Evaluation for Acquiring Method for Agents' Actions Using Pheromone Communication in Multi-Agent System

Hisayuki Sasaoka

Abstract—**This paper has presented acquiring method for agents' actions using Ant Colony System (ACS) in multi-agent system. ACS is one of powerful meta-heuristics algorithms and some researchers have reported the effectiveness of some applications with the algorithm. I have developed agents with proposed method in a multi-agent system. The final goal of this research is an achievement of co-operations for hetero-agent in multi-agent systems. In this research for implementation for one type of agents is the first step of this goal. Then I have reported results of evaluation experiments using these agents.**

Index Terms—**Machine learning, multi-agent system, swarm intelligence, ant colony optimization.**

I. Introduction

Recently, some researchers have reported the effectiveness of systems installed swarm intelligence algorithms [1]-[3]. Ant Colony Optimization (ACO) and Ant Colony System (ACS) have become a very successful and widely used in some applications. These algorithms have been used in programs for network routing, traffic control programs, Traveling Salesman Problem (TSP) and so on [4]-[6].

In real ants' feeding actions, they are able to find the shortest path from a food source to their nest by pheromone information. Pheromone is one of chemical materials and ants deposit pheromone on a path between a food source to their nest and it become one candidate solution. In a case that other ants trail a path deposited pheromone, the candidate is reinforced. Moreover pheromone disappears into air as time progresses. Ants have done these processes iteratively. The behavior of real ants has inspired ACO and ACS. The system based on ACO and ACS are used artificial ants cooperate to the solution of a problem by exchanging information via pheromone.

In this paper, we would like to propose our optimization method based on ACO algorithm and apply to agents of fire brigade agents in my team on RoboCup Rescue Simulation System [7], [8]. The travelling salesman problem (TSP) has no noise for solving and all of distances between each city are given in advance. Moreover their situations have never changed for each simulation steps. However situations or outer information in environment is always changing in the real world, dynamically. In some cases, we are disable to know cues to solve a problem in advance. In other case, some outer noise gets information erased or interpolation them. On the other hand, in a situation of RoboCup rescue simulation system, agents need to handle huge amount of information and take actions dynamically. Therefore, a simulation system of RoboCup rescue is a very good test bed for multi-agent research.

Manuscript received October 16, 2012; revised December 27, 2012.
Hisayuki Sasaoka is with Asahikawa National College of Technology (e-mail: author@ boulder.nist.gov).

II. Our Approach

In our previous study [9], soccer agents based on our proposed approach for RoboCup Soccer Simulation System are able to decide a direction of kicking soccer ball and they are able to do effective action in the simulation soccer games. We have done some experiments with these agents and have reported that the agents improved their abilities of getting scores in soccer games. From the results of the evaluation experiment, we have confirmed the effectiveness of our method. In this section, I would like to explain basic idea of ACO and the related works and show a multi-agent system of RoboCup Rescue Simulation.

A. ACO, Related Works and Our Proposed Algorithm

Agent programs which run based on ACO have decided their actions by a probabilistic value and a generated random value. In each step they have calculated the values and have taken one choice. The probabilistic value is calculated by formula (1).

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in J_i^k}\left[\tau_{il}(t)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} & (if\ j \in J_i^k) \\ 0 & \text{(in other case)} \end{cases} \tag{1}$$

In this formula, a value of τ_{ij} means the value of associated pheromone trail on arc (i, j). A value *of* i means a current position in a simulation field. A value of j means a choice to move to the position. On the other hand, a value of η_{ij} means a heuristic value which is available a priori, α and β are two parameters which are determined by a relative influence of pheromone trail and the heuristic information. The value of η_{ij} has to be decided in advance.

Pheromone trails are updated and pheromone evaporation is able to be calculated by formula (2).

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{\ k}(t) \tag{2}$$

This value of ρ means the pheromone evaporation rate, which is between 0 and 1. A value of $\Delta\tau$ is amount of deposited pheromone and is calculated by formula (3). A value of Q is scored by results of agents' action.

$$\Delta\tau_{ij}(t) = \begin{cases} \sum_{k=1}^{m} \dfrac{Q}{L^k(t)} & (if\ \ (i, j) \in L) \\ 0 & (other) \end{cases} \tag{3}$$

Stützle, T. and Hoos, H.H. proposed *MAX-MIN* Ant

System (MMAS) [10]. It derived from Ant System and achieved an improved performance compared to AS and to other improved versions of AS for the TSP. The basic ideas are below,

1) only one ant who is the best in the process is allowed to update pheromone trails
2) pheromone trail values are restricted
3) trails are initialized to their maximum value.

In our algorithm, the range of pheromone trail value is decided by hand from preliminary experiment. Moreover, we have confirmed that there is a noise of pheromone trail in the initial step of updating pheromone trails. Then, our algorithm has calculated by equal (2)' in the initial step.

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \rho_{init}\Delta\tau_{ij}^{\ k}(t) \qquad (2)$$

This $\rho_{\ init}$ aims to cut down effect from the noise and the value is also decided in advance.

B. About RoboCup Rescue Simulation System

I have employed RoboCup Rescue Simulation System as a test-bed to process a task allocation problem where the tasks arrive dynamically and can change in intensity. This system has its server and four different types of agents. They are a fire-brigade agent, a police-force agent, an ambulance agent and a civilian agent. They hold correspondence with each program and have been able to simulate a situation of a city's disaster. Moreover the system has been able to simulate different situations in each conditions and maps for simulators.

The RoboCup Project System intends to promote researches which scope the disaster mitigation, search and rescue problems. Then we need to develop three types of agents, which are a fire-brigade agent, a police-force agent and an ambulance agent. Fig. 1 shows a screen copy which the simulation system performs. It shows a map of city and deep gray rectangles indicates buildings and light gray rectangle shows roads. Black parts on roads means blockades and agents cannot go through the place at the points. In this figure, red filled circles indicate fire-brigade agents and a mark of fire plug means a center of fire-brigade. Blue filled circles indicate police-force agents and a mark of policeman helmet means a center of police-force agents. White filled circles indicate ambulance agents and a mark of white cross means a center of ambulance agents. Green filled circles indicate civilian agents and a mark of red house means an emergency refuge center.

Fig. 1. An Example of Simulation Map in RoboCup Rescue Simulation System

The simulation's time and score has been controlled by simulation server program. The score is calculated by the program, which is depending on the situation of damaged buildings and affected civilian.

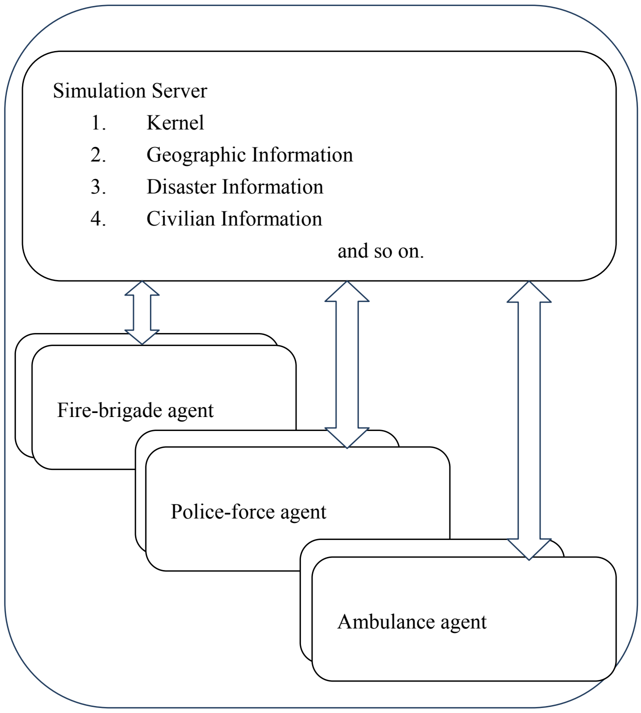

Fig. 2. A relation between simulation server and agent programs

Fig. 2 means a relation between a server program and each agent programs. These programs can hold correspondence with each other by UDP/IP protocol.

C. Basic Algorithm for Fire-Brigade Agents

A basic algorithm for fire-brigade agents in RoboCup Rescue Simulation System is below,

1) In a case that agents has enough water to extinguish a fire,
   - and they have known a place which buildings are burning, they go to the place.
   - and they have not known the place, they move randomly.
2) In a case that agents has no water to extinguish a fire,
   - and they have known a place which supply water to them, they go to the place.
   - and they have not known the place, they move randomly.

Then we have implemented our algorithm into these actions. Our algorithm is below,

1) In a case that agents have been supplied water, they deposit pheromone on a path.
2) In a case that agents has no water to extinguish a fire,
   - and they have known a place which supply water to them, they go to the place.
   - and they have not known the place, they select one path with a probabilistic value which is calculated by a value of pheromone's concentrations.

III. Preliminary Experiments

A. Procedures

We have done some preliminary experiments to compare results of related algorithms. In the experiments, one agent has run on a maze from a point of start to a point of goal. Fig. 3 shows the maze in a computer magazine [11]. A blue square is a place of start and a yellow square is a place of goal. Light gray squares are passage ways and light green squares are walls in a maze. Then an orange square means an agent and it can move on the passage ways. The step of the shortest path is 22 steps in this maze.

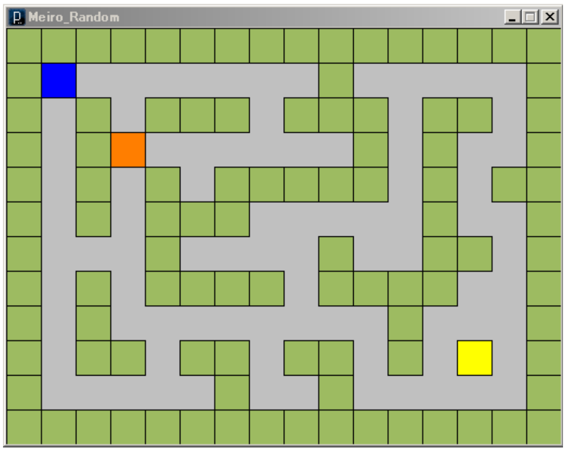

Fig. 3. A maze of this preliminary experiment

We have employed three types of agent program and they are below,

- agent A runs randomly.
- agent B implements an algorithm of Q-learning [12] and runs.
- agent C implements our proposed algorithm and runs

We have repeated 50 times by each agent.

B. Results

Table I has shown results of experiments. "Averages" column is a mean number of steps by each agent. "Maximum" column is the maximum number of steps in 50 trials by each agent. "Minimum" column is the minimum number of steps in 50 trials by each agent. From these results, agent C could achieve the best result slightly.

TABLE I: Results of This Experiment

	Team A	Team B	Team C
1	19.124	18.540	19.486
2	17.009	17.920	17.617
3	19.378	17.577	15.059
4	18.035	**20.840**	18.131
5	18.079	18.939	20.247
Average	18.325	**18.763**	18.108

Moreover we have repeated 200 times with agent C. Fig. 4 has shown a learning curve from 1st to 200th trial. In this figure, x-axis is the number of trial and y-axis is the number of steps by agent C in each trial. We can see the convergence of the learning process by our proposed algorithm. Agent C could find a shortest path in 146th trial.

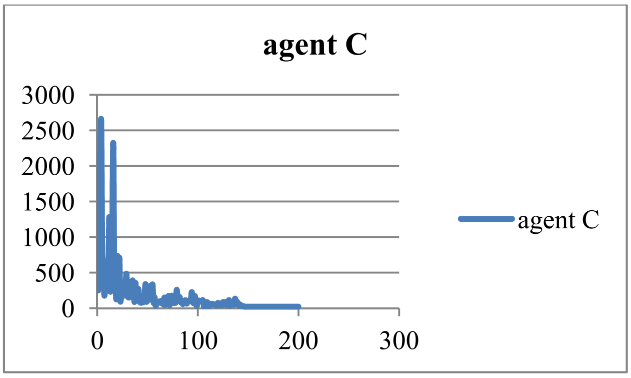

Fig. 4. Results by agent C (from 1st to 200th trial)

IV. Evaluation Experiment

A. Procedures

We have developed experimental agents based on sample agents whose source codes are included Robocup rescue simulator-package file.

We have prepared three different types of fire-brigade agents. They are below,

- Team A has fire-brigade agents which run randomly when they have no water to extinguish fire.
- Team B has fire-brigade agents which are implemented our proposed algorithm.
- Team C has fire-brigade agents which are implemented our proposed algorithm. Moreover when the fire-brigade agents have no water to extinguish fire, they always select only best path calculated by pheromone's concentrate.

Other types of agents are equal to sample agents in RoboCup rescue simulation simulator-package file. Each team has run 1500 steps, which are five trials of simulation term. We have employed a test map and it is also included RoboCup rescue simulation simulator-package file. A score of the map is 117.828 points at the start of simulation. Table II shows results.

TABLE II: Results of Evaluation Experiment

	Averages	Maximum	Minimum
agent A	969.46	8217	78
agent B	1788.42	5747	268
agent C	**470.92**	**2612**	**42**

Mean scores of these trials are shown in Table II. In this table, the mean score 18.763 point with Team B is better than other agents, slightly. Moreover the best score is 20.840 points with Team B in each trial. From this result, we have confirmed the effectiveness of our algorithm.

V. Consideration

Fig. 5 shows one example which the agents have a difficulty in actions for firefighting by fire-brigade agents and rescue civilian agents by ambulance agents. The reason in this situation is that some blockades on this road are disturbing and these agents are disenabling to go through.

Then there is a heavy traffic jam on the road. We can see red circle in Fig. 5.

Fig.5. A difficulty in actions for fire fighting and rescue

On the other hands, Fig. 6 shows another example of removing some blockades on the same road in another trial. Then agents are enable to go through and they success in their jobs. We can see blue circle in Fig. 6. However, in this research, our agents do not communicate with different types of agents and they run independently. In the near future, we have a plan to discuss the method of communication using ACO between the agents.

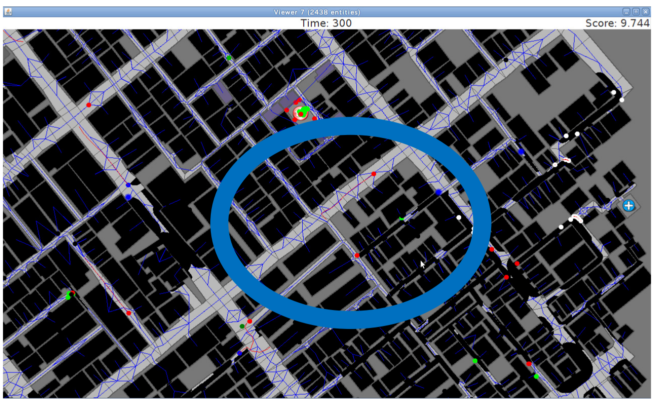

Fig.6. An example of removing blocks on a road for Fire Fighting and Rescue

VI. Conclusion

This paper has presented the proposal and reported the effectiveness of updating pheromone trails in agents with ACO. We have a plan to develop agents installed ACO algorithm on other type agents in RoboCup Rescue Simulation System.

Acknowledgment

We have developed our experimental system with the agent, which is based on source codes included in packages of simulator-package file. This work was supported by Grant-in-Aid for Scientific Research (C) (KAKENHI 23500196).

References

[1] M. Dorigo and T. Stützle, *Ant Colony Optimization*, The MIT Press, 2004.
[2] E. Bonabeau, M. Dorigo, and G.Theraulaz, *Swarm Intelligence from Naturak ti Artificial Systems*, Oxford University Press, 1999.
[3] E.Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, vol. 406, No. 6791, pp. 39-42, 6 July 2000.
[4] H. Hernandez, C. Blum, and J. H. Moore, "Ant colony optimization for energy-efficient broadcasting in AD-HOC networks," in *Proc. 6th International Conference, ANTS 2008*, Brussels, 2008. pp. 25-36.
[5] L. D'Acierno, B. Montella, and B. F. De Lucia, "A stochastic traffic assignt algorithm based ant colony optimisation," in *Proc. 5th International Conference, ANTS 2006*, Brussels, 2006, pp.25-36.
[6] P. Balaprakash, M.Birattari, T. Stützle, and M. Dorigo, "Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem," *Journal of Swarm Intelligence*, vol. 3, pp. 223-242, 2008.
[7] Robocup Rescue Simulation Project. [Online]. Available: http://sourceforge.net/projects/roborescue/
[8] Robocup Simulation League Spring Competition 2011 in the 73th National Convention of Information Processing Society Japan, Available: http://rc-oz.sourceforge.jp/pukiwiki/
[9] H. Sasaoka, "Approach for agent actions' acquiring method using ant colony optimization," in *Proc. of Joint 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems,* 2010.
[10] T. Stützle and H. H. Hoos, *MAX-MIN ant system. Future Generation Computer System*, 2000, 16(8), pp.889-914.
[11] *Nikkei Software (in Japanese)*, Nikkei BP Inc. 2011, vol.14, no.7.
[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction, A Bradford Book*, The MIT Press, 1998.

Hisayuki Sasaoka was born in Japan on January 13th, 1971. He has got Bachelor of Engineering in Hokkai-Gakuen University on 1995; got Master of Engineering in Hokkai-Gakuen University on 1997, Ph.D. in Hokkai-Gakuen University on 2000.

He worked in Asahikawa National College of Technology as an assistant professor from 2000 to 2002. He worked in Asahikawa National College of Technology as a lector in 2003. He works in Asahikawa National College of Technology as an associate professor from 2004. He was part-time lector in Hokkaido University (2001 and 2011), part-time lector in Hokkaido University of Education (2002) and part-time lector in Hokkai-Gakuen University (2011). His major fields of study are machine learning, swarm intelligence and human-computer interaction.

Dr. Sasaoka is a member of IEEE, IPSJ IECE and JSAI.