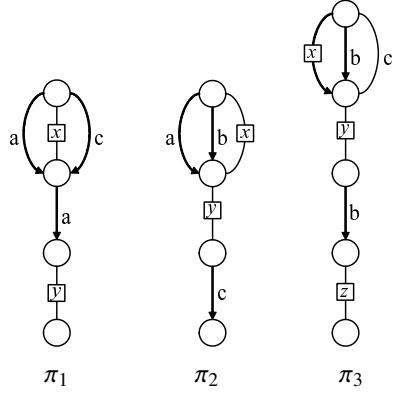
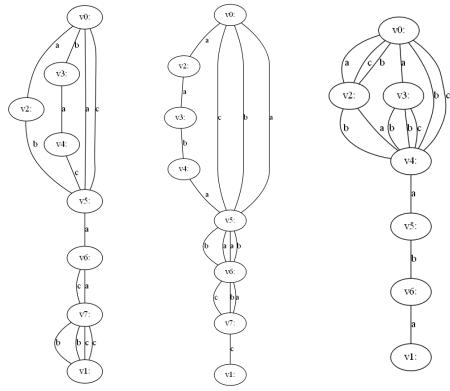


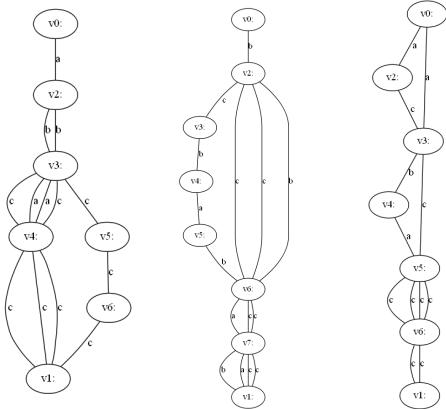
Fig. 12: Set $\Pi_0 = \{\pi_1, \pi_2, \pi_3\}$ of TTSP graph patterns to generate synthetic data.

examples of the best individuals obtained by the proposed and previous methods.

From all experimental results (Table 2), for all parameter settings, our proposed method is shown to obtain successfully characteristic TTSP graph pattern sets of higher fitness than single TTSP graph patterns, from the synthetic data of positive and negative TTSP graphs. By comparing the fitness of parameter settings (Table 2) with $I_{red} = red$ and $non-red$, for example the parameter settings (1) and (10), and from Fig. 15, we can say that the redundancy check of a selected set of individuals from a subprocedure is shown to be effective in obtaining characteristic TTSP graph pattern sets. By comparing the fitness of parameter settings (Table 2) that have different values for C_{add} and the same value for the other parameters, for example the parameter settings (1),(2) and (3), the added fitness is also shown to be effective in obtaining characteristic TTSP graph pattern sets.



positive TTSP graphs



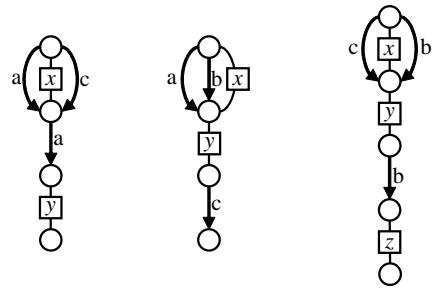
negative TTSP graphs

Fig. 13: Positive and negative TTSP graphs.

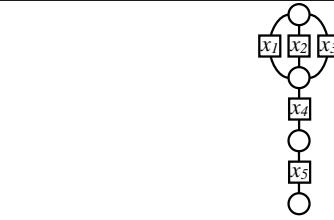
6. Concluding Remarks

In this paper, we proposed an improved evolutionary learning system of multiple TTSP graph structured patterns from positive and negative TTSP graph data. Our learning method is a two-stage evolutionary learning method and incorporates as a subprocedure our previous GP method for acquiring characteristic single TTSP graph patterns from positive and negative TTSP graph data. We showed that a graph pattern display system of our learning method considerably improves the comprehensibility of a whole evolutionary learning process. We reported extensive experimental results on our evolutionary learning system and showed that the redundancy check of a selected set of individuals from a subprocedure and the added fitness in the two-stage evolutionary learning method are effective in obtaining characteristic multiple TTSP graph patterns with high fitness. We plan to apply our method to real data having TTSP graph structures.

Acknowledgments. We would like to thank the anonymous referees for their helpful comments. This work is partly supported by Grant-in-Aid for Scientific Research (C) (Grant Numbers JP15K00312, JP15K00313 and JP17K00321) from



best individual (TTSP graph pattern set)
of fitness 1.000 in 9th run by proposed method



best individual (TTSP graph pattern)
of fitness 0.836 in 1~4 and 6~10 runs by previous method

Fig. 14: Best individual (TTSP graph pattern set) by proposed method, and best individual (TTSP graph pattern) by previous method.

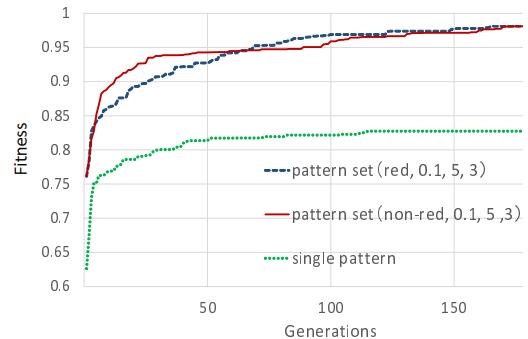


Fig. 15: Fitness of best individuals with $(I_{red}, C_{add}, k, c) = (red, 0.1, 5, 3)$ and $(I_{red}, C_{add}, k, c) = (non-red, 0.1, 5, 3)$

Japan Society for the Promotion of Science.

References

- [1] R. Takami, Y. Suzuki, T. Uchida, and T. Shoudai, “Polynomial time inductive inference of TTSP graph languages from positive data,” *IEICE Transactions on Information and Systems*, vol. E92-D(2), pp. 181–190, 2009.
- [2] S. Rehman, A. Khan, and S. Fong, “Graph mining: A survey of graph mining techniques,” *Proc. International Conference on Digital Information Management (ICDIM) 2012*, pp. 88–92, 2012.
- [3] Y. Yamagata, T. Miyahara, Y. Suzuki, T. Uchida, F. Tokuhara, and T. Kuboyama, “Acquisition of multi-

Table. 2: Average values of fitness of best individuals and run time for each parameter setting

No.		fitness	run time
(0)	single pattern	0.8326	69.03
pattern set			
parameter setting			
No.	I_{red}, C_{add}, k, c	fitness	run time
(1)	$red, 0.1, 5, 3$	0.9813	159.95
(2)	$red, 0.2, 5, 3$	0.9863	163.84
(3)	$red, 0.3, 5, 3$	0.9775	159.26
(4)	$red, 0.1, 8, 3$	0.9869	281.74
(5)	$red, 0.2, 8, 3$	0.9887	268.83
(6)	$red, 0.3, 8, 3$	0.9806	277.55
(7)	$red, 0.1, 10, 3$	0.9783	408.54
(8)	$red, 0.2, 10, 3$	0.9935	487.49
(9)	$red, 0.3, 10, 3$	0.9765	494.82
(10)	$non-red, 0.1, 5, 3$	0.9857	738.82
(11)	$non-red, 0.2, 5, 3$	0.9881	730.59
(12)	$non-red, 0.3, 5, 3$	0.9891	748.72

ple graph structured patterns by an evolutionary method using sets of TTSP graph patterns as individuals,” *Proc. IIAI AAI 2017*, pp. 459–464, 2017.

- [4] Y. Yamagata, T. Miyahara, Y. Suzuki, F. Tokuhara, T. Uchida, and T. Kuboyama, “Using depth label sequences in acquisition of characteristic multiple TTSP graph patterns,” *Proc. 2017 IEEE SMC Hiroshima Chapter Young Researchers’ Workshop (In Japanese)*, pp. 73–77, 2017.
- [5] S. Nagai, T. Miyahara, Y. Suzuki, and T. Uchida, “Acquisition of characteristic TTSP graph patterns by genetic programming,” *Proc. IIAI AAI 2012*, pp. 340–344, 2012.
- [6] Y. Otsuka, Y. Suzuki, T. Miyahara, and T. Uchida, “An improvement of the method for acquiring characteristic ttsp graph patterns by genetic programming,” *Proc. of Annual Conf. JSAT 2013 (In Japanese)*, 2L1–3, 2013.
- [7] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [8] R. Poli, W. Langdon, and N. McPhee, *A Field Guide to Genetic Programming*. Lulu Press, 2008.
- [9] T. Kono, Y. Suzuki, T. Uchida, and T. Miyahara, “Enumerating maximally frequent TTSP graph patterns,” *Proc. 7th Workshop on Learning with Logics and Logics for Learning (LLL)*, pp. 43–50, 2011.
- [10] H. Katagiri, K. Hirasawa, and J. Hu, “Genetic network programming - application to intelligent agents,” *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 3829–3834, 2000.
- [11] S. Shirakawa, S. Ogino, and T. Nagao, “Graph structured program evolution,” *Proc. Genetic and Evolutionary Computation Conference (GECCO) 2007*, pp. 1686–1693, 2007.

- [12] R. Carr, S. Doddi, G. Konjevod, and M. Marathe, “On the red-blue set cover problem,” *Proc. SODA 2000*, pp. 345–353, 2000.
- [13] K. Takamizawa, T. Nishizeki, and N. Saito, “Linear-time computability of combinatorial problems on series-parallel graphs,” *Journal of the Association for Computing Machinery*, vol. 29, pp. 623–641, 1982.
- [14] J. Valdes, R. Tarjan, and E. Lawler, “The recognition of series parallel digraphs,” *SIAM J. Comput.*, vol. 11(2), pp. 298–313, 1982.
- [15] S. Nakai, T. Miyahara, Y. Suzuki, T. Kuboyama, and T. Uchida, “Acquisition of characteristic sets of tree patterns with VLDC’s using genetic programming and edit distance,” *Proceedings of 2014 IEEE 7th International Workshop on Computational Intelligence and Applications (IWCIA 2014)*, pp. 113–118, 2014.
- [16] S. Tani, T. Miyahara, Y. Suzuki, and T. Uchida, “Acquisition of multiple tree structured patterns by an evolutionary method using sets of tag tree patterns as individuals,” *Proc. IIAI AAI 2015*, pp. 213–218, 2015.
- [17] F. Tokuhara, T. Miyahara, T. Kuboyama, Y. Suzuki, and T. Uchida, “Acquisition of multiple block preserving outerplanar graph patterns by an evolutionary method for graph pattern sets,” *Proc. IWCIA 2017*, pp. 191–197, 2017.

Modeling of driving force generated by rod wheel with single rod on loose soil

Taku Matsumoto [†] Keita Nakamura [‡] Keitaro Naruse [†]

[†] The University of Aizu, Aizuwakamatsu, Japan

[‡] Revitalization Center, LICTiA, The University of Aizu, Aizuwakamatsu, Japan

d8201105@u-aizu.ac.jp, keita-n@u-aizu.ac.jp, naruse@u-aizu.ac.jp

Abstract: Our research group has developed small weeding robots to move in a paddy field. This robot has an effect not to grow weeds. The robot has two rod wheels to realize an above effect. However, slip and stuck occur easily in the loose soil such as a rice field. Therefore, in this study, we analyze the driving force generated by this rod wheel from the viewpoint of rotational speed in order to control the robot. In other words, this paper shows the relation between a loose soil and driving force generated by the rod wheel.

Keywords: Modeling, Wheel, Driving force

1. Introduction

The soil and water in the paddy field are stirred when the ducks swim. It has the following effects:

- (1) Suppressing the growth of weeds by shutting out from the sunlight;
- (2) Floating the seeds of weeds from the soil surface.

In other words, rice-duck farming has a weeding effect. However, this method has high costs for breeding ducks and protecting them from vermin. Many organic farmers avoid rice-duck farming for this reason. To reduce the work burden for organic farmers, many weeding robots have been developed in recent years. One research project [1, 2] developed a robot that adopted an approach which is similar to the rice-duck farming. However, this robot can inflict great damage to the plants when it strikes them because it is heavy. Therefore, this robot is expensive because it requires a camera unit to recognize the rice plants to avoid striking them. Generally, a medium- or large-sized robot causes problems by crushing or pulling up rice plants while passing over them.

Our research group has developed a small weeding robot (Fig.1) inspired from rice-duck farming[3, 4]. This robot has two rod wheels (Fig.2) that are rotated with a DC motor. Our robot can drive such as forward, back, and turn by forward/reverse control for two motors. Rod wheels stir the soil and water in the paddy field simultaneously. Thereby weeding in a similar way to rice-duck farming. Our research group showed experimentally in an actual paddy field that this robot does remove weeds[5].

The ultimate goal of our robot is an automatic control and autonomous navigation. This robot has two rod wheels to

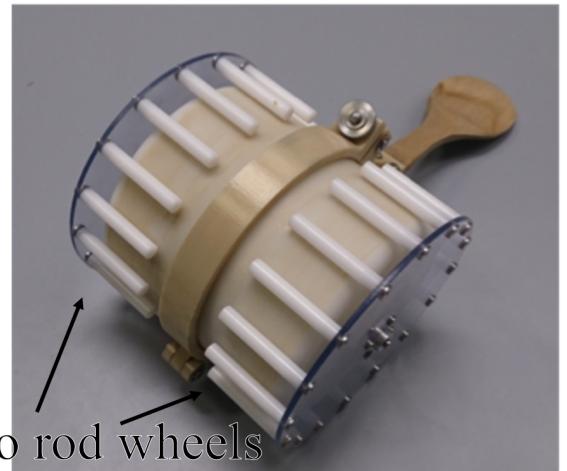


Fig. 1: Our weeding robot.

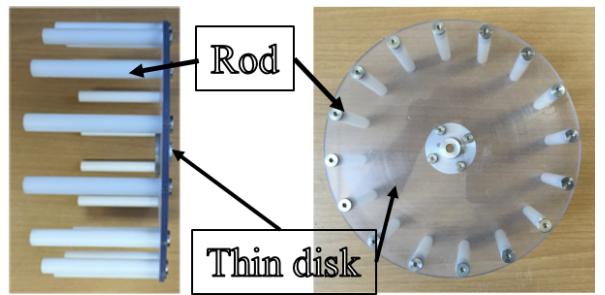


Fig. 2: Top and side views of a rod wheel.

run in the paddy field. For controlling the robot, a model of driving force for a rod wheel is needed. However, the model does not yet exist. Therefore, in this thesis, we model driving force for a rod wheel in order to clarify the mechanism of driving force generated by the wheel in case of the sand.

A flat wheel[6, 7] and lugged[8] are given as examples of models of driving force for a wheel on the loose soil. However, their models cannot apply to a rod wheel because these wheels differ in shape from the wheel. Moreover, a lug of a lugged wheel stirs the soil completely. On the other hand, a rod of a rod wheel always stirs the soil weakly. Therefore, we need to develop new models of driving force for a rod wheel.

In previous research[9], accuracy of the model was low because we assumed that no friction and no direction. Furthermore, the authors expect a waveform of driving force expected sine-wave, they observed forward triangle-wave. The conventional model was a simple model without considering the characteristics of sand. However, Actual sand has a characteristic that shear stress is decided according to compressive stress, and it is a linear model with offset. We express actual phenomenon than the current model by considering the shear stress of the sand. About above phenomenon, we can explain that a waveform of actual values is forward triangle-wave by offset.

In this research, driving force generated by a rod wheel is the reaction force received from the sand, or the friction force received from the sand. However, in this thesis, we do not consider a friction between a disk of a rod wheel and the sand and the gravity of a rod and the disk. We develop an experimental system, obtaining actual driving forces, and compare expected values given from a model and actual values in each condition. The accuracy rate of a proposed model in this thesis is higher than a simple model. Their results are indicated from the precision of the model.

The structure of this paper is as follows. Section 2 presents our models of driving force considering the shear stress characteristics of the sand. Section 3 explains an experimental system and results. Section 4 discusses accuracy of modeling by comparing a simple model, a proposal model, and actual values. Finally, section 5 shows the conclusion of this paper.

2. Modeling

In this section, we model the driving force generated by rod type wheels. We think that the driving force generated by rod type wheels is caused by the shear stress of sand. In addition, modeling from the motor driving force we considered as a comparison target will also be explained. These models are described in their respective subsections.

2.1 Simple model B

When a rod wheel rotate, a single rod rotate at same time too. Moreover, when a rod touches the soil, it generates a force. We define this force as driving force. In this subsection, we introduce our proposal model of the driving force. Figure 3 shows this model.

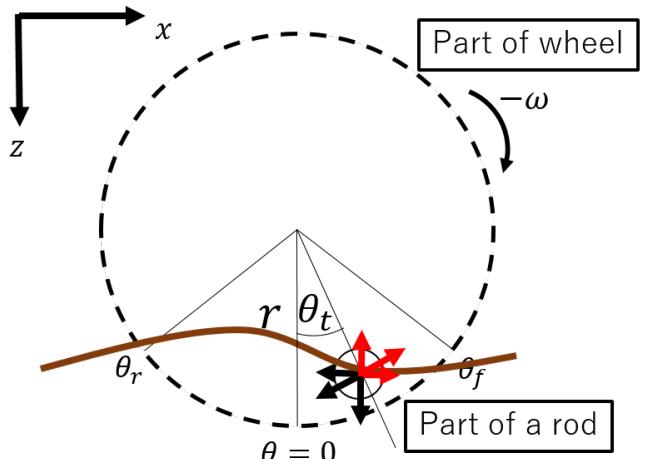


Fig. 3: Model B.

The motor torque is calculated by multiplying the torque constant of the motor and the motor current per unit of time, and reduction ratio expressed by

$$f_m = K_t I * K, \quad (1)$$

where f_m is the motor torque, K is reduction ratio of gear, k_t is torque constant of the motor, and K is reduction of each gears.

Here, the force generated by a motor f_m obtained by dividing motor torque τ_m by distance between the center of a rod wheel and the center of a rod as following equation.

$$\tau_m = f_m/r. \quad (2)$$

When the rod wheel rotates, the soil puts on the rod after the rod enters the soil and. Therefore, we consider that the gravity which is received from the soil on a rod. Here, we model the gravity on this subsection (Fig.4). In order to model the gravity from the soil on a rod, we need to calculate the volume of the soil. At first, the depth on a contact angle of the rod θ is defined from a contact angle of the rod and maximum depth of the rod. In this model, we assume that the soil do not deform. The depth on a contact angle of a rod $h(\theta)$ is expressed by

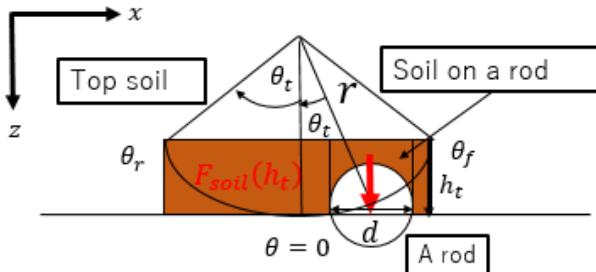
$$h(\theta) = r(\cos(\theta) - \cos(\phi)), \quad (3)$$

where r is the distance between the center of a rod wheel and the center of a rod and ϕ is the incident angle to the soil.

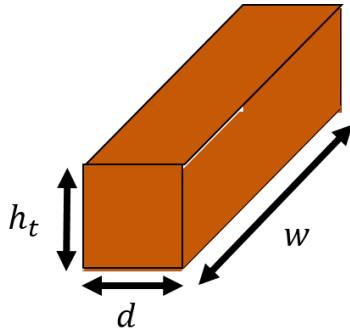
The volume of the soil on a rod V is defined by the depth on a contact angle of the rod, the diamiter of the rod d , the width of the rod w as a below equation.

$$V = rdw \quad (4)$$

Finally, the surface area of a rod receive the gravity of the soil. Therefore, the gravity received against the rod surface



(a) Model of the gravity of the soil on a rod.



(b) Volume of the soil on a rod.

Fig. 4: Model of the gravity of the soil on a rod.

is expressed by

$$\sigma_g(\theta) = \rho V g / (d * w), \quad (5)$$

where $\sigma_g(\theta)$ is the gravity received against the rod surface, g is the gravity acceleration and ρ is the specific gravity of the sand.

Figure 4 shows that the gravity of the soil $h(\theta)$ is calculated by actual values.

Model B F^B is decomposed into x-axis component and z-axis component as follows.

$$F^B(\theta) = \begin{bmatrix} F_x^B(\theta) \\ F_z^B(\theta) \end{bmatrix} = \tau_m \begin{bmatrix} \cos \theta \\ -\sin \theta \end{bmatrix} \quad (6)$$

X-component and z-component of the force are F_x^B and F_z^B . Here, we determine the positive direction of each components is the direction of the direction of rotation of a rod wheel and the vertically downward direction.

2.2 Model A1

Figure 5 shows model A1. In this model, modeling is performed assuming compression occurs in the z-axis direction. Therefore, shear stress $\tau_f(\sigma)$ is expressed by

$$\tau_f(\sigma) = c + \sigma_g(h) \tan \phi, \quad (7)$$

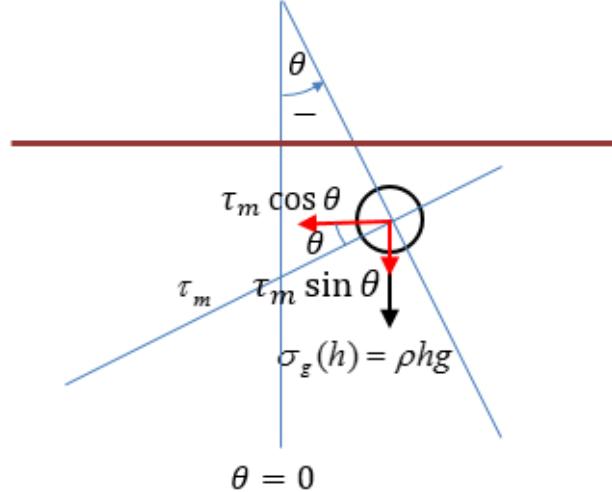


Fig. 5: Model A1.

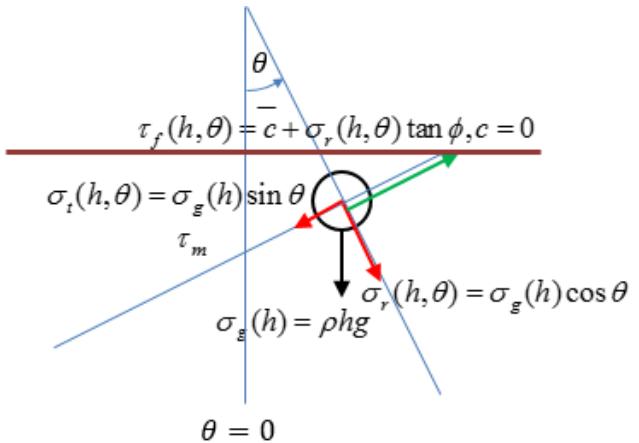


Fig. 6: Model A2.

where c is cohesion stress and ϕ is friction angle. Model A1 F^{A1} is decomposed into x-axis component and z-axis component as follows.

$$F_x^{A1}(\theta) = \begin{cases} \tau_f(\sigma) & \text{if } \tau_m \cos \theta > \tau_f(\sigma) \\ \tau_m(\sigma) \cos \theta & \text{otherwise} \end{cases} \quad (8)$$

$$F_z^{A1} = \tau_m \sin(\theta) - \sigma_g(h) \quad (9)$$

2.3 Model A2

Figure 6 shows model A2. In this model, the force generated on the rod shall be two. Compressive stress due to the gravity of the upper sand by the rod σ_g and the reaction force of the sand on the rod or friction force τ_f . The maximum frictional force of the sand is determined by the compressive stress in the radial direction against the sand. The force measured by the sensor is the resultant of σ_g and τ_f .

$$\tau_f(\sigma) = c + \sigma \tan(\phi) \quad (10)$$

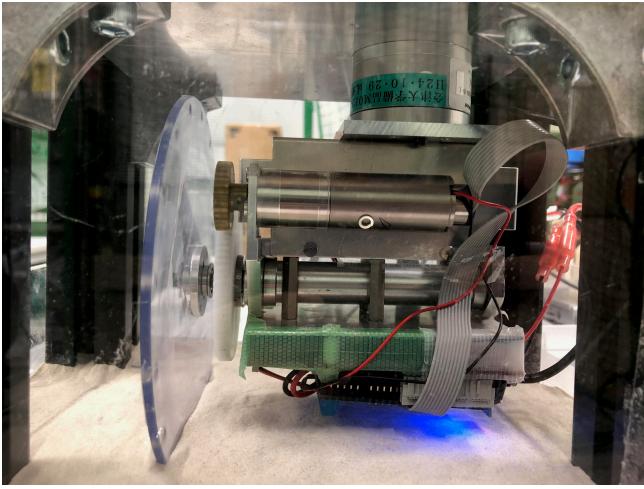


Fig. 7: Overview of the experimental system

Model A2 F^{A2} is decomposed into x-axis component and z-axis component as follows.

$$F^{A2}(\theta) = \begin{bmatrix} F_x^{A2} \\ F_z^{A2} \end{bmatrix} = \tau_f(\sigma_g \cos(\theta)) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (11)$$

Model A1 and model A2 are proposed models considering shear stress characteristics of the sand.

3. Wheel rotation experiment

In order to verify our proposal models in a previous section, we experiment for rotating the rod wheel. This experiment focuses the rod wheel with single rod. Moreover, this section shows an experimental system and the experimental condition.

3.1 Experimental system

Figure 7 shows an overview of our experimental system. This system has a motor, a control system, and a F/T sensor. This control system sends a control signal to the motor for rotating a rod wheel with single rod. Moreover, we install F/T sensor at the top of the unit and it measures driving forces occurred by the wheel[10]. This sensor can obtain 3-axis forces and 3-axis torques. In order to obtain actual values accurately, we use Tohoku Silica Sand No.7 (Fig.8) which is known as components in an experimental environment. Angle of a rod wheel is calculated by a wheel angular velocity measured from an motor encoder.

3.2 Experimental method

We set the distance from a soil surface to the center of the rod to 10 mm. Also, we flatten the soil surface each time to set same conditions before measuring. We set measurement time is 20 [s]. In the first ten seconds, a rod wheel do not rotate in order to measure initial values of 3-axis forces obtained



Fig. 8: Tohoku Silica Sand No.7.

Table. 1: Simulation parameters.

parameters	values	units
Wheel radius r	0.084	[m]
Rod diameter d	0.008	[m]
Rod width w	0.060	[m]
Specific gravity of the sand ρ	$2.57 * 10^3$	[kg/m ³]
Gravitational acceleration g	9.8	[m s ⁻²]
Depth h	0.010	[m]
Torque constant k_t	0.0157	[N]
Gear ratio	117.3	
Motor current I	0.060	[A]
Cohesion stress c	1	[kPa]
Friction angle ϕ	35.0	[deg]
Time t	[0.0:20.0]	[s]

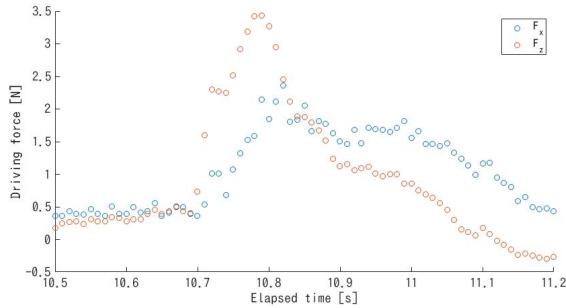
from a six-axis sensor. After ten seconds, we obtain them of it when the wheel rotates. We experiment 20 times under above experimental conditions for comparing actual values and our proposal models. Figure 9 shows example data of each forces and motor current obtained from 6-axis sensor and an motor encoder in this experiment. Table 1 is simulation parameter.

3.3 Results

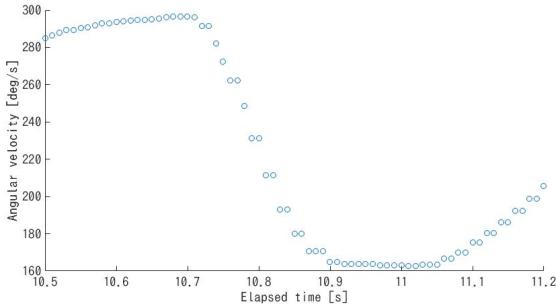
Figure 10 shows results of each models and an actual value. The vertical line shows driving forces of each model and an actual value and the horizontal one rotation angles. Moreover, values of each model divide by the area multiplied by a rod radius and rod width.

4. Discussions

In this section, we discuss a waveform and an value of the driving force model from experimental results. In the subsection, consider the force in the x-axis direction and z-axis

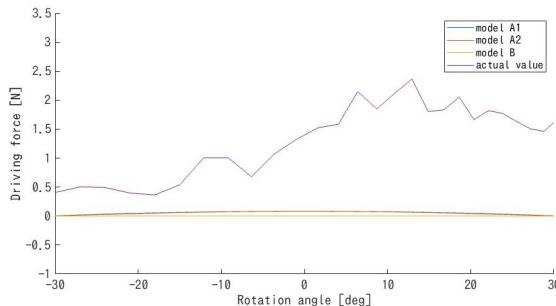


(a) Driving force F_x and F_z obtained from 6-axis sensor.

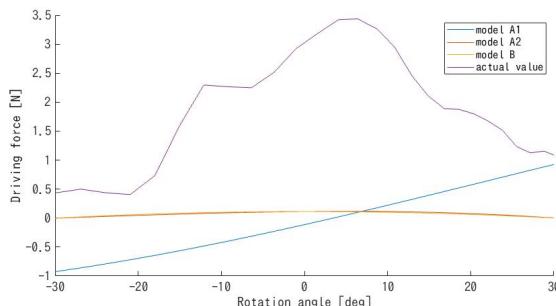


(b) Rotation velocity calculated from a motor encoder.

Fig. 9: Example of an actual data.



(a) Each models F_x and an actual value.



(b) Each models F_z and an actual value.

Fig. 10: Each model and an actual values.

direction separately.

4.1 x-axis direction

The force in the x-axis direction of the measured value has a sinusoidal waveform. As the common point of each model, the value of each model is smaller than the measured value. Moreover, the waveform of X-axis direction of an actual value is backward-triangle wave. This is due to the rod gradually pushing the sand backwards. Due to the above reason, the driving force increases until a rod escapes the sand.

4.2 z-axis direction

The force in the z axis direction of the measured value can be observed as if it is the waveform of the backward slanting triangular wave. The similar waveform is only model A 2, and it can be said that model A 2 is suitable among the proposed models. However, as with the force in the x-axis direction, the value of the driving force is smaller than the measured value.

Before the rod enters the sand, a bias is applied in the + direction with respect to the x-axis direction and the z-axis direction. It is considered that this is always influenced by the disk part on which the rod type ground contacts. This paper is based on not considering the disk part of the rod type wheel. Therefore, we need to model driving force generated by a disk of the wheel.

5. Conclusion

In this paper, the driving force generated by the rod type wheel is modeled considering the shear stress characteristics of the sand. In order to evaluate this model, we also developed an experimental apparatus that acquires measured values. This model considers the shear stress characteristics of sand.

In the future, in order to raise the accuracy of the model more, consider the amount of movement of sand which was not considered in this research as well. We plan to improve model accuracy by comparing future models with measured values.

References

- [1] Mitsui, T., Kobayashi, T., Kagiya, T., Asano, Y., Bando, N., "Research on the advanced technology for organic agriculture development of rice field rover for weeding out", pp. 33–36, 2004.
- [2] Fujii, K., Tabata, K., Yokoyama, T., Hirayu, H., Endo, Y., "Development of a small weeding robot (AIGAMO ROBOT) for paddy fields", 15th Research Report of Gifu Prefectural Research Institute of Information Technology, pp. 32–34, 2014.
- [3] Maruyama, A., Naruse, K., "Feasibility study of weeding robots in rice fields inspired by natural ducks", Proceedings of the 8th International Conference on Bio-inspired Information and Communications Technologies, pp. 378–381, 2014.

- [4] Maruyama, A., Naruse, K., "Development of small weeding robots for rice fields", Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies, pp. 99–105. 2014.
- [5] Nakamura, K., Kimura, M., Anazawa, T., Takahashi, T., Naruse, K., "Investigation of weeding ability and plant damage for rice field weeding robots", System Integration (SII), 2016 IEEE/SICE International Symposium on, pp. 899–905, 2016.
- [6] Bekker, M. G., "Off-the-road locomotion: research and development in terramechanics", University of Michigan Press, 1960.
- [7] Wong, J. Y., Reece, A. R., "Prediction of rigid wheel performance based on the analysis of soil-wheel stresses part I. Performance of driven rigid wheels", Journal of Terramechanics, Vol. 4, No. 1, pp. 81–98, 1967.
- [8] Mizukami, N., Iizuka, K., Kunii, Y., "Discussion on validity of lugged wheel model and evaluation method", System Integration (SII), 2014 IEEE/SICE International Symposium on, pp. 597–602, 2014.
- [9] Matsumoto, T., Oyama, Y., Ogawa, J., Nakamura, K., Naruse, K., "Mechanism of generating drawbar pull of rod wheel on loose soil", Artificial Life and Robotics, Vol. 22, No. 4, pp. 503–508, 2017.
- [10] MinebeaMitsumi-Inc, "Discontinuance of 6-axis force sensor OPFT/MinebeaMitsumi Inc Sensing Device BU", <http://www.minebea-mcd.com/en/product/6-axial/opft.html>, Accessed Oct. 2018.

The Substitute Worker Selection Method Using Agent-based Simulation

Kohei Hatamoto [†] Soichiro Yokoyama [†] Tomohisa Yamashita [†] Hidenori Kawamura [†]

[†] Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan
hatamotok@complex.ist.hokudai.ac.jp

Abstract: Shift scheduling has been studied in various fields for many years, but little consideration has been given to absence, which occurs after determining schedule. Recent years have seen some research on rostering, but most of it tries to correct schedules with minimum cost. While substitute attendant selection requires an enormous amount of labor, this operation has not been taken into account. In this research, we create an agent-based simulation environment and investigate substitute attendant selection methods for some simple cases.

Keywords: agent-based simulation, substitute worker, scheduling, rostering

1. Introduction

The worker scheduling problem has been studied for many years in various fields. This problem is always prevalent in business and constitutes a matter of concern. Nurse scheduling is a typical example. In recent years, reducing the burden of shift management has attracted considerable attention because the number of part-time workers, who want to work at a time convenient to them, is increasing.

Many previous works addressed ways to optimize schedules under given constraints. Many systems assist manager shift creation based on such research. Other than the problems mentioned above, shift adjustment with absenteeism presents another important issue. Absenteeism refers to the impossibility of attending work due to a sudden non-business-related occurrence on the scheduled work day for a predetermined shift. For example, for a business such as a call center, whose workers are predominantly housewives, it is conceivable that a worker may be unable to attend work as expected because of her child contracting fever. A sudden absence is thus inevitable in an industry employing part-time workers.

When absenteeism occurs, managers need to ask other workers to work on behalf of those who are absent. This task is burdensome for managers, and comprises a mental component as well a time component. The mental component relates to anxiety about whether the shift will be filled in time, and the time component entails the time taken by managers to call other workers and request them to fill in for the shift. In particular, the burden on time is heavy, and often, all the working hours of the manager's day may be consumed by this task. Therefore, in this research, we considered a method that can complete this task more efficiently.

The burden of asking workers to work on behalf of absentees is not as heavy for small offices. It is therefore necessary

to increase the efficiency of this task at offices where it is cumbersome for managers to ask hundreds of other workers to fill in for absentees. Therefore, in this research, we focused on examining an efficient request method that can be used at large offices. We validated an algorithm to this effect by simulation because applying the method in practice without confirming its effectiveness is risky; it could result in an unintended increase in labor, on the contrary.

We constructed an appropriate request method for shift management when absenteeism occurs. In order to confirm the effectiveness of the request method, we used agent-based simulation to address various occurrences of absenteeism.

The remained of this article is organized as follows. Section 2 reviews previous studies on shift scheduling. Section 3 describes the simulation system used in our experiments. Section 4 introduces methods for asking workers to work on behalf of those who are absent. The simulation results appear in Section 5 and Section 6 concludes this paper.

2. Literature review

2.1 Staff scheduling

Research on staff scheduling has been widely conducted over the decades[1]. Nurse scheduling, that is, deciding the work schedule of a nurse in a hospital, is representative among many derivations. In nursing scheduling, it is necessary to create a work schedule that satisfies many conditions, such as nurses' desired work schedules or the number of workers required at the ward. Many studies on nurse scheduling have been conducted[2][3].

In general, staff scheduling is formulated as a combinatorial optimization problem. Because scheduling problem is NP-hard, studies on solving the optimization problem have

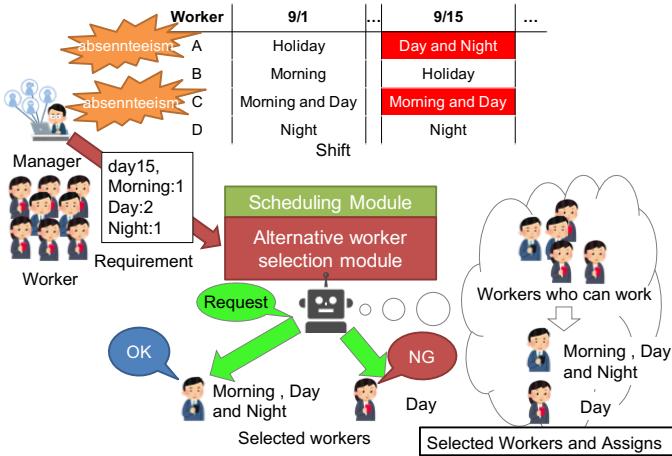


Fig. 1: System proposed in this study for verification of alternative attendance request methods

been conducted [4][5][6]. Recent advances in computers have made it possible to identify strict solutions for large-scale problems.

2.2 The rerostering problem

Staff scheduling addresses cases of absenteeism after the work schedule has been created, and this topic has received much attention in recent years. This is called the rerostering problem. Despite the frequent occurrence of abrupt absences by nurses at hospitals due to sudden illness or family issues, research on rerostering gained prominence only recently[7]. However, conventionally, research in this area was aimed at revising work shifts at the lowest cost when absenteeism occurred[8][9][10], and requests to workers to fill in for absentees, which is a task that requires to be completed at such times, were not considered. Therefore, in this research, we focused on making requests for substitute workers when absenteeism occurs.

3. The simulation system

The system proposed in this study to perform verification of alternative attendance request methods via simulation is shown in figure1. Next, each module and model within the proposed system are described.

3.1 Scheduling module

This section describes the scheduling module. The scheduling module considers the number of people required by the manager as inputs, and provides the scheduled work shift as the output.

3.1.1 Shift roster

This section describes the shift roster handled by the proposed system. Most of the work shifts used in the actual office adopt one hour as the unit. It is not impossible to handle such work shifts within a simulation but doing so makes the system extremely complicated, making it more difficult to identify a problem when it occurs. In this study, the shift for one day is subdivided into three parts: morning, daytime, and night. To analyze the trends of workers' preferences for work shifts and to compare the acceptance rate for each time period when such requests occur, this study presents a partitioning method that is not too complex.

It is necessary to set unnatural work shifts for workers. Specifically, we needed to exclude less realistic schedules, such as working in the morning, taking a break in the day, and working again in the evening. In order to realize this, shift assignment of workers was decided in terms of combinations of working time slots for the day rather than with regard to the time period. Here, we considered seven combinations: "morning, day, and night," "morning and day," "day and night," "only morning," "only day," "only night," and "holiday." Limiting shift assignment to this combination makes it possible to eliminate unnatural shift assignment. Various types of projects/work are undertaken in real-life offices, and it is common that different workers take charge of different tasks depending on the type of project. Some workers can also handle multiple projects. In order to consider this situation, this system created a shift table for each project.

The shift table is shown in Table1 . The shift table in Table1 shows that Worker 1 will work all day, Worker 2 will work in the morning and afternoon, and Worker 3 will work only at night. Each worker can be allotted only one time slot per day. Furthermore, workers cannot work at multiple projects on the same day. The shift for one entire project for one month comprises the sum of the one-day shifts allotted to that worker.

3.1.2 Request for schedule

This section describes the format of the request for scheduling according to the schedule table defined in the previous section. The worker chooses time slots for each day he wishes to attend

Table. 1: A Day-shift Roster for a Project

Time	Worker 1	Worker 2	Worker 3
Morning, Day, and Night	✓		
Morning and Day		✓	
Day and Night			
Only Morning			
Only Day			
Only Night			✓

Table. 2: Example of a worker's preferred schedule

Time	Preference
Morning, Day, and Night	
Morning and Day	✓
Day and Night	
Only Morning	✓
Only Day	✓
Only Night	
Holiday	

work as well as the holidays. The desired schedule is treated as the work shift timetable for that worker within the desired time slot. For example, if a worker chooses to work in time slot "Morning and Day," he/she can work within the time slots "Only Morning" and "Only Day" only. The corresponding schedule request is shown in Table 2. In addition, shift request is not distinguished for each project, and workers can be placed in any project.

3.1.3 Demand for workers

When scheduling is performed, information on how many people should be placed in a certain time slot is a necessary input. This is termed as worker demand. As per the shift table of this system, managers can decide upon the number of workers required in several ways. In this research, we set the number of workers for each morning, afternoon, and evening by considering how we may determine the number of workers necessary to complete the work at hand in real-world situations. For example, if three workers are needed in the morning, the manager needs a total of three people who are supposed to work in time slots including "Morning."

3.1.4 Scheduling with mathematical optimization

The schedule needs to satisfy workers' preference for their schedules and managers' demand for workers, as described in previous sections. In order to create a shift satisfying these conditions, we used mathematical optimization. The constraints in the proposed mathematical optimization at this time are as follows.

- (a) Do not allow workers non-preferred work times/shifts.
- (b) The number of workers to be placed is as low as possible.
- (c) Do not allow a worker to work for seven consecutive days.
- (d) Workers retain the skills necessary to take charge of a project.
- (e) Do not allow workers to work in multiple time slots a day.

We adopted the lowest number of constraints so as to avoid customization for a specific condition. With regard to conditions (a) and (b), we allowed a slight violation if the preferred shift cannot be granted because of worker shortage. Even if the number of workers is insufficient, managers must decide the schedule, as is the case in real-life offices.

In this research, the problem was formulated as a mixed integer linear programming problem, and a solution was obtained using a general-purpose solver. The details of the formulation are shown below.

$$\text{minimize} \quad 1000 \sum_{d \in D} \sum_{e \in E} \sum_{p \in P} \sum_{t \in T} r_{det} x_{dept} + 10 \sum_{d \in D} \sum_{p \in P} \sum_{t \in T} l_{dpt}$$
(1)

s.t

$$x_{dept} = \{1, 0\} \quad d \in D, e \in E, p \in P, t \in T$$
(2)

$$y_{de} = \{1, 0\} \quad d \in D, e \in E$$
(3)

$$r_{det} = \{1, 0\} \quad d \in D, e \in E, p \in P$$
(4)

$$N_{dpt} \in \mathbb{N} \quad d \in D, e \in E, p \in P, t \in T$$
(5)

$$l_{dpt} \leq \sum_{e \in E} x_{dept} - N_{dpt} \quad d \in D, p \in P, t \in T$$
(6)

$$l_{dpt} \leq -(\sum_{e \in E} x_{dept} - N_{dpt}) \quad d \in D, p \in P, t \in T$$
(7)

$$\sum_{p \in P} x_{dept} \leq 1 \quad d \in D, e \in E, t \in T$$
(8)

$$\sum_{t \in T} \sum_{p \in P} x_{dept} \leq 1 \quad d \in D, e \in E$$
(9)

$$y_{de} = \sum_{t \in T} \sum_{p \in P} x_{dept} \quad d \in D, e \in E$$
(10)

$$\sum_{d \in S} y_{de} \leq 6 \quad e \in E, S \in D_6$$
(11)

In these equations, P is a set of projects, T is a set of time slots, E is a set of workers, D is a set of days, and D_6 is a set of combinations of six days in a row. The variable x_{dept} represents whether worker e is assigned to shift of project p in time slot t on day d , y_{de} represents whether worker e is assigned to shift on day d , r_{det} represents whether worker e chooses to work in time slot t on day d , N_{dpt} represents the number of demand of project p in time slot t on day d , and l_{dpt} represents the difference between the number of demand and assigned workers of project p in time slot t on day d . Many general-purpose solvers exist for solving linear programming problems, and we used Cbc (Coin-or branch and cut)[11]. Cbc is a freely available solver.

3.2 Alternative worker selection module

This module selects alternate attendants who satisfy the constraints. As per the content of the constraint, the worker will not work in time slots in which absenteeism occurs, and

Table. 3: Utility value of the student class

	Morning, Day, and Night	Morning and Day	Day and Night	Only Night	Only Day	Only Morning	Holiday
Mon	5	10	10	30	20	10	30
Tue	5	10	10	30	20	10	30
Wed	5	10	10	30	20	10	30
Thu	5	10	10	30	20	10	30
Fri	5	10	10	30	20	10	30
Sat	20	30	30	40	40	40	5
Sun	20	30	30	40	40	40	5

Table. 4: Utility value of the housewife class

	Morning, Day, and Night	Morning and Day	Day and Night	Only Night	Only Day	Only Morning	Holiday
Mon	40	50	50	20	20	20	2
Tue	40	50	50	20	20	20	2
Wed	40	50	50	20	20	20	2
Thu	40	50	50	20	20	20	2
Fri	40	50	50	20	20	20	2
Sat	5	10	10	30	20	10	50
Sun	5	10	10	30	20	10	50

he/she does not work continuously for seven days if he/she works on behalf of an absent worker. The former constraint is necessary to determine whether filling in for an absentee is possible. The latter constraint is set because as per the law, workers are not permitted to work for more than seven consecutive days without a holiday.

3.3 Agent models

This section defines the models used for the simulation.

3.3.1 Worker model

The worker model represents workers, who select the days they want to work and their responses to an attendance request. Attributes of workers in actual offices are often diverse, but a certain trend can be deciphered for each day job. Therefore, we proposed a worker model consisting of three day job classes with different parameter distributions. We called these classes “student,” “housewife,” and “part-timer” because of the characteristics of the parameters, as described later. The behavior definition of these classes and the attributes used in the models are described later.

The worker model is thus intended to check the workers’ requests for their schedules and select whether the worker works on a particular day. Workers request a one-month schedule at a time. The format of the schedule request is described in Section 3.1.2. In this model, we set the utility for each shift’s time slot for each day of the week and modeled the action as a request schedule that maximizes utility.

In order to depict that each professional class has a certain tendency, each class was allotted a different base for the utility value. This utility value was set for each day of the week and shift time slot. The utility value V of each class is shown Table 3.

The utility value was obtained by multiplying the utility

Table. 5: Utility value of the part-timer class

	Morning, Day, and Night	Morning and Day	Day and Night	Only Night	Only Day	Only Morning	Holiday
Mon	40	50	50	20	20	20	2
Tue	40	50	50	20	20	20	2
Wed	40	50	50	20	20	20	2
Thu	40	50	50	20	20	20	2
Fri	40	50	50	20	20	20	2
Sat	40	50	50	20	20	20	2
Sun	40	50	50	20	20	20	2

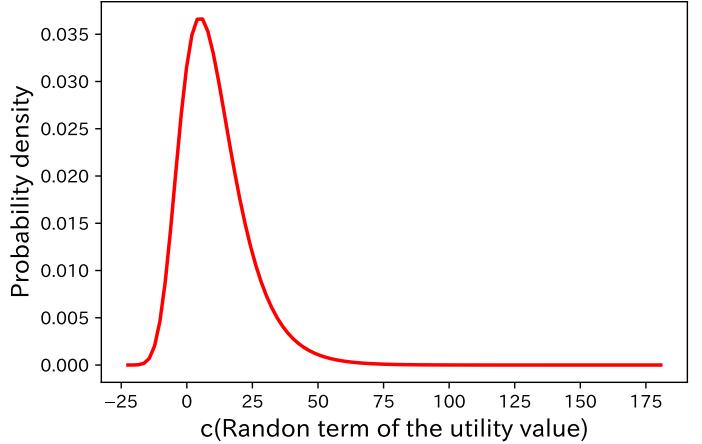


Fig. 2: Probability density function of the Gumbel distribution

value for each class by the weight for each individual and adding the random value c . The weight W was a 7×7 uniform random number matrix with the range [0, 1.5]. The utility value of each time slot for a certain day d was obtained by the following equation.

$$V_d W_d + c \quad (12)$$

where c follows the Gumbel distribution ($\beta = 10, u = 5$), which is often used in discrete choice experiments. Figure 2 shows the probability density function of the Gumbel distribution.

Next, a model that considers replies to requests is described. Replying to a request consists of the following two steps:

- notice contact
- return answer

There are various ways to establish contact, but for this study, we assumed communication by telephone. Whether to notice contact is determined by the probability set for each class. Table 6, Table 7, and Table 8 show the probability of establishing contact for each class.

Setting a different probability for each time or day of the week, as seen in these tables, expresses the fact that some worker classes are easily contactable for specific time slots,

Table. 6: Probability of contacting for the student class

	Morning	Day	Night
Mon	0.2	0.8	0.8
Tue	0.2	0.8	0.8
Wed	0.2	0.8	0.8
Thu	0.2	0.8	0.8
Fri	0.2	0.8	0.8
Sat	0.4	0.4	0.8
Sun	0.4	0.4	0.8

Table. 7: Probability of contacting for the housewife class

	Morning	Day	Night
Mon	0.9	0.8	0.5
Tue	0.9	0.8	0.5
Wed	0.9	0.8	0.5
Thu	0.9	0.8	0.5
Fri	0.9	0.8	0.5
Sat	0.9	0.4	0.5
Sun	0.9	0.4	0.5

whereas others are not. In this work, students are difficult to contact in the mornings and on Saturdays and Sundays, while housewives are difficult to contact at noon on Saturdays and Sundays.

The model used for returning answers is described next. In this research, we assumed that acceptance of a request is related to the schedule request, and acceptance is decided by using the utility value of each time slot in the schedule request. Specifically, when V_r , obtained by adding a random value to the utility value of a certain time slot, exceeds a threshold value, the request is accepted. An interview with a large call center hiring many workers revealed that workers are more likely to receive requests as the request date draws closer to the day on which absenteeism occurs because their schedules are often fixed. In order to express this fact, the utility value V_r decreases further away from the absence day. We set the decreasing width so that it varies stepwise by the number of days from the absence day. This stepwise function is shown

Table. 8: Probability of contacting for the part-timer class

	Morning	Day	Night
Mon	0.8	0.8	0.8
Tue	0.8	0.8	0.8
Wed	0.8	0.8	0.8
Thu	0.8	0.8	0.8
Fri	0.8	0.8	0.8
Sat	0.8	0.8	0.8
Sun	0.8	0.8	0.8

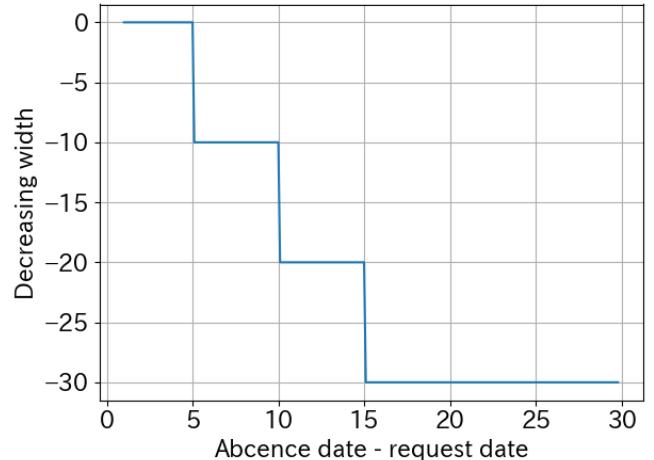


Fig. 3: Decreasing width

in Fig3.

3.4 Manager model

The manager model represents a person requesting a worker to accept a substitute attendance. In the real world, differences in compatibility exist between each worker and the administrator/manager, but in this research, we modeled the manager as the request function to prevent complicating the models and their analysis.

4. Proposed method

This section explains the assumptions employed for the method.

Various types of information can represent absenteeism situations. Some of these types may be as follows.

- Number of workers who are absent
- Date when absence occurs
- Kind of project for which the absence occurs
- Number of managers
- Number of workers managers can call at one time

If the number of absentees is one, the manager simply searches for one person who can substitute for the absentee. On the other hand, in the case of multiple absenteeism occurrences, additional information, such as the order of requesting, should be considered. As for the date on which the absence occurs, there is no other way to request a substitute immediately if the absenteeism occurs on the day of work itself, but if the managers recognize the absence in advance, there is a method to fill in for the absentee over several days. Absenteeism can occur for multiple project types, and some workers can handle specific projects. Thus, it is necessary

to consider which workers should be requested to fill in for the absentee depending on the concerned project type. In the event two or more managers exist, it is necessary to consider compatibility between managers and workers. When asking multiple people to fill in for the absentee at the same time, consideration of a parallel number is a must.

The problem of selecting alternative workers is not as simple as it seems, as these examples show. Therefore, this study considered using the request method in a limited number of situations. Specifically, it deals with two cases, wherein (1) the notification of absenteeism is received from one worker on the day of work and (2) when two cases of absenteeism are known in advance. The following section explains the details of the simulation environment for these two cases and describes the algorithm to be applied.

4.1 Receiving a notification on the day of work

When a worker's absence is known on the work day itself, it is necessary to arrange for a substitute worker as soon as possible. Therefore, it is a good idea to contact the person who is most likely to receive the request to fill in when the absence becomes known. In this research, we compared several algorithms to determine the order of such requests.

We assumed that the probability of response in each time slot is given in this method. Since our model set a correlation between workers desiring attendance and accepting requests, we investigated the difference in the average number of people requested to fill in, depending on whether requests for the schedule were taken into account. The proposed request order determination methods are of seven types.

Method 1: In descending order of probability of replying.

Method 2: Adhering to the worker request schedule for the day is the highest priority.

Method 3: In descending order of probability of replying. /

In case of the same order, adhering to the worker request schedule for the day is the highest priority.

Method 4: Adhering to the worker request schedule for the day has the highest priority. / For the same priority, the descending order of probability of replying is the highest priority.

Part-timer: Part-timer has priority.

Student: Student has priority.

Random: Random order.

4.2 Two absences known in advance

In this case, it is necessary to decide when to request substitution for each case of absenteeism. If two absences occur on the same day, this problem can be considered as one where the manager needs to replenish two workers per day. In this research, we considered several days' difference between two absent days. Possible request methods for such cases are as follows.

- Simultaneously: Requesting substitutes for two absences at one time several days before the previous absence.
- Respectively: Requesting substitutes several days before each absence.
- In Combination: Requesting substitutes for two absences at one time several days before the previous absence until a substitute for the previous one is found.

Common to each method, priority is given to those who can take charge of both projects where the absenteeism occurs. An overview of the order of requests for each method is shown in Figure4.

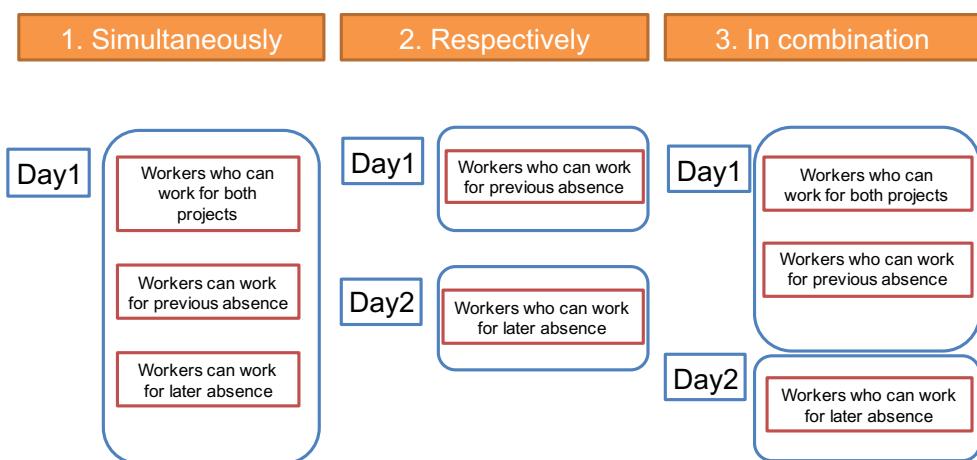


Fig. 4: Overview of the order of requests for each method

Table. 9: Conditions

Condition	Previous Absence	Later Absence (5-day interval)	Later Absence(10-day interval)
quiet period	60%	40%	25%
usual period	30%	20%	13%
busy period	1%	under 1%	under 1%

We investigated the difference in the average number of people asked to substitute by changing the conditions for two absences for these methods. Three conditions were used: types of project, interval between absence days, and percentage of people who receive the request. Depending on the proportion of people who receive the request, we demarcated the period as a quiet period, a usual period, or a busy period. The percentage of people who receive requests under each condition is as shown in Table9. We considered two types of projects: the same and different. We chose two intervals, namely 5 days and 10 days, between the two absentee days.

5. Experiments

This section shows the results of the simulation, which was conducted using the method described in the previous section.

First, consider the common setting of the simulation environment. The number of days in the schedule was assumed to be 30 and the number of workers was 170. Two projects were considered, with 85 people and 45 people assigned to the first and second, respectively. In addition, 40 workers can handle both projects. In order to carry out a simulation as close to reality as possible, the necessary number of persons in the schedule was considered to be the number of workers required for the schedule used by TMJ ,Inc. , which is a large-scale call center.

5.1 Simulation of receiving a notification on the day of absenteeism

In this setting, the absence day can be located anywhere within the 30 days, and we simulated a case where the absenteeism occurred in the morning for all days for 30 days for one schedule. To eliminate variations due to schedule, absence was generated for 100 types of schedules generated by the scheduling module. In this experiment, we compared the average number of workers being asked to fill in until the substitute worker is found when one instance of absenteeism occurred. The results are shown in Table10. Table10 shows that all the methods show improvement over randomly asking workers to fill in. Comparing Method 1 and Method 2 shows that considering the scheduling preference of the workers is more effective at reducing the number of workers being asked despite the convenience of using the telephone for the latter. The results of Method 3 and Method 4 show that the order

Table. 10: Average number of workers being asked

Method	Average number of workers being asked
Method1	4.36
Method2	3.70
Method3	3.44
Method4	3.39
Part-timer	4.42
Student	14.42
Random	4.87

of consideration does not influence the number of people considerably. As per the results of Method 1 for students and part-timers, it can be said that the evaluation of ease of calling each time is important.

5.2 Simulation of the case where two absences are known in advance

The average number of workers being asked to fill in for an absentee using each method is shown in Tables11, 12, and 13. Table11 shows that when the interval of absentee days is small, it is preferable to request substitutes for all projects simultaneously. The “in combination” method also performs well, but using the method “simultaneously” allows the manager to close the request task within one day. When the interval of absentee days is large, the “combination” method becomes the method of choice as the number of clients requesting projects simultaneously increases. As Table12 shows, when only a few people receive a request, all the methods perform equally as there is no difference in the type of projects between the methods “in combination”

Table. 11: Average number of workers being asked to fill in for an absentee in a quiet period

Condition	Simultaneously	Respectively	In combination
same project, 5-day interval	4.2	4.2	3.1
same project, 10-day interval	7.0	4.2	3.5
different project, 5-day interval	4.1	4.1	3.1
different project, 10-day interval	7.2	4.2	3.5

Table. 12: Average number of workers being asked in the usual period

Condition	Simultaneously	Respectively	In combination
same project, 5-day interval	11.1	9.6	8.5
same project, 10-day interval	17.4	9.9	8.9
different projects, 5-day interval	12.2	10.3	8.0
different projects, 10-day interval	20.9	9.4	8.9

Table. 13: Average number of workers being asked to fill in for an absentee in a busy period

Condition	Simultaneously	Respectively	In combination
Same projects, 5-day interval	74.1	97.6	71.3
Same projects, 10-day interval	76.9	95.1	72.9
Different projects, 5-day interval	86.1	93.8	88.4
Different projects, 10-day interval	89.3	96.2	87.9

and “respectively.” In addition, when the interval of absentee days is 10, the average number of workers in the “simultaneously” method increases considerably. This is because the number of workers managers can simultaneously request to fill in for the two cases decreases. Table13 shows that the “no request” method is effective when there only a few people wish to work. In such a case, external actions such as raising wages are necessary.

6. Conclusion and future studies

In this research, we addressed the problem of contacting alternative workers to fill in for absentees, an issue that has not been tackled so far in the literature. We created a simulation environment to compare various alternative worker selection methods. We then estimated the average number of workers to find a substitute in this environment. The simulation results show that a certain client reduction effect is to be expected when using our method.

Future research will focus on improving the simulation environment to reproduce a real-world situation. We will also conduct a method application test set in the real world and fine-tune the method proposed in this paper based on the result.

7. Acknowledgment

We wish to thank TMJ, Inc. for their support toward this research.

References

- [1] Jorne Van Den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, Vol. 226, No. 3, pp. 367–385, 2013.
- [2] Jeffrey L. Arthur and A. Ravindran. A multiple objective nurse scheduling model. *A IIE Transactions*, Vol. 13, No. 1, pp. 55–60, 1981.
- [3] K A Dowland and J M Thompson. Solving a nurse scheduling problem with knapsacks, networks and tabu search. *Journal of the Operational Research Society*, Vol. 51, No. 7, pp. 825–833, Jul 2000.
- [4] Holmes E. Miller, William P. Pierskalla, and Gustave J. Rath. Nurse scheduling using mathematical programming. *Operations Research*, Vol. 24, No. 5, pp. 857–870, 1976.
- [5] Uwe Aickelin and Kathryn A. Dowland. An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research*, Vol. 31, No. 5, pp. 761 – 778, 2004.
- [6] Brigitte Jaumard, Frédéric Semet, and Tsevi Vovor. A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, Vol. 107, No. 1, pp. 1 – 18, 1998.
- [7] Alistair Clark, Pam Moule, Annie Topping, and Martin Serpell. Rescheduling nursing shifts: scoping the challenge and examining the potential of mathematical model based tools. *Journal of Nursing Management*, Vol. 23, No. 4, pp. 411–420, 2013.
- [8] Michael Mutingi and Charles Mbohwa. Fuzzy Multi-Criteria Simulated Evolution for Nurse Re-rostering. In *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management*, 2016.
- [9] Broos Maenhout and Mario Vanhoucke. Reconstructing nurse schedules: Computational insights in the problem size parameters. *Omega (United Kingdom)*, Vol. 41, No. 5, pp. 903–918, 2013.
- [10] Zdeněk Bäumelt, Jan Dvořák, Přemysl Šůcha, and Zdeněk Hanzálek. A novel approach for nurse rerostering based on a parallel algorithm. *European Journal of Operational Research*, Vol. 251, No. 2, pp. 624–639, 2016.
- [11] Coin-or branch-and-cut mip solver. <https://projects.coin-or.org/Cbc>. (Accessed on 10/10/2018).

Neural Network using Support Vector Machine and Genetic Algorithm

Kyoko Ogawa Naoki Mori Michifumi Yoshioka

‡ Osaka Prefecture University, Osaka, Japan
ogawa@ss.cs.osakafu-u.ac.jp

Abstract: Recently, deep learning has been studied thoroughly and many results have been reported. However, the most effective way to construct neural networks has not yet been determined. Besides, interpretation of an obtained network is difficult. In a previous study, we proposed a novel method to construct a neural network using a support vector machine. In this method, the number of neurons in the neural network and their weights and biases are initialized based on support vector information. This method has an advantage in that we can understand the mechanism of the network based on the support vectors. However, there is also problem in that, for a nonlinear problem, performance is significantly lower than that for a linear problem. In this study, we propose a new network structure, which introduces AND/OR layers. We also utilize the genetic algorithm to pair support vectors to solve the above problem. The effectiveness of the proposed method is confirmed by computer simulations of problems from sklearn.

Keywords: Neural Network, Support Vector Machine, Genetic Algorithm

1. Introduction

Deep neural networks have recently attracted much attention, and many achievements in deep learning have been reported in various fields. Although deep learning is a very powerful tool, it is difficult to understand the meaning of obtained networks. In addition, many researchers study how to construct neural network[1]. To solve this problem, we previously proposed a neural network with a support vector machine[2] (SVM) called “SVM-NN”[3]. In this method, the number of neurons in the neural network and their weights and biases are determined by support vectors in the SVM. Then, the hyper plane of the neural network is expected to behave similarly to that of the SVM prior to training. SVM-NN demonstrated almost the same accuracy rate as the SVM[3]. In several cases, SVM-NN failed to replicate the hyper plane as in the SVM. To solve this problem, we only utilized neurons close to the input data, i.e., those similar to input data, for construction. This solution improved the accuracy, but it was still lower than that of an SVM.

In this study, we aim to improve the performance of SVM-NN so it is at least better than the performance of an SVM. First, we propose a novel method where AND/OR layers are introduced in order to solve nonlinear problems. We also utilize the genetic algorithm (GA) to pair support vectors to improve the performance of SVM-NN. The effectiveness of the proposed method is confirmed by computer simulations of problems from sklearn.

2. Related Work

2.1 Multi Layer Perceptron

Multi Layer Perceptron (MLP)[4] belongs a class of feed forward neural networks. The perceptron algorithm was invented in 1957 by Frank Rosenblatt. In 1969, Minsky and Papert discovered that simple perceptrons were incapable of processing nonlinear classifications such as exclusive-or circuits. Later, Rumelhart et al. proposed a backpropagation algorithm, which can learn weights and biases of multilayer perceptrons. Hence, it is clear that MLP can be used as a nonlinear classifier. The nodes that MLP consists of are called “neurons”. In our method, we use a sigmoid function as an activation function.

2.2 SVM

The SVM [5] is a supervised learning model used for pattern recognition. By using maximization of the margin of two classes as an evaluation function, it determines the optimal hyperplane that separates two classes. In detail, maximization of the minimal distance between training data and the hyperplane determines the hyperplane. When the hyperplane is determined by maximizing the evaluation function, there are generally more than one training data whose distance is minimal. These data are called support vectors because they appear to support the hyperplane. SVMs are categorized as linear or nonlinear. By mapping training data to multidimensional space utilizing a kernel function, a nonlinear SVM can perform nonlinear classification.

2.3 Genetic Algorithm

The genetic algorithm[6] (GA) mimics the natural biological genetic and evolutionary process. It was first introduced by Holland JH in 1975. It is mainly used in computers to solve optimization problems. A typical application of the GA is to search the largest "space" of possible solutions to a problem effectively[7],[8],[9]. The GA uses the generation of evolutionary random individuals to search. Each newly generated individual comes from the previous one using crossover and mutation, and the concept of the imitation selection comes from Darwin's natural selection. The GA randomly generates a candidate solution to form an initial population[10],[11]. In every generation, all individuals are ranked. The individual most able to adapt to the environment is selected as the "father". The future population (or the next "generation") then replaces the current population[12]. Repeating this process, more individuals with the ability to solve the problem are produced[13].

The genes of the new generation are created by genetic operations of crossover and mutation[14]. The specific implementation of this crossover and mutation depends on the solution, which is selected based on the application field.

2.4 SVM-NN[3]

In order to overcome the weight initialization problem of conventional neural networks, we utilize support vectors in SVM. In this section, we discuss the construction of the neural network in detail. First, we train an SVM and obtain support vectors for creating neurons. Then, we set these neurons as the hidden layer of MLP. Support vectors are located to support the hyperplane; therefore, they have important information for defining the hyperplane. In fact, the SVM is established by support vectors. Secondly, we consider a combination of each support vector as a neuron. The procedure to generate neurons from support vectors is as follows:

- (1) Calculate subtraction of each vector
- (2) Normalize the vectors. Let \mathbf{a} be one support vector and let \mathbf{b} be another support vector.

The weight vector \mathbf{w} is calculated as follows:

$$\mathbf{w} = \frac{\mathbf{b} - \mathbf{a}}{|\mathbf{b} - \mathbf{a}|} \quad (1)$$

The vector \mathbf{W} expresses the direction from \mathbf{a} to \mathbf{b} . Calculating the dot product of input data X and \mathbf{w} , the line vertical to vector \mathbf{w} , as a separate plane, can divide two areas. For example, if vector \mathbf{w} is given \mathbf{w} in \mathbb{R}^2 , when $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, the class is \mathbf{b} . When $\frac{\pi}{2} < \theta < \pi$, $-\pi < \theta < -\frac{\pi}{2}$, the class is \mathbf{a} . Similarly, the input data X in Fig. 1 is classified into \mathbf{b} . In the end, in order to adjust the middle point of each vector, the bias is calculated with middle point vector \mathbf{m} as follows:

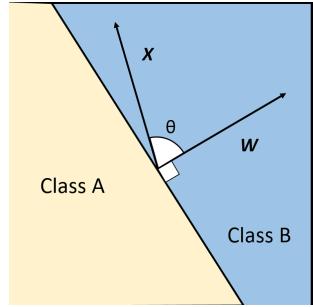


Fig. 1: Recognition by vectors

$$\mathbf{w}_o = -\mathbf{w}^T \cdot \mathbf{m} \quad (2)$$

As a result of this adjustment, when the angle of the weight vector and input data is $\frac{\pi}{2}$ or $-\frac{\pi}{2}$, the neuron's output is zero. Therefore, the result is able to be explain the class by a plus or minus. In this way, many neurons can be created. The output of the hidden layer neurons is calculated as sum of all inputs. Generally, in the classification application, the number of output layer neurons is determined as the number of classes. Additionally, the maximum output neuron corresponds to the result of classification. In our method, the number of output layer neurons is only two because we address 2-class classification for simplification. The weights of output layer neurons are either all +1 or all -1. In the -1 case, in order to maximize the output, all elements of matrix \mathbf{w} must be -1. In the +1 case, all elements must be +1. In order to select a combination of support vectors, the pairs of support vectors are determined by euclidean distance. Calculating euclidean distance from a support vector in class \mathbf{a} , it is paired with the nearest support vector in class \mathbf{b} . In this way, the number of neurons in the hidden layer is determined by the number of support vectors. This prevents combinatorial explosion caused by the number of support vectors.

3. Proposed Method

3.1 SVM-NN with an AND and OR layers

There was a problem in that SVM-NN cannot be applied to nonlinear problems. In order to solve this problem, we have tried to introduce AND and OR layers to a neural network to identify the nonlinear data. In the feature space, we suppose that there are valid support vectors for each input data, and we group them using a training data set. Using this procedure, it is possible to limit the support vectors used for identifying input test data close to certain training data and eliminate influences from unnecessary support vectors. For this purpose, for each learning datum, a structure for collecting support vectors showing correct results and outputting logical products of these is introduced as an AND layer.

For example, suppose that the result is different for certain data d due to the influence of misidentifying a support vector of a specific area. At this time, by introducing the AND layer, only the support vector group in which all of the data d is positive is activated, so the above problem can be solved. In the OR layer, data activates when one or more groups of the AND layer are activated. We call this type of SVM-NN "SVM-NN(AND/OR)".

3.2 Support Vector Pairing by the GA

The basic concept of SVM-NN is to make the hyper plane a vertical bisector of two selected support vectors. If the problem is linear, there is no problem in pairing the closest 2 support vectors. However, if the problem is nonlinear, SVM-NN cannot solve the problem with this type of pairwise closest method. To solve this problem, we introduce the GA to find the best pair set for SVM-NN.

The representation of GA genotype as follows:

Set the number of support vector(SV) of class 0 side to N_0^{SV} , and that of class 1 side to N_1^{SV} . First, we denote the index number to all SVs of class 0 and class 1 respectively. For each SV in class 0 or 1, one target SV of the different class is decided. Genotype is represented as arranging target SV number. If the two lines class 0 to class 1 and class 1 to class 0 are the same, only one line is selected. Otherwise, there are two lines class 0 to class 1 and class 1 to class 0. The chromosome length is $N_0^{\text{SV}} + N_1^{\text{SV}}$ in this setting.

Fig. 2 represents the example of chromosome. Left 6 alleles show the SVs of class 0, while right 6 alleles show the SVs of class 1. The leftmost allele shows the "one line is drawn from SV of index 0 of class 0 to SV of index 2 of class 1."

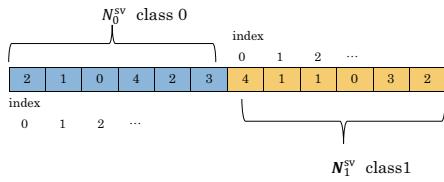


Fig. 2: Example of genotype

4. Experiments

In this study, the two problems "Moons" and "Gaussian quantiles" from *sklearn.datasets* are considered. Moons is for binary classification and will generate a swirl pattern or two crescent moons. We can control how noisy the moon shapes are and the number of samples generated. We set the noise value to 0.06 and the number of samples to 600, and the other parameters were given their default values. Gaussian quantiles is constructed by taking a multi-dimensional standard

normal distribution and defining classes separated by nested concentric multi-dimensional spheres such that all classes have approximately the same number of samples. We set the number of classes to 3 and the number of samples to 600, and the other parameters were given their default values. Fig. 3 shows all data from Moons. Fig. 4 shows all data from Gaussian quantiles. In both problems, the input vector is 2-dimensional and each axis represent input vector x and y .

In Moons, 20% of data ($600 \times 0.2 = 120$) was test data, and 480 data were set as training data. In Gaussian quantiles, there is 3-class classification, so we selected 2 classes from the dataset in order to exchange 2-class classification and used only 400 data for experiment. We tried all ${}_3C_2$ patterns of 2-class classifications. As with Moons, 20% of data ($400 \times 0.2 = 80$) was test data, and 320 data were training data.

These tests are suitable for algorithms that can learn non-linear class boundaries. We first applied SVM to these data

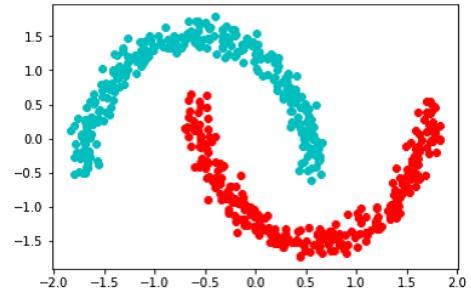


Fig. 3: The sample moons data set

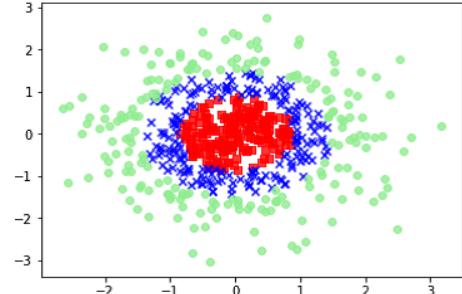


Fig. 4: The sample Gaussian data set

to obtain support vectors, and then we initialized SVM-NN and SVM-NN(AND/OR).

4.1 SVM

Several parameters were set by the grid search as shown in the following table. Other parameters used the default values set by *sklearn*. Table 1 shows the SVM parameters in Moons, and Table 2 shows the SVM parameters in Gaussian quantiles.

In this experiment, a multi-class SVM (one versus one) was applied.

Table. 1: SVM Parameters in Moons

parameters	value
kernel	RBF
C	1
gamma	1

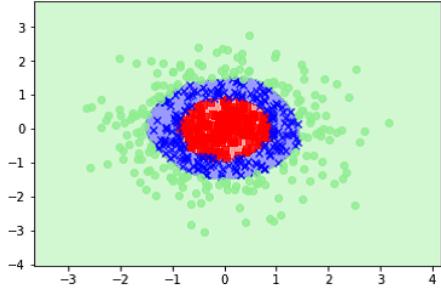


Table. 2: SVM Parameters in Gaussian quantiles

parameters	value
kernel	RBF
C	1000
gamma	1

The accuracy rate of Moons in test data was 100% and the accuracy rate of Gaussian quantiles in test data was 99%. Fig. 5 shows the SVM results of Moons. Fig. 6 shows the SVM results of Gaussian quantiles.

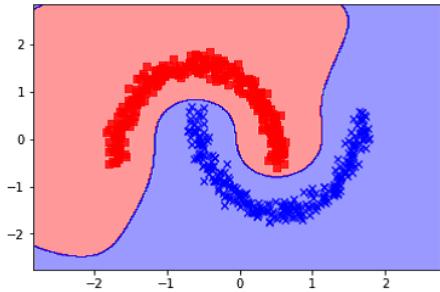


Fig. 5: SVM result in Moons

4.2 SVM-NN

Here, we show the results of the original SVM-NN[3]. The accuracy rate of Moons was 88.4% at the initial state. The accuracy rates of Gaussian quantiles were as follows:

between label 0 and label 1 50.0%

between label 1 and label 2 58.0%

between label 2 and label 0 50.0%

The baseline of this problem is 50.0%, so the above results are almost meaningless. Fig. 7 shows the SVM-NN results of Moons. From Fig. 7, it can be seen that SVM-NN cannot recognize the end of the distribution data. Figs. 8, 9, and 10 show the SVM-NN results of Gaussian quantiles between two labels (0,1), (1,2), and (2,0), respectively. In the case of Gaussian quantiles, it can be seen that the SVM-NN cannot distinguish the data at all. In fact, the outputs of SVM-NN are only one class. These results show that we cannot apply SVM-NN to nonlinear problems.

Fig. 6: SVM results in Gaussian quantiles

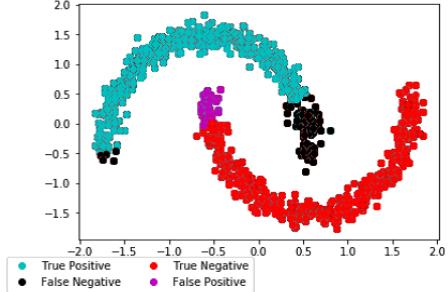


Fig. 7: SVM-NN results in Moons

4.3 SVM-NN(AND/OR)

Here, we show the results of the original SVM-NN(AND/OR). The accuracy rate for Moons was 96.7% at the initial state.

The accuracy rates of Gaussian quantiles were as follows:

between label 0 and label 1 60.0%

between label 1 and label 2 50.1%

between label 2 and label 0 93.0%

The accuracy rate between label 2 and label 0 is very high. However, accuracy rate between label 1 and label 2 is almost

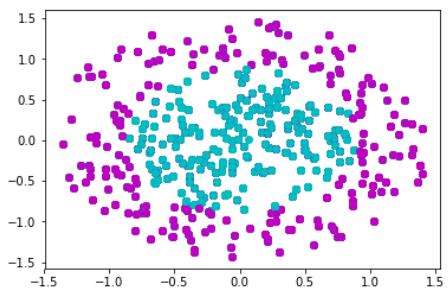


Fig. 8: SVM-NN results in Gaussian quantiles (label 0 vs label 1)

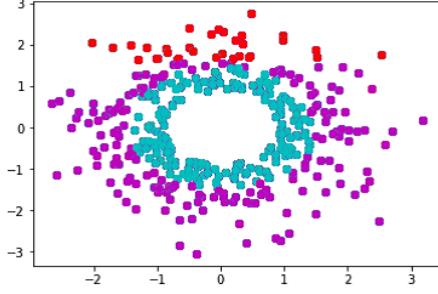


Fig. 9: SVM-NN results in Gaussian quantiles (label 1 vs label 2)

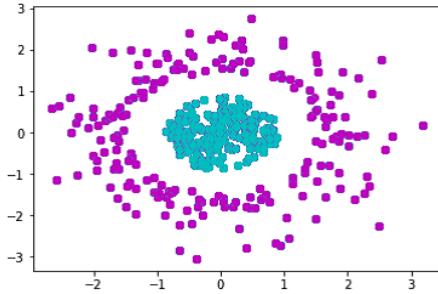


Fig. 10: SVM-NN results in Gaussian quantiles (label 0 vs label 2)

baseline. These results show that the performance of SVM-NN(AND/OR) is higher than that of SVM-NN.

Fig. 11 shows the SVM-NN results of Moons. From Fig. 11, it can be seen that SVM-NN cannot recognize the end of the distribution data. Figs. 12, 13, and 14 show the SVM-NN(AND/OR) results of Gaussian quantiles between two labels (0,1), (1,2), and (2,0), respectively. In the case of Gaussian quantiles, it can be seen that the SVM-NN(AND/OR) can distinguish some data, i.e., label 0 vs label 2. However, in case of label 1 vs label 2, SVM-NN(AND/OR) did not perform well. This is because when both classes data were close, the AND layer of the support vectors does not work well. It is also true for label 0 vs label 1.

SVM-NN and SVM-NN(AND/OR) have the same support vectors pair. Therefore, we show the result of support vectors pair of SVM-NN and SVM-NN(AND/OR).

Fig. 15 shows the support vector pairs of Moons. The pairs of support vectors are reasonable except at the end of the data. In Fig. 15, the problem may be that the numbers between two classes are unbalanced.

Figs. 16, 17, and 18 show the support vector pairs of Gaussian quantiles between two labels (0,1), (1,2), and (2,0), respectively.

Figs. 16 and 17 show that support vector pairs is parallel to the boundary. This is because the accuracy rates between

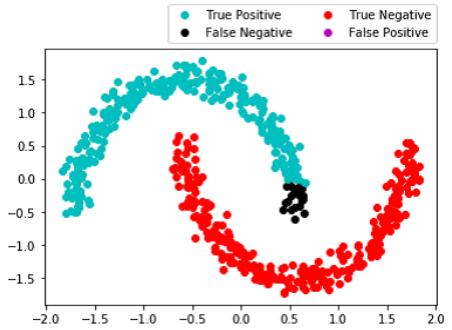


Fig. 11: SVM-NN(AND/OR) results in Moons

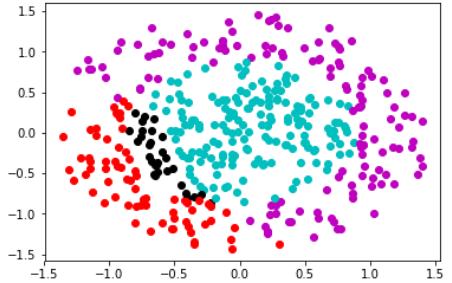


Fig. 12: SVM-NN(AND/OR) results in Gaussian quantiles (label 0 vs label 1)

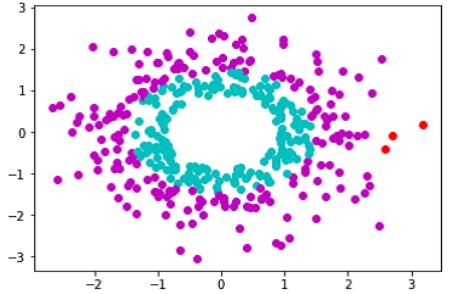


Fig. 13: SVM-NN(AND/OR) results in Gaussian quantiles (label 1 vs label 2)

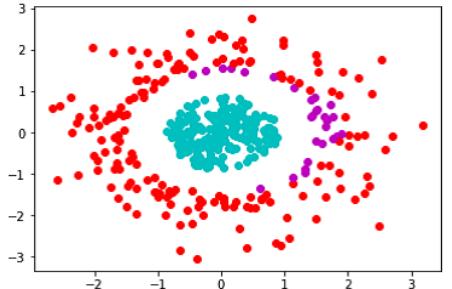


Fig. 14: SVM-NN(AND/OR) results in Gaussian quantiles (label 0 vs label 2)

label 0 and label 1 and between label 1 and label 2 are very low. On the other hand, the pairs of support vectors in Fig. 18 are reasonable.

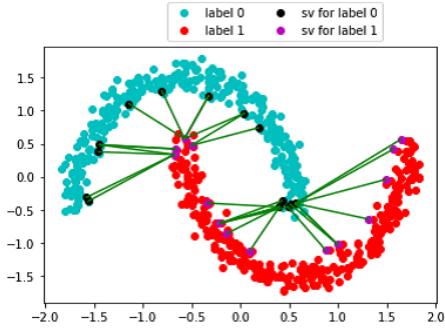


Fig. 15: Support vector pair of Moon

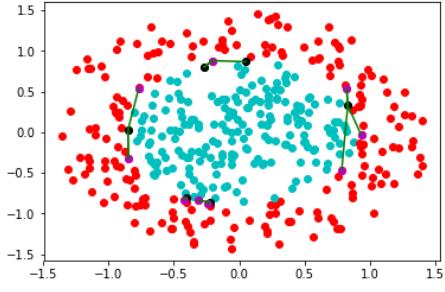


Fig. 16: Support vector pair of Gaussian quantiles (label 0 vs label 1)

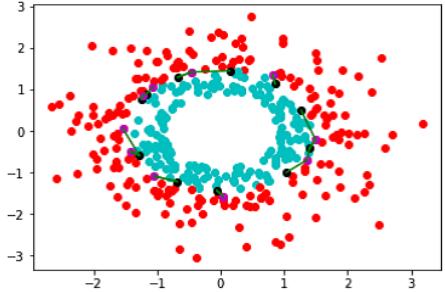


Fig. 17: Support vector pair of Gaussian quantiles (label 1 vs label 2)

Next, we compare the SVM-NN(AND/OR) and a normal neural network (NN). The size of the normal NN is same as that of SVM-NN(AND/OR), but a ReLU function is utilized instead of a sigmoid as the activation function.

Fig. 19 shows the variation of test accuracy rate of Moons. The vertical axis is epoch, and the horizontal axis is accuracy rate. From Fig. 19, it can be seen that the learning speed of SVM-NN(AND/OR) is faster than that of a normal NN. We

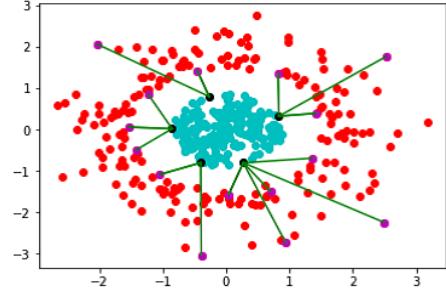


Fig. 18: Support vector pair of Gaussian quantiles (label 0 vs label 2)

know that SVM-NN(AND/OR) uses the SVM before training, so the result is not fair. However, the high accuracy rate of SVM-NN(AND/OR) in the early stages of training is useful.

Figs. 20, 21, and 22 show the variation of accuracy rate of Gaussian quantiles. These figures show the same results as Moons. In the case of label 0 vs label 2, the learning speed of SVM-NN (AND / OR) is faster than that of normal NN. An important point to make from these results is that SVM-NN(AND/OR) can solve Gaussian quantiles after training, while previous results were obtained before training. However, in case of label 1 vs label 2, the performance of SVM-NN(AND/OR) is low. This is because when both classes data were close in the feature space, the support vectors for the AND layer are hard to distinguish those two classes effectively. Similar trends are observed in label 0 vs label 1.

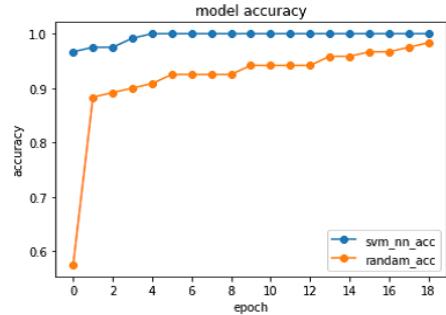


Fig. 19: The variation of accuracy rate in Moons

4.4 SVM-NN(AND/OR) with GA

Table 3 shows the setting of GA.

As a result of selecting a pair of support vectors for the Moons data with GA, the accuracy rate was 98.7% at the initial state. This is the best value among SVM-NN and SVM-NN(AND/OR) at the initial state. Fig. 23 shows the selected pair of support vectors. Fig. 24 shows the variation of accuracy rate. The vertical axis and horizontal axis are the same as in Fig. 19. From the viewpoint of training, the

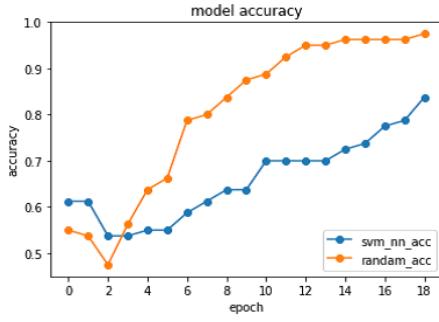


Fig. 20: The variation of accuracy rate in Gaussian quantiles (label 0 vs label 1)

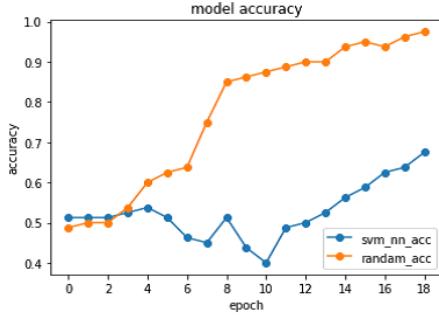


Fig. 21: The variation of accuracy rate in Gaussian quantiles (label 1 vs label 2)

training speed of SVM-NN(AND/OR) with GA is slower than that of SVM-NN(AND/OR). However, the test accuracy of SVM-NN(AND/OR) with GA reaches 1.0. These results show that SVM-NN(AND/OR) with GA is useful and shows the same ability as SVM-NN(AND/OR).

4.5 Search Dynamics of GA

Statistical analysis has been carried out to show the search dynamics of GA.

Table 4 shows the results of the statistics of fitness (accuracy) with various random seeds. Although the three trials are not enough to analyze the GA sensitivity of parameters, because of computer resources, we set the trial number to three.

Table 5 shows the results of statistics of the accuracy of the random search.

From tables 4 and 5, the performance of GA is robust and excellent value to find the pairs of support vectors.

Fig. 25 shows the variation of max fitness value with generation. The horizontal axis shows the number of generation, and the vertical axis shows the average of max fitness value in 3 trials.

From Fig. 25, the search dynamics of GA is well to solve the problem.

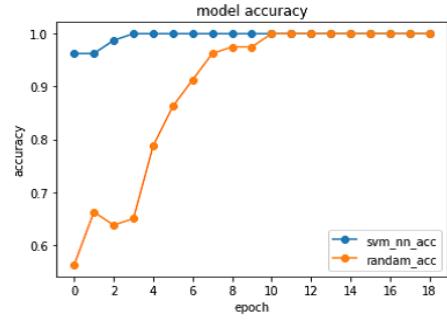


Fig. 22: The variation of accuracy rate in Gaussian quantiles (label 0 vs label 2)

Table. 3: Setting of GA

generation size	20
population size	100
chromosome length	28
crossover type	two-point crossover
crossover rate	0.5
mutation rate for each individual	0.2
mutation rate for each locus	0.05
selection	tournament selection
tournament size	3

5. Conclusion and future works

In this study, we proposed two methods for improving SVM-NN performance. First, we introduced AND and OR layers into SVM-NN in order to adapt it to nonlinear problems. Next, we attempted to pair support vectors using the GA. From the results of computer experiments, we have shown that the performance of the proposed SVM-NN is superior to that of SVM even for nonlinear problems. Important future works are as follows:

- Applying SVM-NN to practical problems.
- Comparison study between SVM-NN(AND/OR) with GA and SVM-NN.
- Utilizing the GA to not only make support vector pairs but to determine network hyperparameters.

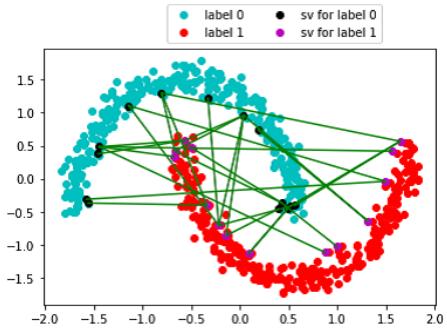


Fig. 23: Support vector pair of Moon with GA

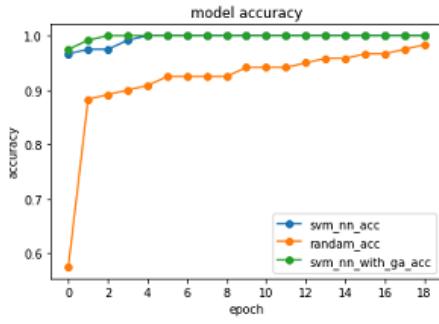


Fig. 24: The variation of accuracy rate in Moons with GA

Table. 4: Statistics of fitness values

mean	0.9867
variance	0.1111×10^{-4}
standard deviation	0.3333×10^{-2}

- Gathering the support vectors in SVM with a difference kernel function.
- Combining the proposed method with other types of machine learning such as the random forest.
- Improve the performance of the proposed method in cases that both classes data are close in the feature space.

Acknowledgement

A part of this work was supported by a JSPS KAKENHI Grant, Grant-in-Aid for Scientific Research(C), 26330282. A part of this work was supported by a research grant from the Murata Science Foundation.

References

- [1] Eric B Baum and David Haussler. What size net gives valid generalization? In *Advances in neural information processing systems*, pages 81–90, 1989.
- [2] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables.
- [3] Yuto Mukai, Michifumi Yoshioka, and Katsufumi Inoue. Optimal initialization of neural network weights by support vectors. *Proc. of 6th Asian Conference on Information Systems*, pages 153–156, 2017.
- [4] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. *IJI-MAI*, 4(1):26–30, 2016.
- [5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [6] David E Goldberg. *Genetic algorithms*. Pearson Education India, 2006.
- [7] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 4(03):66, 2012.
- [8] Ali Niazi and Riccardo Leardi. Genetic algorithms in chemometrics. *Journal of Chemometrics*, 26(6):345–351, 2012.
- [9] Hassan Khajemohammadi, Ali Fanian, and T Aaron Gulliver. Efficient workflow scheduling for grid computing using a leveled multi-objective genetic algorithm. *Journal of grid computing*, 12(4):637–663, 2014.
- [10] Daniel Álvarez, Roberto Hornero, J Víctor Marcos, and Félix del Campo. Feature selection from nocturnal oximetry using genetic algorithms to assist in obstructive sleep apnoea diagnosis. *Medical engineering & physics*, 34(8):1049–1057, 2012.
- [11] H Tian, C Liu, XD Gao, and WB Yao. Optimization of auto-induction medium for g-csf production by escherichia coli using artificial neural networks coupled with genetic algorithm. *World Journal of Microbiology and Biotechnology*, 29(3):505–513, 2013.
- [12] Farhana Naznin, Ruhul Sarker, and Daryl Essam. Vertical decomposition with genetic algorithm for multiple sequence alignment. *BMC bioinformatics*, 12(1):353, 2011.
- [13] Amorn Wongsarnpigoon and Warren M Grill. Energy-efficient waveform shapes for neural stimulation revealed with a genetic algorithm. *Journal of neural engineering*, 7(4):046009, 2010.
- [14] Nazar Zaki, Salah Bouktif, and Sanja Lazarova-Molnar. A combination of compositional index and genetic algorithm for predicting transmembrane helical segments. *PloS one*, 6(7):e21821, 2011.

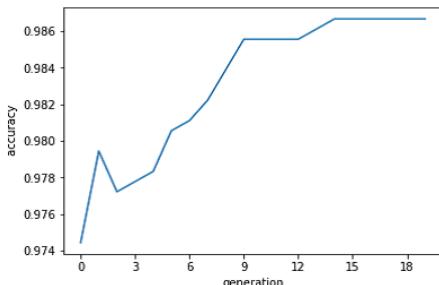


Fig. 25: Max fitness with generations (average of 3 trials)

Table. 5: Technical statistics for random cases

mean	0.8267
variance	0.1479×10^{-1}
standard deviation	0.1216

Journal of machine learning research, 6(Sep):1453–1484, 2005.

Memory Augmented Neural Networks with Paired Controllers for One-shot Learning

Trung Kien Tran [†] Hiroshi Sato [†] Masao Kubo [†]

[†] National Defense of Academy, Yokosuka, Kanagawa, Japan
`{ed17005, hsato, masaok}@nda.ac.jp`

Abstract: One-shot learning is always one of the most difficult challenge in machine learning. However, there are also some novel researches for this task and achieve the very well results. One of them is the Memory-Augmented Neural Network, which is an application of Neural Turing Machine in field of One-shot/Few-shot learning task. In this paper, an improvement of this model is proposed. Unlike the original model that use only one neural network as a controller, we use two controller networks to control the inputs as well as the outputs. This alteration makes the model handle better the output of memory lead to the higher accuracies. Particularly, in doing 5-way one-shot tasks on Omniglot Dataset, this model could get average 91.07% accuracy at the second instance and 96.73% accuracy at the fifth instance compared to 82.8% and 94.9% of the original model respectively.

Keywords: one shot learning, Neural Turing Machine, Memory Augmented Neural Network, Least Recently Used Access, Paired Controller Memory Augmented Neural Network

1. Introduction

Machine learning has been successfully used in a variety of applications today. Its community as seen enormous improvements on object recognition and classification that make the application almost limitless. However, these algorithms are often broken down if they had to deal with the problems which information/data is insufficient. Such situations are very common in real life such as quickly detect a new kind of disease with only a few symptoms recognized from the early state, etc. For those purposes, specialists define the one-shot/few-shot learning algorithms that could solve a task with only one or a few known samples. Despite these one-shot learning challenges are evaluated the most difficult tasks, the current researches still bring the very well results. These works are mostly developed in computer vision first, and then adapted to the other specific fields of society.

In terms of one-shot learning, many methods have been proposed to deal with the problems of a lack of data. Some concepts base on Bayesian approach such as the works of Li Fei-Fei et al. in 2006 about leaning object categories with One-shot learning [1], or probabilistic program induction by Brenden M Lake et al. [2], or R. Salakhutdinov et al. with the work of Hierarchical Non parametric Bayesian Model [3], while the others use meta-learning to teach the network how to learn itself. Thanks to the huge development of deep learning approaches recently, some researchers make use of Deep Neural Networks in doing this kind of task even it requires lots of samples to run through lots of iterations of stochastic gradient descent and gradually refine the weights

of the network. For example, there are Memory Augmented Neural Network introduced by A. Santoro [4], Matching Network by O. Vinyals et al. [5], and Siamese Network [6] by G. Koch. Generally, due to a requirement of large data for training with neural network, most of these approaches use a method of transfer knowledge from pre-trained model with known dataset to recognize new samples of new kind of categories.

Among the mentioned researches above, we focus on Memory Augmented Neural Network, which is a promising model. It is one of the meta-learning methods using an external memory associated with a Recurrent Neural Network (RNN) or Long Short Term Memory networks (LSTM). It is based on Neural Turing Machine [7], and adapted to one-shot learning task. However, in this model, there are still some minor points need to be improved.

In this paper, we propose an advanced modification of Memory Augmented Neural Network so that it could do better the one-shot learning task. For performance evaluation, we implement our model with Omniglot dataset [2], which is a specific dataset for one-shot/few-shot learning task of handwriting character recognition, and then compare the results to the original one.

2. Related works

2.1 Neural Turing Machine

Introduced by Grave et al. [7] from Google Deepmind in 2014, Neural Turing Machine (or NTM) is the at-the-edge-of-research approach that resembles the brain's working

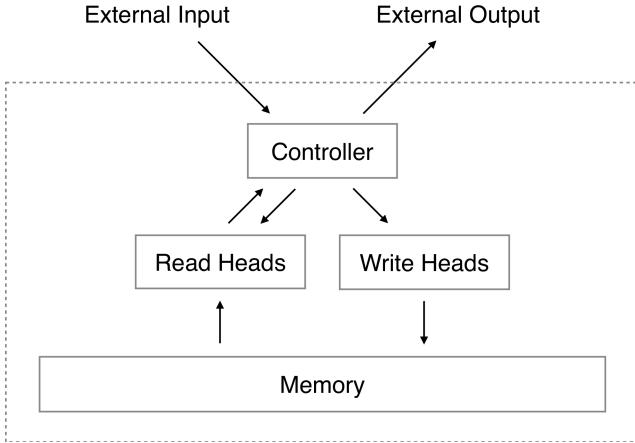


Fig. 1: Neural Turing Machine Architecture. In practice, a controller is usually RNN (LSTM) networks, and could have several read and write heads. Excerpt from [7]

memory that consists of a neural network as a controller, control read/write operation of input's content to an external memory module via read/write heads. The model architecture is somehow similar to the computer system, which content external memory represents a computer's Read Only Memory (RAM) module, the read/write heads in place of read/write buses, and the controller is functioning as CPU. There are also some differences between them such as unlike CPU which fed the program instructions, Controller's Neural Network learns its inputs. Another difference is that the whole architecture of NTM is differentiable to compute the gradient of some output loss with respect to the model's parameters, while digital computer accesses RAM via discrete memory addresses. The way of accessing the memory of NTM is also noticeable. NTM uses Attention-based Memory Access that each head generates a normalized softmax attention vector to access all the location in the memory at the same time to select the most appropriate location depends on the memory addressing mechanisms. According to the authors, 2 addressing mechanisms are introduced, content-based addressing and location-based mechanism.

Thanks to the novel architecture, NTM could do a variety of tasks in practice such as copying, sorting, as well as associative recall from input and output. Although NTM is still suffering from a couple of issues, it could represent a promising direction for future research.

2.2 Memory Augmented Neural Network with Least Recently Used Access mechanism

Regarding one-shot learning tasks, some meta-learning approaches are proposed. One of them uses Recurrent models to teach a model how to learn itself over tasks. S. Hochreiter et al. [8] proved that meta-learning system could be a self-learning system by including a fixed supervisory procedure

and a adjustable subordinate system, which must be run on certain medium. Taking advantage of memory networks such as NTM [7] and Memory Networks [9], which provides an up-and-coming approach for meta-learning in deep learning networks, A.Santoro et al. [4] from Google DeepMind, has demonstrated the capability of meta learning in one-shot learning tasks in term of short and long term memory demands. Their model "combines the best of two worlds: the ability to slowly learn an abstract method for obtaining useful representations of raw data, via gradient descent, and the ability to rapidly bind never-before-seen information after a single presentation, via an external memory module." [4]. For one-shot learning tasks, the LRUA MANN model is considered as a recurrent models approach for meta-learning, but consistently outperforms the performance of LSTM since it overcomes the limitations of LSTM network. This model uses fixed size external memory matrix which is stable and not directly related to size of a network, and has a specific memory addressing mechanisms, which allow accessing memory location dynamically.

To archive that, they use newly designed memory writing module called the Least Recently Used Access(LRUA) addressing mechanism. This mechanism writes information into either the least used memory location (rarely-used locations) or the most recently used memory location (last used location - update memory with newer information) of the external memory. It calculates the write-weight vector w_t^w as following:

$$w_t^w \leftarrow \sigma(\alpha)w_{t-1}^r + (1 - \sigma(\alpha))w_{t-1}^{lu} \quad (1)$$

where, $\sigma(\alpha)$ is a sigmoid function of a scalar parameter α , w_{t-1}^r is the read-weight vector of a previous step, and w_{t-1}^{lu} is the least used weight vector, generated from the usage weight vector w_{t-1}^u that update every steps with a decay parameter γ as

$$w_t^u \rightarrow \gamma w_{t-1}^u + w_t^u + w_t^w \quad (2)$$

Using this vector, an important weight called least-used weight w_t^{lu} is defined accordingly:

$$w_t^{lu}(i) = \begin{cases} 0 & \text{if } w_t^u(i) > m(w_t^u, n) \\ 1 & \text{if } w_t^u(i) \leq m(w_t^u, n) \end{cases} \quad (3)$$

Memory will be written in accordance with this write-weight vector

$$M_t(i) \leftarrow M_{t-1}(i) + w_t^w(i)k_t \quad (4)$$

For reading out of memory, first, it computes a cosine distance between query key vector generated from controller output k_t and all the memory cells as

$$K(k_t, M_t(i)) = \frac{k_t \cdot M_t(i)}{\|k_t\| \|M_t(i)\|} \quad (5)$$

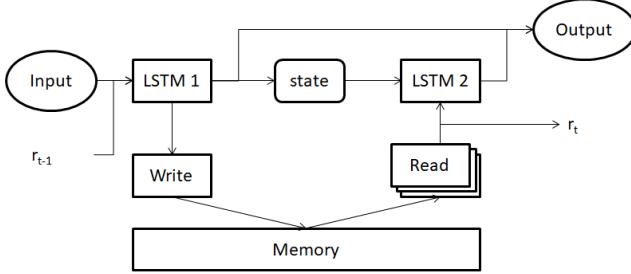


Fig. 2: Memory Augmented Neural Network with Paired Controllers' structure

Then, this measures is used to create a read-weight vector $w_t^r(i)$ which is a result of its softmax function.

$$w_t^r = \text{softmax}(K) \quad (6)$$

Finally, read vector r_t is generated

$$r_t \leftarrow \sum_i w_t^r(i) M_t(i) \quad (7)$$

This read vector will be used in conjunction with the hidden state of the controller to produce the output of the network.

In general, NTM as well as LRUA MANN are perfect for meta-learning and one-shot/few-shot learning tasks as they could look back to the learned materials by both long-term memory via network's updated weights and short-term memory of their external memory.

3. Memory Augmented Neural Network with Paired Controllers

Inspired by the recently success of the application of Neural Turing Machine in on-shot learning task, LRUA-MANN, our approach, Memory Augmented Neural Networks with Paired Controllers (MANN-PC) adds one more extra controller network to the original MANN to handle the output of it. A recurrent neural network's unit as an output controller is added to the outbound of the memory in conjunction with the already existing one to encode the state of the reading vectors of the memory. Then, it is combined with the output of the input controller for the output of the network. Also, this modification turns the input controller and output controller into interdependent units as these controllers exchange their states. This modification works well since it could help the model controlling the output of network to increase the overall prediction's accuracies. By doing that, the two controllers are in our network: one is for controlling input, and the other is for the output. The structure of our model is depicted in the figure 2.

Beside, Long Short Term Memory - LSTM cells are used in the place of controller network. Although two types of controller networks, namely LSTM network and

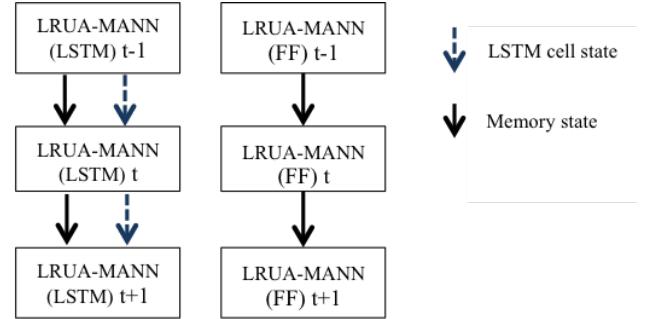


Fig. 3: Two types of LRUA-MANN: LSTM controller (left) and Feed Forward controller (right)

FeedForward neural network, could be used in the MANN model as well as our model, LSTM network are chosen because the models could make use of its internal memory to memorize data. The comparison of these controller networks is pictured in the figure 3.

The input controller process its input, a combination of x_t and y_{t-1} as well as the previous time-step read-out vector of the memory r_{t-1} and an output controller's state ($h2_{t-1}$, $c2_{t-1}$), and generates controller vector according to:

$$h1_t = \text{LSTM}([x_t, y_{t-1}], r_{t-1}, h2_{t-1}, c2_{t-1}) \quad (8)$$

The output controller reads r_t from the memory, which is produced by eq.7 and prepare for the next time step as a part of the input of the input controller, associates with the state of the input controller $h1_t$, $c1_t$ to generate a output of the output controller ($h2_t$, $c2_t$)

$$h2_t = \text{LSTM}(h1_t, r_t, h1_{t-1}, c1_{t-1}) \quad (9)$$

The output of MANN-PC is a categorical distribution p_t of a combination of the input controller's and the output controller's output.

$$o_t = (h1_t, h2_t) \quad (10)$$

$$p_t = \frac{\exp(W^{op}(i)o_t)}{\sum_j \exp(W^{op}(j)o_t)} \quad (11)$$

where W^{op} are the weights from the controller output to the linear layer output.

4. One-shot Learning Classification

4.1 Omniglot Dataset

In this experiment, we test introduced model using Omniglot dataset conducted by Lake et al.[2] This dataset is a collection of 50 different alphabets divided into 1623 different classes. Each class represents a character drawn online by 20 different people via Amazon's Mechanical Turk. For example, some Japanese writing system's Hiragana characters extracted from



Fig. 4: Three example character classes of Omniglot dataset: letter a, i and u in the Hiragana, a syllabary of the Japanese writing system. Each letter is written in 20 different handwriting styles.

this dataset are illustrated in the figure 4. Like the other characters in this dataset, these 105x105 pixel black&white Hiragana character images contain only letter's stroke data with no background noise, make themselves as simple as possible for learning; hence, This Omniglot dataset is an ideal benchmark data for one-shot learning classification tasks.

4.2 Parameters

In our experiments, we keep all the hyper parameters of the network used in the original experiment with MANN executed by A. Santoro et al. except the optimize method. Particularly, instead of using Rmsprop as a method for optimization with indicated parameters, we use ADAM optimizer that makes our experiments better. The other parameters are the followings: 128 memory slots of size 40, 200 hidden units for both LSTM controllers' size , 4 read heads, usage decay of write weights 0.99, size of minibatch of 16.

4.3 Omniglot Classification

Since we accomplish the meta-learning task for self-learning system, we follow a setup suggested by Hochreiter et al. [8] In this task, each episode involves the presentation of a sequence of random inputs $S = \{s_t\}_{t=1}^T = \{(x_t, y_{t-1})\}_{t=1}^T$, where we combine image samples with the previous label, or classification target y_{t-1} . For each time step, the system will predict the appropriate label y_t for a combination (x_t, y_{t-1}) . Hence, we feed the network the following sequence $(x_1, \text{null}), (x_2, y_1), (x_3, y_2), \dots (x_n, y_{n-1})$, and the system must return $y_1, y_2, y_3, \dots, y_n$ respectively.

Using Omniglot dataset, the training phase is conducted with 1200 classes, and the rest 423 classes are used for testing phase. Since there are only 20 images per class, the model might understand the its data too well lead to the problem of overfitting. To avoid it, each input is randomly rotated by multiples of 90 degrees as well as shifting the image as data augmentation.

Moreover, in this experiment, we implement 5-way 1-shot learning task, which is a classification of 5 categories using only one or more training sample(s). At each episode of training process, we randomly select 5 classes among 1200

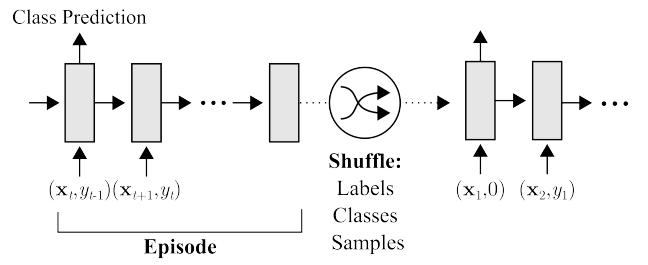


Fig. 5: One-shot learning task setup. Excerpt from[4]

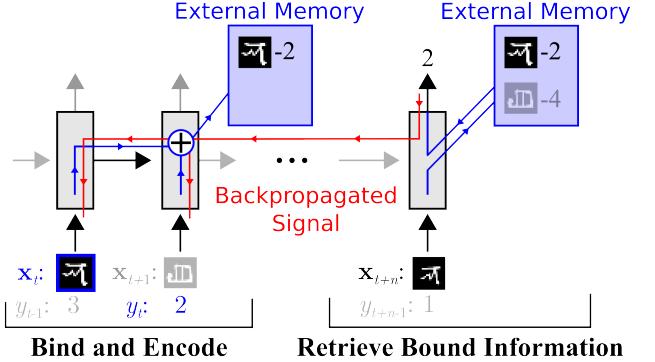


Fig. 6: One-shot learning task's strategy. Excerpt from[4]

classes in the training phase and 423 classes of the testing phase, and label them as a one-hot vector number from 1 to 5. Also, to avoid a situation that the system will slowly learn binding-class in its weight, labeled classes are also shuffled randomly from episode-to-episode. This process is demonstrated in the figure 5.

The external memory plays an importance role in this task. It is used to store the representation of a combination of image and class label for later use, which is considered as a reference point when an image from the known class is presented. The figure 6 demonstrated the strategy of bounding image in and retrieving information out from external memory. For a given episode, the first representation of a class is randomly guessed, and is stored temporarily in memory until the correct label is presented in the subsequent episode. This appropriate combination of image-label is then used for correct prediction consequently.

For the experiment procedures, firstly, the network will be trained with 1200 specific classes using one-hot vector representation for 5 random chosen classes each episode. This training process is terminated after 100000 episodes, and turned to the testing phase. As pictured in the figure 7, after a few hundred episodes, from the second instance of input image series, the model could learn pretty well and understand image class over 80%. Even though, the first instance accuracy (37.42%) could also above chance (random guessing of 1 of 5 possible chance will be around 20%)

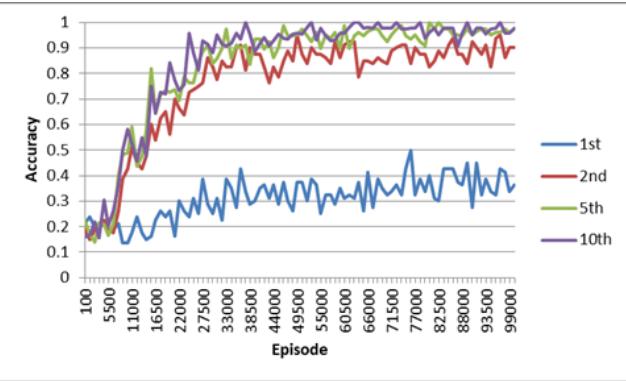


Fig. 7: Training process with five random classes/episode using one-hot vector labels. Except the first instance, from the second instance of samples, the accuracies are all above 90% after 100.000 training episodes.

Table. 1: Comparison of our results with other methods

Model	1 st	2 nd	3 rd	4 th	5 th	10 th
MANN-PC	37.42±0.36	91.07±0.47	94.68±0.15	95.98±0.2	96.73±0.2	97.84±0.13
MANN [4]	36.4	82.8	91.0	92.6	94.9	98.1
LSTM [4]	24.4	49.5	55.3	61.0	63.6	62.5
FeedForward [4]	24.4	19.6	21.1	19.9	22.8	19.5

indicating that like the original MANN model, our model is also performing guessing task very well based on what it has learned in the past.

In testing phase, the model will predict 5 random classes out of 423 classes of the input samples, and no further learning for the other disjointed 423 never-seen-before classes is conducted. The testing is attempted 5000 times, and the final result is the average accuracies. According to the result listed in the table 1, the network does the one-shot classification job quite well when even from the second presentation of the sample, an accuracy could reach up to 91.07%, and 96.73% at the fifth instance compared to the result of MANN 82.8% and 94.9% respectively. Although the tenth instance of our model is not high as the MANN, the gap between them is not considerable (97.84% compare to 98.1%). Regarding the baselines, the author of MANN also implemented these experiments with LSTM network and vanilla FeedForward neural network [4]. However, due to the limit of memory capacity, these baselines are outperformed by the LRU MANN and our model. For example, at the second instance, the LSTM networks could correctly guess about 49.5% accuracy, and the FeedForward network is only 19.6%. Moreover, the results also show that the prediction of all the test samples are close to the average value since the calculated standard deviation is very low indicates that the prediction is able to trust.

5. Discussion

In this paper, we introduced a modification of the original model Memory-Augmented Neural Networks, by way of enhancing the training process as well as the performance on predicting the one-shot learning task. Particularly, by adding one more LSTM controller network to the output of the network, the MANN-PC model could take advantage of both long-term memory of network's weights and short-term memory of external memory better than using only one for the controller.

Although this MANN-PC could outperform the LRU MANN in some aspects of one-shot learning task, to fully evaluate the performance, other experiments still need to be conducted such as 5-way one-shot learning task using five character-long strings as labels, regression task, etc.

Nevertheless, despite this approach could improve the MANN's performance, compare to the other methods such as Matching Network, the result is not as good as expected. Hence, the future works will focus on renovating this model, such as enhance the controllers, writing and reading mechanisms, etc., and make the use of Neural Turing Machine in one-shot learning tasks much more efficient.

References

- [1] Li Fei-Fei et al., "One-Shot Learning of Object Categories", IEEE Transactions on pattern analysis and machine intelligence, vol.28, No.4, April 2006, pp594-611, 2006
- [2] Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. Science, 350(6266), 1332-1338., 2015
- [3] Ruslan Salakhutdinov et al., "One-Shot Learning with a Hierarchical Nonparametric Bayesian Model", JMLR: Workshop and Conference Proceedings 27, pp. 195-207 2012
- [4] A. Santoro et al. "One-shot learning with memory-augmented neural networks". Proc. of the International Conference on Machine Learning (ICML), 2016
- [5] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, pages 3630–3638, 2016
- [6] G Koch, R Zemel, and R Salakhutdinov. "Siamese neural networks for one-shot image recognition", ICML Deep Learning workshop, 2015
- [7] Graves, Alex, Wayne, Greg, and Danihelka, Ivo. Neural turing machines. arXiv preprint arXiv:1410.5401, 2014

- [8] S. Hochreiter, A Steven Younger, Peter R. Conwell ,
“Learning to Learn Using Gradient Descent”, Artificial
Neural Networks ICANN 2001, pp.87-94. Springer,
2001
- [9] Weston, J., Chopra, S., and Bordes, "A. Memory
networks", In ICLR, 2015b

Differentiation of communication signals to establish cooperation using Deep Q-Network

Hironobu Horiuchi [†] Hiroyuki Iizuka [‡] Masahito Yamamoto[‡]

[†] School of Engineering, Hokkaido University, Hokkaido, Japan

[‡] Graduate School of Information Science and Technology, Hokkaido University, Hokkaido, Japan

horiuchi@complex.ist.hokudai.ac.jp

Abstract: In multi-agent systems, coordination among agents can obtain better performances than a single agent can do. The communication signals play an important role to form the coordination. The previous study shows that the communication signals can be trained on Deep Q-network, however, the number of signals need to be predefined. In this study, we propose a learning method that can learn the necessary communication signals without predefining the number of signals. A cooperative task for multiple robots is designed and used to demonstrate the effectiveness of our method. Our results show that the communication signals are differentiated and used according to the situations, and the task can be achieved using the communication signals.

Keywords: Deep Q-Network, Multi-agent system, Self-organization of communication

1. Introduction

Multi-agent systems can achieve the task that a single agent cannot achieve. It becomes important how they can communicate to establish cooperative behaviors. From the evolutional point of views, Quinn and our studies show that the communication signals can emerge in the embodied motions without using any explicit communication channels [1, 2]. For the engineering purposes, the signals exchanging between agents are prepared in advance, however, it becomes a problem how the signals should be designed. To build the multi-agent system that uses an appropriate communication signal according to the situation is inevitable.

Foerster et al. (2016) proposed the method that the communication signals can be trained and self-organized by reinforcement learning with Deep Q-Network [3]. However, this method can only be applied to the tasks where the number of states is clear and the number of communication signals is predefined.

In this paper, we proposed a novel DQN-based reinforcement learning that can learn how to design the signals to establish cooperative behaviors. The agent is controlled by the Deep Q-Network (DQN), that is optimized to output the next action and the communication signal according to the other agents' communication signals and states. In order to evaluate our proposed method, we designed the task that requires communication to achieve. The task is that four robots are divided into two groups consisting of two robots, and each group needs to reach a different target area of two. We discuss the differentiation of the communication signals outputted by trained DQN.

2. Related Work

2.1 Q-learning

Reinforcement learning is the algorithm that trains the agent to maximize the reward given by the environment according to the action. Q-learning is one of reinforcement learning and optimizes the action-value function $Q_\pi(s, a)$. Choosing action a in state s and following the policy π , the action-value function $Q_\pi(s, a)$ is defined as follows.

$$Q_\pi(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1}), \quad (1)$$

where, s_t and a_t is the state and action at the time t respectively, and r_t is the reward given by the state s_t . $Q(s, a)$ is updated by the following.

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \{r_{t+1} + \gamma \max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)\}, \quad (2)$$

where, α is the learning rate, γ is the discount rate.

It is shown that $Q_\pi(s, a)$ is converged to the optimal function when $Q_\pi(s, a)$ is trained by the Q-learning in finite Markov Decision Process even if the random policy is used. However, an infinite time ($t \rightarrow \infty$) is assumed, and the time needed by converging is very long. ϵ -greedy and the method using Boltzmann distribution are often used as a policy to accelerate the convergence.

2.2 Deep Q-Network (DQN)

Deep Q-Network (DQN) trains and uses a deep neural network to estimate the action-value function $Q_\pi(s, a)$ [4]. DQN

can be used in the task having a lot of states because it is not necessary to define the number of states.

DQN is updated by the stochastic gradient descent, and the loss function $L_\theta(s, a)$ is defined as follows.

$$L_\theta(s, a) = \frac{1}{2}(target - Q_\theta(s, a))^2, \quad (3)$$

$$target = r_t + \gamma \max_{a_{t+1}} Q_{\theta^-}(s_{t+1}, a_{t+1}). \quad (4)$$

where, θ is the parameter of the Q-Network, and $Q_{\theta^-}(s, a)$ is the target network. The target network is used for stabilizing learning of DQN and is copied from $Q_\theta(s, a)$ at certain intervals. To stabilize the learning more, experience replay memory is useful [5]. The experiences are accumulated in replay memory, and DQN is updated by using data randomly sampled from the replay memory.

3. Simulation Environment

Before introducing our proposed method, we explain the task, simulation environment and a mobile robot.

3.1 Task

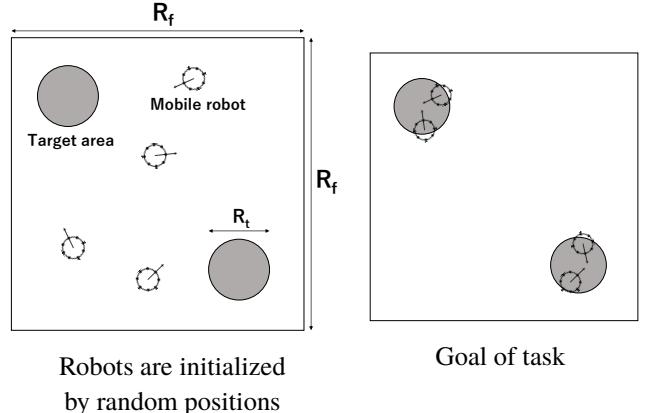
The simulation field is a square of $R_f \times R_f$, and there are two circular target areas of R_t in diameter. There are four mobile robots of R_r in diameter. Figure 1 (left) shows the simulation field.

The task is that four mobile robots are divided into two groups consisting of two robots, and each group needs to reach two different target areas (see Fig. 1 (right)). In order to achieve the task, the robot needs to inform that the robot is in the target area and know the number of robots in the area by using the communication signal.

We referred to Marocco et al. (2007) for the simulation field [6]. They simulated that the communication system is built when a task requiring cooperative action is given to agents in the multi-agent environment. The purpose of our study is to propose a novel DQN-based reinforcement learning that can learn how to design the signals to establish cooperative behaviors.

3.2 Autonomous Mobile Robot

The autonomous mobile robot consists of two wheels, the distance sensors, the communication sensors, and one ground sensor. The distance sensors can sense the distance to the other agents and walls of the field. The communication sensors can receive the communication signal that the other agents send, which means that robots can communicate with each other through the communication signal. The ground sensor can sense whether the agent is inside the target area or not. The robot decides the action and the communication signal to send based on the input of sensors.



Robots are initialized by random positions

Fig. 1: Field

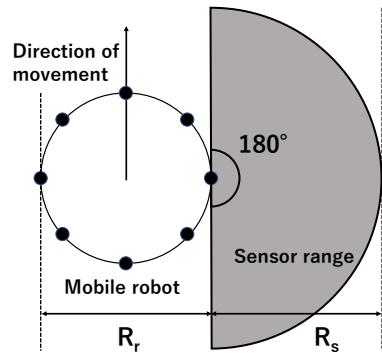


Fig. 2: Autonomous mobile robot and eight distance sensor

3.2.1 Distance Sensor

The mobile robot has eight distance sensors (Fig. 2). The distance sensors can sense the distance to the walls of the field and other agents.

The sensing range of each distance sensor is R_s , and the distance sensor senses only the closest object. The distance sensor value is $(R_s - d)$, where d is the distance of sensing the object, normalized in the range $[0.0, 1.0]$. In addition, the distance sensor value is -1 when the object does not exist within the range of R_s .

3.2.2 Communication Sensor

The mobile robot has the four communication sensors in the center of the robot. The four sensors can receive signals from the direction corresponding to each sensor. The angle ranges are $[-45^\circ, 45^\circ]$, $[135^\circ, -135^\circ]$, $[45^\circ, 135^\circ]$, and $[-45^\circ, -135^\circ]$ (see Fig.3). When there is no signal, the value of the communication sensor is -1 . If the sensor receives the multiple signals, the sum of the signal value is used as the input of the communication sensor.

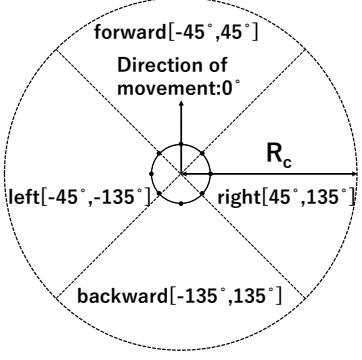


Fig. 3: Communication sensor

3.2.3 Ground Sensor

The ground sensor can sense whether the mobile robot is in the target area or not. The value of the ground sensor is 1 if the robot is in the target area; otherwise, the value of the ground sensor is 0. There is no difference between the two target areas.

3.2.4 Action of Autonomous Mobile Robot

DQN decides the action of the mobile robot, and the number of actions is eight. The actions are arbitrarily discretized, set and represented as the combinations of the right and left wheel velocities (v_r, v_l). The eight combinations ($v_r, [cm/s]$, $v_l, [cm/s]$) are (0, 0), (0, 10), (4, 8), (4, 6), (6, 4), (8, 4), (10, 0), and (10, 10). The position of the robots is calculated by the following.

$$x_t = x_{t-1} + \frac{1}{2}v\Delta t \cos(\theta_{t-1} + \frac{1}{2}\omega\Delta t), \quad (5)$$

$$y_t = y_{t-1} + \frac{1}{2}v\Delta t \sin(\theta_{t-1} + \frac{1}{2}\omega\Delta t), \quad (6)$$

$$v = \frac{v_r + v_l}{2}, \quad \omega = \frac{v_r - v_l}{R_r}, \quad (7)$$

where, θ_t is the direction of the robot at the time t . The action a is chosen by the following.

$$a = \arg \max_{(v_r, v_l) \in A} \pi(s, (v_r, v_l)), \quad (8)$$

$$\pi(s, (v_r, v_l)) = \frac{Q(s, (v_r, v_l))}{\sum_{k \in A} Q(s, k)}, \quad (9)$$

3.2.5 Communication Signal

The mobile robot sends communication signals in the [0.0, 1.0] range from the center of it. The value of communication signal does not change with distance, and the transmission range is R_c .

Table. 1: Structure of neural network and parameters

	Number of nodes	Activation function
Input layer	14	
FC1	256	ReLU
FC2	256	ReLU
Output(action)	8	ReLU
Output(signal)	1	sigmoid

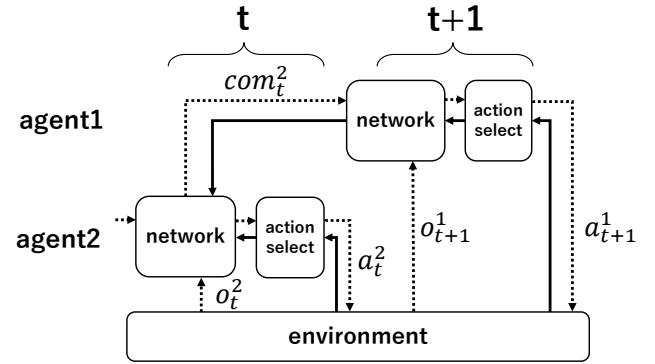


Fig. 4: Proposed method

Dotted line shows the input and output of DQN, and the error of network flows the solid line.

4. Proposed Method

In this study, we use DQN to control the robot and decide what signal is emitted. Table 1 shows the structure of the neural network.

4.0.1 Input of DQN

The input of DQN consists of 14 factors as follows:

- Eight distance sensors
- Four communication signals
- One ground sensor
- One communication signal emitted by itself one step before

4.0.2 Output of DQN

The output of DQN consists of 9 factors as follows:

- Eight action-values
- One communication signal

The action is chosen according to the selection probability $\pi(s, a)$ as follows.

$$\pi(s, a) = \frac{Q(s, a)}{\sum_{k \in A} Q(s, k)}, \quad (10)$$

where, A is a space of the actions.

Table. 2: Parameters of the simulation environment

Length of a side of the field R_f	120[cm]
Diameter of the target area R_t	25[cm]
Diameter of the robot R_r	10[cm]
Sensing range of distance sensor R_s	10[cm]
Transmission range of communication signal R_c	30[cm]
Δt	0.1[s]

In general, since input and output between agents are independent, DQN is trained to minimize the error function defined as equation (4). In our model using communication signals, however, the input and output of the two agents' networks are connected as shown in Fig. 4. We describe our DQN using Agent 1 and Agent 2 in Fig. 4. The action-value of Agent 1 is calculated using the communication signal outputted by Agent 2. The error generated by the calculation of the action-value in Agent 1 can be propagated to Agent 2. Our proposed method is to propagate the error to the other agents through the input and output. This is an extension of Differentiable Inter-Agent Learning (DIAL) proposed by Foerester et al. [3]. The difference between our DQN and DIAL is the value of the communication signal. DIAL outputs the discrete values [0 or 1], however, our DQN outputs the continuous values [0, 1]. Our DQN can outputs signals within the range of [0, 1] regardless of the number of states of the agent.

4.1 Reward

The rewards are defined as follows.

$$r_t = \begin{cases} 1.0 & (d \leq \frac{R_t}{2}, 1 \leq n \leq 2) \\ -1.0 & (d \leq \frac{R_t}{2}, n \geq 3) \\ 0.0 & (d > \frac{R_t}{2}), \end{cases} \quad (11)$$

where, d is the distance between the mobile robot and the center of the target area, and n is the number of robots in the target area. The best reward can be obtained when two agents stay in each of the two target areas. The worst reward is given when three or more agents stay in a same target area.

5. Experiment

We show that the robot can achieve the task by using the communication signals. The task requires the communication signals between the robots to achieve.

5.1 Experimental Conditions

An episode ends when the task is achieved or when the time reaches 1000 steps. The positions and the directions of the

Table. 3: Learning parameters

Learning rate α	1.0×10^{-4}
Discount rate γ	0.9
Batch size	32
Optimizer	Adam [7]
Replay memory size	160,000

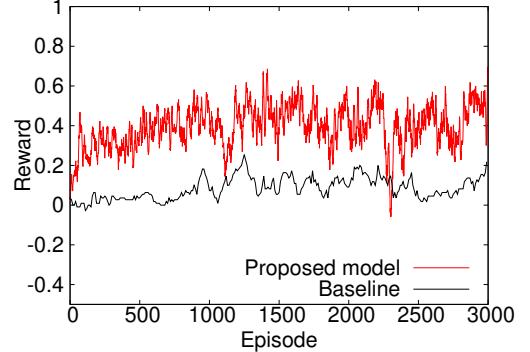


Fig. 5: Moving average of the rewards

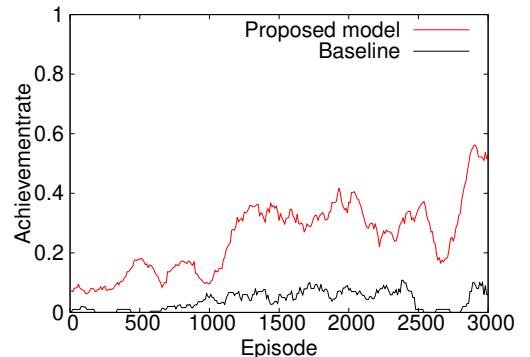


Fig. 6: Moving average of the achievement rate of the task

agents at the beginning of the episode are initialized by random positions excluding the target area and by random directions [0°, 360°]. Table 2 shows the parameters of the simulation environment. Table 3 shows the training parameters.

5.2 Experimental Result

In order to show that the agents can achieve the task by using communication signals, we compare the two models. One is our proposed model, and another is the model trained without using the communication signals (baseline). Figure 5 shows the moving average of the rewards that are given in the terminal step of episodes. Figure 6 shows the moving average of the achievement rate of the task. These figures show that the acquired rewards and the achievement rate of the task have increased in the proposed method, while the learning does not proceed well with the baseline.

The robots do not know the exact number of robots in the

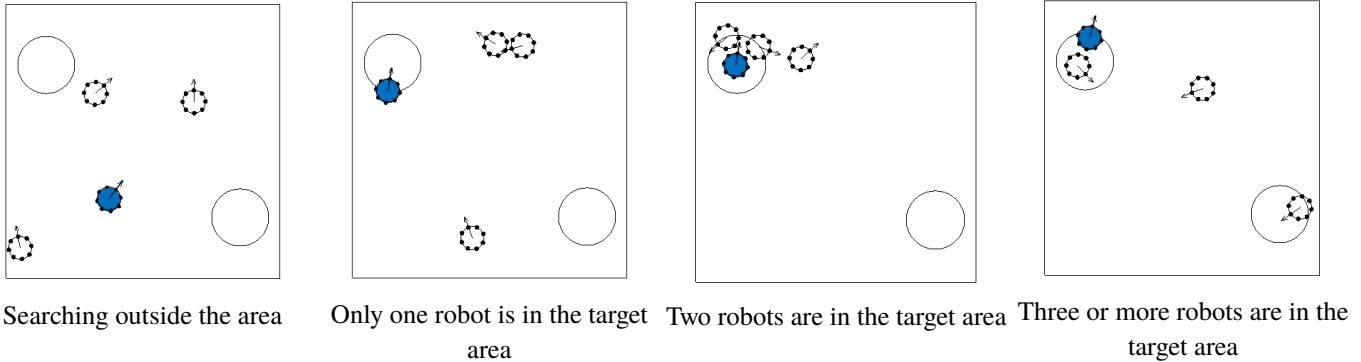


Fig. 7: The four situations

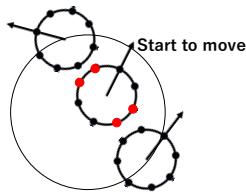


Fig. 8: The robot starts to move
The distance sensors of red sense objects

same target area in the case of the no-communication condition. Without communication, the robot leaves the target area if the distance sensors are activated strongly from two different directions, which means that the agent interprets it as the situation where three agents stay in the same target area. This is partially correct (Fig. 8) however it fails as shown in Fig. 9.

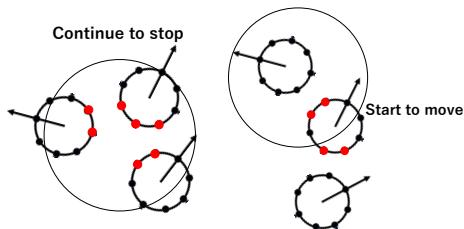


Fig. 9: The failure cases by not using communication signals
The distance sensors of red sense objects

It becomes very difficult to achieve the task without communication signals, which causes the differences of rewards obtained during training and task achievements.

The achievement rate of the task improves in the case of communication signals trained by our proposed DQN method. Although the moving average of the task achievement remained around 50%, the best achievement rate in the

epochs was 92%. The reason why there is a big gap between the rates is because the learning of DQN is unstable. The modification of the Q-network does not necessarily lead to the better or equal performances. In the most cases of the failure trials, they failed to find the target areas.

These results mean that agents use communication signals to achieve the task.

5.2.1 Division of the Communication Signals

The communication signal emitted by the robots roughly divided into four situations. Figure 7 shows the four situations.

The example of how those signals are used is explained as follows. The mobile robot moves around the field looking for the target area and emits about [0, 0.2] communication signal (Fig. 10(a)). The robot arriving in the target area stops and emits about [0.2, 0.4] communication signal (Fig. 10(b)). When another robot comes in the same target area, the robot emits about [0.4, 0.9] communication signal (Fig. 10(d)). The other robots come in the same target area, all robots in the same area emit about [0.9, 1.0] communication signal, and they start moving and leave (Fig. 10(c)). They move until the number of robots in the same area is two. Figure 10 shows the example of the communication signals emitted by robots in one episode. Figure 11 shows the histogram of the communication signals emitted by robots in 20 episodes. It can be seen that the communication signals are divided into four types and they are used without overlapping the current states.

We trained DQN by changing the initial value of the network three times. The value ranges of the communication signals corresponding to the four situations are differently self-organized. The order of signals that represent the number of agents in the target areas does not change, i.e., if it is not on the target, it emits the small signal and if three or more agents stay in the target, they emit the strong signal.

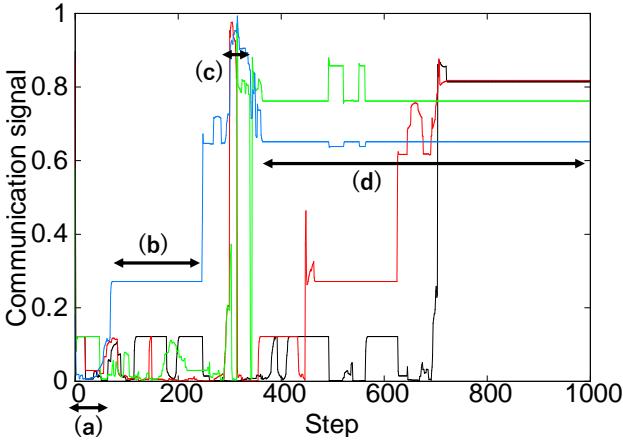


Fig. 10: Signals emitted by robots in one episode
The colors mean different agents.

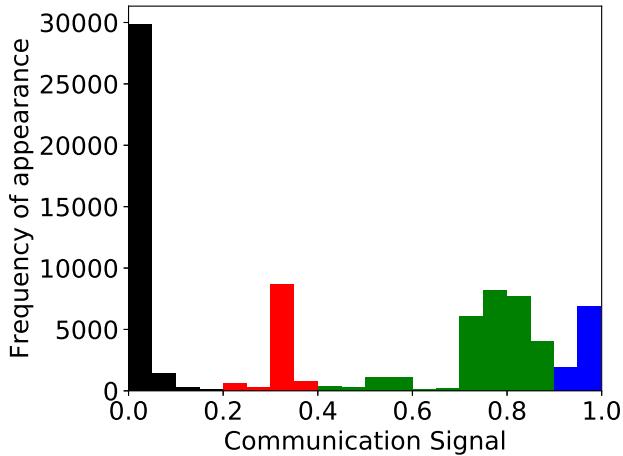


Fig. 11: Histogram of the communication signals
The bars are colored depending on the situation of the emitting agent. Black : not on a target, Red : alone on a target, Green : sharing a target with another agent, Blue: sharing a target with other two or three agents.

6. Discussion

The robot always outputs the signal of small value when the robot is outside the target area. If the robot in the area receives the large signal from the robot searching outside the area, the input of the DQN changes greatly, but the agent in the target area should stay there, which means that the DQN should not change the outputs of action-state values so much. Therefore, DQN seems to be trained to output the small signals when the robot is outside the area. If the robot in the area outputs the small signals, it is hard to know the number of robots in the same target area. Also, it is susceptible to the signal outputted by robots outside the target area. Therefore, the value of the signal outputted by the robot also seems to be

large when the number of robots in the target area increases. Those communication signals were self-organized through the reinforcement learning of our proposed method.

There are some other methods to perform cooperative actions using reinforcement learning in a multi-agent environment [8, 9, 10]. ACCNet backpropagates errors between networks of different agents to learn communication [8]. Although ACCNet uses Actor-Critic instead of DQN, it is similar in that they also use the error backpropagation to establish communication signal. However, they propose a global communication module called coordinator communication channel that receives the communication signals from all agents and outputs the signals to agents. Building a path for back-propagating errors between agents is one of the basic ideas when making agents learn communication in reinforcement learning. Our proposed method can organize the communication signals in a local manner where all agents have own communication channel and the inputs of the receiver and outputs of the sender in the communication range are connected and the error propagates. As a result, they self-organize the effective communication signals to solve the task. For the future work, we will expand the model in the time direction using a recurrent neural network. However, it is expected that propagation of error becomes very complicated due to paths among multiple agents.

7. Conclusion

In this paper, we proposed the DQN model that can train the output of the communication signal. In the multi-agent environment, the output of the communication signal was differentiated to achieve the task. When the number of robots in the target area is one or two, the robot outputs different communication signals, however, the robots cannot take appropriate action corresponding to the signals. For example, the robot moves toward or away from the robot that outputs signals in the target area. In the future, we will strive to improve the model and make robots achieve task efficiently by using the neural network with internal memory or Asynchronous Advantage Actor-Critic.

References

- [1] Matt Quinn. Evolving communication without dedicated communication channels. In *European Conference on Artificial Life*, pages 357–366. Springer, 2001.
- [2] Naohiro Shibuya, Hiroyuki Iizuka, and Masahito Yamamoto. Evolution of communication through differentiation of communicative and goal-directed behaviors. *Artificial Life and Robotics*, 23(2):225–234, 2018.
- [3] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

- [4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [6] Davide Marocco and Stefano Nolfi. Emergence of communication in embodied agents evolved for the ability to solve a collective navigation problem. *Connection Science*, 19(1):53–74, 2007.
- [7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [8] Hangyu Mao, Zhibo Gong, Yan Ni, and Zhen Xiao. Accnet: Actor-coordinator-critic net for "learning-to-communicate" with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1706.03235*, 2017.
- [9] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.
- [10] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

Deep Reinforcement Learning Method Enabling Stable Learning in Multi-Agent Environment

Kenji Hirata [†] Hiroyuki Iizuka [†] Masahito Yamamoto [†]

[†] Graduate School of Information Science and Technology, Hokkaido University, Hokkaido, Japan
hirata@complex.ist.hokudai.ac.jp

Abstract: There are several studies about acquisition of cooperative behaviors using reinforcement learning in multi-agent environments. However, there is a difficulty in applying reinforcement learning to multi-agent environments because Markov property is not usually satisfied in multi-agent environments. In this study, we propose a method that enables stable learning in multi-agent environments. In our proposed method, the learning process is split into the phases learned by each agent and the learning is performed alternately between the agents. Our proposed method is compared with the normal method that the agents learn at the same time in the game of tag environment. The agents trained by our proposed method acquire the cooperative behaviors that cannot be acquired by the normal method.

Keywords: Multi-agent, Cooperation, Reinforcement learning, Deep Q-Network

1. Introduction

Cooperation among multiple agents is necessary to achieve a goal that cannot be solved by a single agent. There are several studies about acquisition of cooperative behaviors in multi-agent environments. Tampuu *et al.* (2017) showed that cooperative behaviors can be acquired in the tennis game environment of Atari by using deep reinforcement learning [1]. The agent that learns with the learning agent is more robust than the agent that learns with a fixed agent. The environment of Tampuu's research is not sensitive to the changes of the other agent's actions in a sense that the learning is performed stably. The individuality of the cooperative behaviors makes the learning more stable. However, in general, Markov property is not satisfied in multi-agent environments because multiple agents learn independently and change action policy dynamically. Therefore, reinforcement learning becomes unstable in the environments that sensitive to the changes of the other agent's actions.

Lowe *et al.* (2017) proposed the deep reinforcement learning method named Multi-Agent Deep Deterministic Policy Gradient (MADDPG) that enables stable learning in multi-agent environments [2]. MADDPG uses observation and action information of other agents to estimate the values of the environment and decide the own next action by the policy function. Using this information help learning stable because there is no missing information that breaks Markov property for learning. Therefore, MADDPG cannot be applied in incomplete information games.

In this study, we propose a deep reinforcement learning method that enables stable learning in multi-agent environ-

ments without observation and action information of other agents.

2. Background

2.1 Markov Decision Process (MDP)

MDP is discrete time stochastic control processes that satisfy Markov property. At each time step, the process is in the state s and the agents select the action a . Then the process transits to next state s' and the agents receive the reward r according to the transition. s' and r depend only on s and a . In other words, these do not depend on past states of s . MDP is useful for studying reinforcement learning.

2.2 Q-learning

Q-learning [3] is a kind of reinforcement learning. In Q-learning, the learning is performed by sequentially updating the action-value function $Q_\pi(s, a)$ in repeated trials. $Q_\pi(s, a)$ represents the value of selecting the action a at the state s under the policy π . $Q_\pi(s, a)$ function is defined as the expected reward when the agent selects a at s and then follows π :

$$Q_\pi(s, a) = r + \gamma \max_a Q_\pi(s', a'), \quad (1)$$

where r_{t+1} represents a reward, and γ is a discount factor. This Q function is updated as:

$$Q_\pi(s, a) \leftarrow Q_\pi(s, a) + \alpha \{r + \gamma \max_{a'} Q_\pi(s', a') - Q_\pi(s, a)\}, \quad (2)$$

where α represents a learning rate. It is known that the Q function converges to the optimal function even if the agent

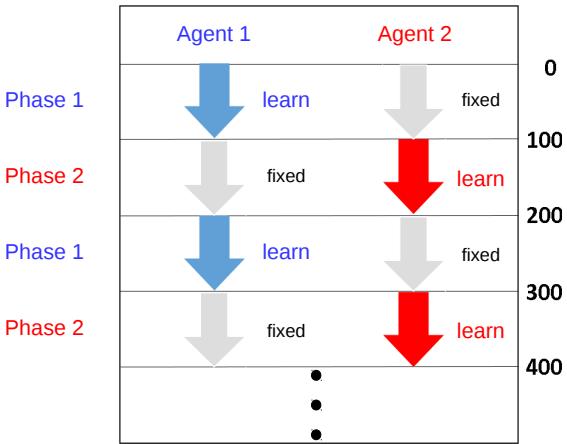


Fig. 1: Proposed method

acts under the random policy. However, it is difficult to converge within a realistic time because it takes a lot of time. Therefore, the action selection method such as the ϵ -greedy method and the roulette selection method is used. In this study, we use the ϵ -greedy method.

2.3 Deep Q-Network (DQN)

DQN [4] is a method combining Q-learning and Deep Learning. DQN approximates the action-value function by a deep neural network called Q-network Q_θ . Q_θ is learned by minimizing follows loss function:

$$L(\theta) = \frac{1}{2} \{Q_\theta(s, a) - (r + \gamma \max_a Q_{\theta'}(s', a'))\}, \quad (3)$$

where $Q_{\theta'}$ is a neural network called target-network that updates its weights from Q_θ at regular intervals. Target-network stabilizes learning because the weights of this network are fixed for a while.

3. Proposed method

Q-learning including DQN has been applied to multi-agent environments. However, there is a difficulty in applying Q-learning to multi-agent environments. In order to converge the learning of an agent, the environment including the other agents must be stationary. However, multi-agent environments are non-stationary for agents and not MDP because each agent independently learns and updates their policies. Therefore, the convergence of Q-learning is not guaranteed. In this section, we propose a method that approximates multi-agent environments to MDP environments.

In our proposed method, the learning process is split into the phases learned by each agent. In the case of two agents, the learning is performed by repeating the phases alternately. Markov property is guaranteed in a whole experi-

ment because only one agent learns and the other agents follow their fixed policy with ϵ -greedy in the phase. In the simple reinforcement learning methods where multiple agents are trained at the same time, the agents need to learn and change policy for how to behave with the other agents, whose policy change dynamically in the same way. That is why the environment dynamics including the other agents' behaviors become more increase for each agent as the number of agents increases. In contrast, in our proposed method, the complexity of the environment dynamics does not increase even if the number of agents increases because the fixed agents behave deterministically. The replay memory [5] for DQN is initialized to empty at the beginning of the phase because the accumulated past transitions in response to the old partners' policy cannot be used for the current environment.

Consider the environment where N agents exist. The phase number $n = e/L \pmod N$ indicates a phase where the agent n is trained and the others are fixed. e is the current episode number. L is the length of a phase. After the agent learns for L , it will be the learning phase of the next agent. Fig. 1 shows the overview of our proposed method in the case of two agents.

4. Environment

4.1 Multi-Agent Particle Environment

In order to show that the agents implemented with our proposed method can learn stably in multi-agent environments and that it is possible to acquire cooperative behaviors, we use Multi-Agent Particle Environment [6], which is a two-dimensional continuous space where N agents and L landmarks exist. This environment can set various scenarios. The number of agents and their purpose differ for each scenario, that is, cooperation, competition or a combination of them emerge for each scenario. We use the following simple-tag scenario which requires cooperation.

4.2 Simple-tag

Simple-tag scenario is the game of tag. This scenario consists of N_{prey} prey agents, $N_{predator}$ predator agents, and L landmarks. In this study, $N_{prey} = 1$, $N_{predator} = 2$, $L = 0$. They are randomly placed in a circular field. Fig. 2 shows simple-tag environment. The purpose of the prey is to escape from the predators. The purpose of the predators is to chase and touch the prey. However, because the maximum speed of the prey is faster than the predator, the cooperative behaviors of multiple agents are required for the predators to achieve their purpose. Each agent observes its own absolute position and velocity, the relative positions of the other agents and the landmarks. There is inertia in the environment. Each agent changes its velocity by adding the acceleration every step. Table1 shows the parameters of the field and the agents.

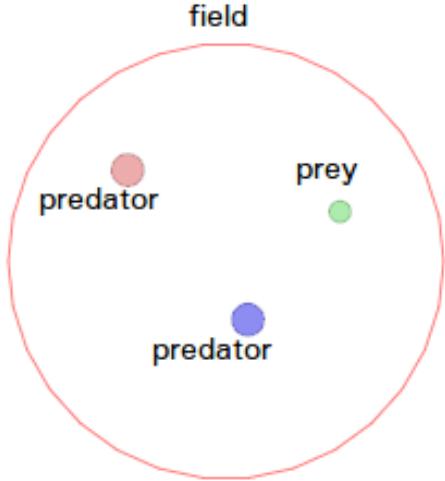


Fig. 2: Simple-tag environment

Table. 1: The parameters of the field and the agents

The radius of the field	1.0 [cm]
The radius of the predator	0.075 [cm]
The radius of the prey	0.050 [cm]
The maximum speed of the predator	0.10 [cm/s]
The maximum speed of the prey	0.13 [cm/s]
The maximum acceleration of the predator	0.03 [cm/s ²]
The maximum acceleration of the prey	0.05 [cm/s ²]

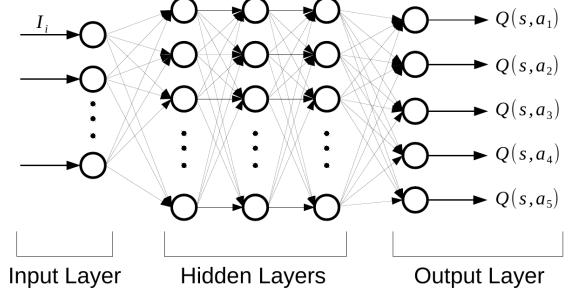
4.3 Predator agent

Because the predators require the cooperative behaviors, we focus on the predators. The predators are trained by our proposed method. Each agent is controlled by different DQN. Fig. 3 shows the structure of the network of DQN. The network is a feed forward neural network consists of an input layer with seven nodes, three hidden layers with 64 nodes and an output layer that outputs the action-values for five actions.

4.3.1 Input

Originally, the predator observes its own position p_x, p_y and velocity v_x, v_y in global coordinates and the relative position of the other agents p'_x, p'_y ($i = 1, 2$). However, we convert the observations for several reasons. First, the predator's own absolute position is converted into the distance from the center of the circular field R for reducing the number of possible states:

$$R = \sqrt{p_x^2 + p_y^2}. \quad (4)$$



	Number of nodes	Activation function
Input	7	
FC1	64	PReLU
FC2	64	PReLU
FC3	64	PReLU
Output	5	Linear

Fig. 3: The structure of DQN

Accordingly, the predator's own velocity is converted into v'_x, v'_y so that it is based on the position vector:

$$v'_x = v_x \cos(-\alpha) - v_y \sin(-\alpha), \quad (5)$$

$$v'_y = v_x \sin(-\alpha) + v_y \cos(-\alpha), \quad (6)$$

$$\alpha = \arctan \frac{p_y}{p_x}. \quad (7)$$

There is no problem even if these conversions are performed because the circular field is isotropic. Furthermore, the relative position of the other agents is converted from Cartesian to polar coordinates ($(p_x^i, p_y^i) \rightarrow (d_i, \theta_i)$):

$$d_i = \sqrt{p_x^i{}^2 + p_y^i{}^2}, \quad (8)$$

$$\theta_i = \arctan \frac{p_y^i}{p_x^i} - \arctan \frac{v'_y}{v'_x}. \quad (9)$$

Then, the inputs for the information of the other agents are $\frac{\cos \theta_i}{d_i}, \frac{\sin \theta_i}{d_i}$. The input becomes stronger as the distance becomes closer and the input becomes weaker as the distance becomes farther.

Consequently, all inputs are the seven-dimensional real vector $\{R, v'_x, v'_y, \frac{\cos \theta_1}{d_1}, \frac{\sin \theta_1}{d_1}, \frac{\cos \theta_2}{d_2}, \frac{\sin \theta_2}{d_2}\}$. Fig. 4 shows the variables of the inputs.

4.3.2 Action

Although the action space of simple-tag scenario is continuous, it is necessary to set a discrete action space because DQN cannot deal with a continuous action space. The predator takes one of five actions of accelerating to front, rear, left

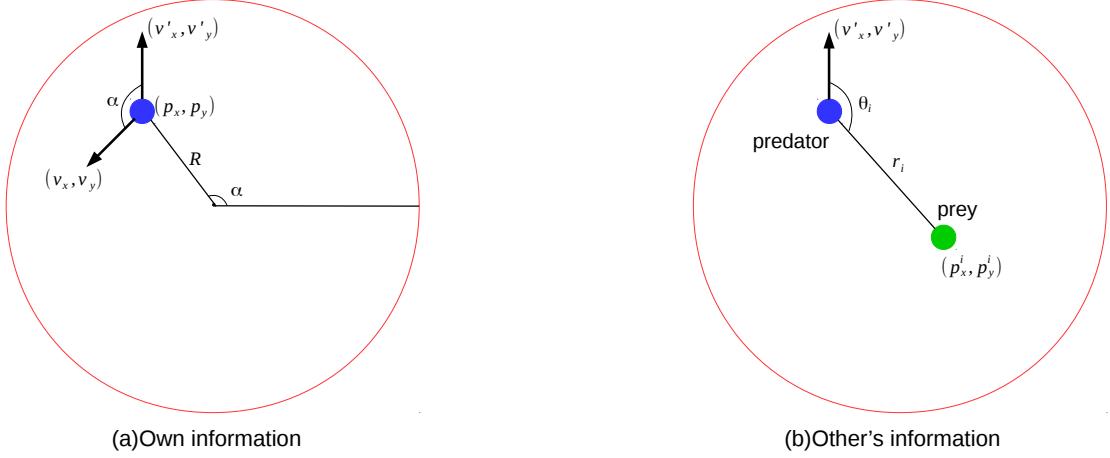


Fig. 4: Information of agent's own and other's agent.

and right or not accelerating. The direction is based on predator's own velocity.

4.3.3 Reward

When any predators touch the prey, all predators obtain +1. Otherwise, they obtain ± 0 .

4.4 Prey agent

The prey is controlled by the simple rule-based algorithm. The prey receives the power from two predators and the boundary of the circular field. The prey moves by these resultant forces:

$$F = - \sum_{pos \in POS} \frac{pos}{|pos|^2},$$

where POS is a set of the relative positions of two predators and the nearest boundary. With this rule, the prey escapes from the predators in the field.

5. Experiment

5.1 Experimental setting

DQN model is trained by repeating episodes. Each episode starts two predators and one prey are randomly placed in the circular field. The episode ends when any predators touch the prey or when it becomes 200 steps. Since the maximum speed of the prey is 1.3 times faster than the maximum speed of the predator, the predators cannot touch the prey with random actions. Therefore, the successful experience is not accumulated and the learning of the predators does not proceed. For this reason, at the beginning of the learning, we set the maximum speed of the prey to 0 and it gradually speeds up

to 1.3 times the maximum speed of the predator as the learning of the predators progress. The predators exchange the learning phase every 100 episodes.

5.2 Experimental result

We train the DQN with our proposed method and the normal method that two predators learn at the same time and then evaluate them by averaging 40 different experiments. In each experiment, the DQN with each method is trained for six to seven hours on a single NVIDIA GeForce GTX 1050 GPU.

Figure 5 shows the simple moving average (SMA) of the number of the steps until the predators touch the prey. Table 2 shows the mean of the number of the steps until the predators touch the prey when the network is fixed with the best DQN in all experiments. The t-test was conducted on this result, and the significant difference was observed between methods ($p = 0.00107$). Since the diameter of the circular field is 2.0cm and it takes 20 steps to cross it at earliest, the catching time less than 40 time steps indicated by our proposed method is fast enough considering achieving the task of the tag.

Figure 6 shows the comparison of the behaviors between our proposed and the normal methods. In our proposed method, the predators chase down the prey from both sides to the boundary. Consequently, the prey cannot escape from them although it can move faster than the predators. On the other hand, in the normal method, either predator agent chases the prey, however, the other one goes away from the field. Even the single predator sometimes catches the prey successfully by approaching it from the vertical direction. However, it is not as successful as the coordinated way.

In our proposed method, the predators acquire the cooperative behaviors that chase down the prey from both sides to

Table. 2: Mean of the number of the steps until the predators touch the prey. n is the number of the samples, M is the mean and SD is the standard deviation.

	n	M	SD
Proposed method	1000	37.516	24.740
Normal method	1000	41.354	27.537

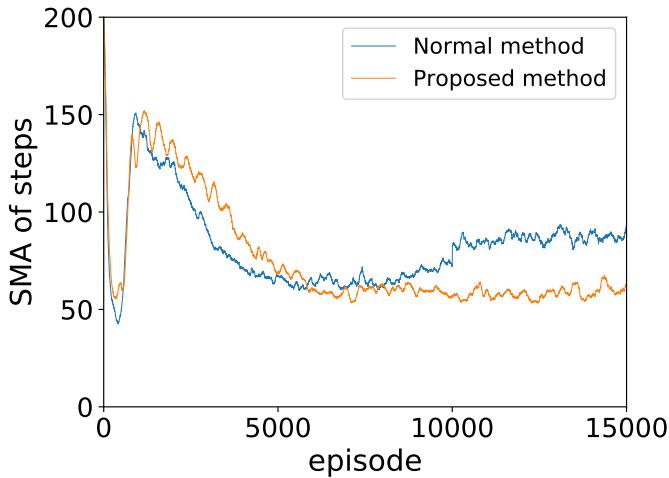


Fig. 5: SMA of the number of steps until the predators touch the prey.

the boundary. In order to acquire such behaviors, the predators must learn to keep pace with the other predator. However, it is difficult in the environment in which the behaviors of the agents change rapidly. Therefore, in the normal method, the predators cannot acquire such behaviors. Only either predator chases the prey and the other one goes away. This is attributed to that the predator learns to not disturb the behaviors of another predator. By contrast, in our proposed method, the predators can learn to keep pace with the other predator because the policy of the other predator is fixed for a while.

6. Discussion

Our proposed method can acquire the cooperative behaviors that cannot be acquired by the normal method; therefore, our proposed method is useful for learning in the multi-agent environment. However, we have only experimented with our proposed method in Simple-tag scenario. This environment is almost complete information environment because the observation has no limitation of distance. We need to confirm that our proposed method can be applied to the more incomplete information environments. Our proposed method can be extended to incomplete information environments without major changes because our proposed method does not use the information of the other agents.

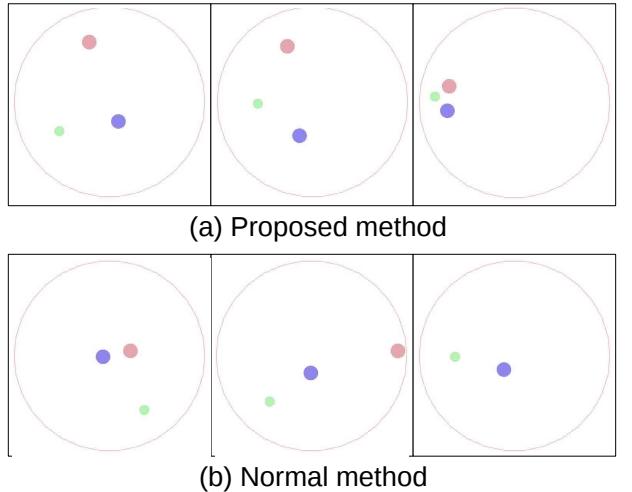


Fig. 6: The comparison between our proposed and the normal methods. In our proposed method, two predators chase down the prey from both sides to the boundary. In the normal method, one predator goes away from the field.

On the other hand, MADDPG cannot be extended to incomplete information environments as it is because the method uses the information of the other agents. The information must be guessed from the limited information if MADDPG is applied to incomplete information environment.

7. Conclusion

This study proposed a method that enables stable learning in multi-agent environments. The agents could learn stably by splitting the learning processes and the learning was performed alternately between the agents. Therefore, the agents acquired the cooperative behaviors that could not be acquired by the normal method. We have not experimented our proposed method in other environments. In particular, it is necessary to confirm whether our proposed method works well in incomplete information environments. Furthermore, our proposed method cannot apply to environments that have continuous action space. Therefore, even though the environment in this study has continuous action space, we set discrete action space. The actor-critic method that can output the continuous values is widely used for continuous action space tasks [7]. In future work, we will use the actor-critic method such as Deep Deterministic Policy Gradient as a learning method [8].

References

- [1] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition

- with deep reinforcement learning. *PLoS one*, Vol. 12, No. 4, p. e0172395, 2017.
- [2] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.
 - [3] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, Vol. 8, No. 3-4, pp. 279–292, 1992.
 - [4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, p. 529, 2015.
 - [5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
 - [6] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.
 - [7] Jun Morimoto and Kenji Doya. Reinforcement learning of dynamic motor sequence: Learning to stand up. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, Vol. 3, pp. 1721–1726. IEEE, 1998.
 - [8] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.