Cooperatively Transporting Unknown Objects using Mobile Agents

Ryo Takahashi[1], Munehiro Takimoto[1] and Yasushi Kambayashi[2]
[1]*Department of Information Sciences, Tokyo University of Science, 2641 Yamazaki, Noda 278-8510, Japan*
[2]*Department of Computer and Information Engineering, Nippon Institute of Technology,*
4-1 Gakuendai, Miyashiro-machi, Minamisaitama-gun 345-8501, Japan

Keywords: Mobile Agent, Multiple Robots, Ant Colony Optimization, Swarm Intelligence.

Abstract: This paper presents an algorithm for cooperatively transporting objects by multiple robots without any initial knowledge. The robots are connected by communication networks, and the controlling algorithm is based on the pheromone communication of social insects such as ants. Unlike traditional pheromone based cooperative transportation, we have implemented the pheromone as mobile software agents that control the mobile robots corresponding to the ants. The pheromone agent has the vector value pointing to its birth location inside, which is used to guide a robot to the birth location. Since the pheromone agent can diffuse with migrations between robots as well as a physical pheromone, it can attract other robots scattering in a work field to the birth location. Once the robot finds an object, it briefly pushes the object, measuring the degree of the inclination of the object. The robot generates a pheromone agent with the vector value to pusing point suitable for suppressing the inclination of the object. The process of the pushes and generations of pheromone agents enables the efficient transportation of the object. We have implemented a simulator based on our algorithm, and conducted experiments to demonstrate the feasibility of our approach.

1 INTRODUCTION

In the last decade, robot systems have made rapid progress not only in their behaviors but also in the way they are controlled. In particular, a control system based on multiple software agents can control robots efficiently (Takimoto et al., 2007). Multi-agent systems introduced modularity, reconfigurability and extensibility to control systems, which had been traditionally monolithic. It has made easier the development of control systems on distributed environments such as multi-robot systems.

On the other hand, excessive interactions among agents in the multi-agent system may cause problems in the multiple robot environments. In order to mitigate the problems of excessive communication, mobile agent methodologies have been developed for distributed environments. In a mobile agent system, each agent can actively migrate from one site to another site. Since a mobile agent can bring the necessary functionalities with it and perform its tasks autonomously, it can reduce the necessity for interaction with other sites. Mobile agent systems are especially useful in an intermittently connected ad hoc network environment. In the minimal case, a mobile agent requires that the connection is established only when it performs migration (Binder et al., 2001).

The model of our system is a set of cooperative multiple mobile agents executing tasks by controlling a pool of multiple robots as shown in Figure 1 (Kambayashi and Takimoto, 2005). The property of inter-robot movements of the mobile agents contributes to the flexible and efficient use of the robot resources. A mobile agent can migrate to the robot that is most conveniently located to a given task, e.g. closest robot to a physical object such as a soccer ball. Since the agent migration is much easier than the robot motion, the agent migration contributes to saving power consumption (Takimoto et al., 2007). Here, notice that any agents on a robot can be killed as soon as they finish their tasks. If the agent has a policy of choosing idle robots rather than busy ones in addition to the power-saving effect, it would result in more efficient use of robot resources.

We have proposed our model in the previous paper (Takimoto et al., 2007) and have also shown the effectiveness of saving power consumption and the efficiency of our system for searching targets (Nagata et al., 2009; Abe et al., 2011) and transporting them to a designated collection area (Shibuya et al., 2013). In this paper, we focus our attention on transportation of a large object that a single robot alone cannot

Takahashi R., Takimoto M. and Kambayashi Y..
Cooperatively Transporting Unknown Objects using Mobile Agents.
DOI: 10.5220/0004816400600068
In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence* (ICAART-2014), pages 60-68
ISBN: 978-989-758-016-1

Figure 1: A team of mobile robots are working under control of mobile agents.

move. In order to deal with such a large object, several robots have to cooperate to achieve the objective tasks, which seems to require too artificial behaviors for each robot. For the cooperatively solving of such problem, *swarm-based* approaches, which are based on the social insect metaphor, have been proposed. The swarm-based system, which consists of several robots with simple behaviors, can achieve complex tasks, just like ants that behave based on simple rules cooperatively transport a large prey. The swarm of simple robots makes a system more flexible and fault-tolerant, and contributes to suppressing costs of building a large complex system.

We have implemented the ants as actual mobile software agents that control the mobile robots. The ant agent migrates among robots to look for an available one. Once the ant agent finds the available robot, the ant agent physically drives the robot. We also implemented pheromone as mobile software agents, which attract many robots to the large object to convey through diffusing with the migrations. The pheromone agent has a vector value pointing to the birth point. The vector value is modified in order to always point to the birth point, each that the pheromone agent migrates to another robot. Once the pheromone agent migrates to the robot where an ant agent resides, it guides the ant agent to lead the robot to its birth point. Thus, the pheromone agents enable scattering free robots to efficiently attend to the transportation of the object. In our new approach, we take advantage of the pheromone agents not only to collect robots, but also to suppress the rotation of an object during the transportation. We assume that each robot has a simple sensor for checking the degree of the inclination of the object that it is pushing. When each robot slightly pushes the object, it generates pheromones at the suitable location to suppress the increase of the inclination of the object. As a result, the object can be linearly transported to a target area, even if the object has the form such as a stick. Notice that such an object could not be transported without suppressing the rolling.

The structure of the balance of this paper is as follows. In the second section, we describe the background. The third section describes our transport model based on pheromone agents. In the fourth section, we the numerical experiments using a simulator based on our algorithm. Finally, we conclude our discussions in the fifth section and present future research directions.

2 BACKGROUND

Making multiple robots cooperatively carry and push common objects has been intensively studied and yet to established the standard way. Many research projects have dealt this topic but few of them have demonstrated on physical multi-robot systems. One of the most demonstrated tasks involving cooperative transport is the pushing objects by multiple robot teams (Rus et al., 1995; Stilwell and Bay, 1993). This task is inherently easy to accomplish when comparing to carrying tasks, because a carrying task involves multiple robots' gripping a common object and navigating to a destination in a coordinated fashion (Khatib et al., 1996; Wang et al., 2000).

One of the most famous transportation problems is the box-pushing problem (Mataric et al., 1995). The problem is defined in (Gerkey and Mataric, 2002) and consists of cooperatively moving a box, which is relatively large when compared to the size of the multi-robots, from an initial position to a destination using robots that can only perform pushing movements.

On the other hand, algorithms that are inspired by behaviors of social insects such as ants to communicate to each other by an indirect communication called stigmergy are becoming popular. (Stilwell and Bay, 1993; Dorigo and Gambardella, 1996; Dorigo et al., 2006). Upon observing real ants' behaviors, Dorigo et al. found that ants exchanged information by laying down a trail of a chemical substance (called pheromone) that is followed by other ants. They adopted this ant strategy, known as ant colony optimization (ACO), to solve various optimization problems such as the traveling salesman problem (TSP) (Dorigo and Gambardella, 1996). Deneubourg has originally formulated the biology inspired behavioral algorithm that simulates the ant corps gathering and brood sorting behaviors (Deneubourg et al., 1991). Wang and Zhang proposed an ant inspired approach along this line of research that sorts objects with multiple robots (Wand and Zhang, 2004). Lumer has improved Deneubourg's model and proposed a new simulation model that is called Ant Colony Clustering (Lumer and Faiesta, 1994). His method could cluster similar objects into a few groups. Mizutani et al. has proposed to implement pheromones as mobile agents in ACO (Mizutani et al., 2010). In their system, ants are also mobile agents, that repeatedly migrate to robots to searching free robots corresponding to objects. Once the ant agent find a free robot, it drives the robot to a cluster of robots. The pheromone agent is generated by the ant agent when its driving robot reaches a cluster, and repeatedly migrates to robots to guide ant agents to drive robots to the cluster. The mobile agent based algorithm has been improved to serialize collected robots (Shintani et al., 2011b; Shintani et al., 2011a)

The applications of ants' behaviors to cooperative transportation has been proposed by Kube and Bonabeau (Kube and Bonabeau, 2000). It transports an object through the interplay of forces. The robots can recognize an object through the light emitted from the object, which works as the pheromone, but its effectiveness is restrictive. Fujisawa et al. have proposed the efficient cooperative transportation using ethanol as physical pheromone (Fujisawa et al., 2010).

3 TRANSPORT MODEL

We assume that each robot just affects the transported object through pushing it, and has to simultaneously push it together while other robots to move it. Also, since we assume that our multiple robots system is used for several tasks (Nagata et al., 2009), the robots available for the transportation are just ones not engaging other task.

In order to achieve the transportation over the robots shared with several tasks, we introduce two kinds of agents, which are called Ant Agents (AA) and Pheromone Agents (PA). The AA has a role for searching a robot with no other task, and driving a robot to the object to transport. The PA has a role for efficiently attracting the free robots to the object by guiding AAs on robots instead of directly driving them. In this section, we describe the robot's functionalities required in our system, and the details of two agents.

3.1 Robots

We assume mobile robots such as PIONEER 3-DX, which have two servo-motors with tires, one camera and sixteen sonic sensors (Nagata et al., 2009). The power is supplied by rechargeable battery. The PIONEER 3-DX has a servo-motor and sensor controller board that sends/receives data to/from a host computer on it through a USB cable. The camera is directly operated by the host. Each robot holds a laptop computer as a host, which is also used as a server for the migration of our control agents through the wireless LAN.

Also, in our transportation system, we assume that each robot has additional features as follows:

- it can push an object,
- it has the sensor for checking the degree of the inclination of the pushing object,
- it can always know the direction to which the object should be transported, and
- it knows the IP addresses of other robots by identifying them through the camera.

All functionalities for controlling the robot are introduced into each robot through migrations of mobile agents. Once an agent migrates to a host, the agent can communicate with other agents on the same host, so that the user can construct a larger system by migrations to the host.

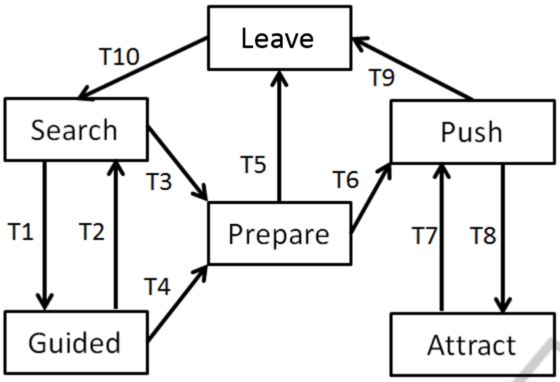

Figure 2: State transition diagram of Ant agent.

3.2 Ant Agent

AA has the list of IP addresses of all the robots in order to traverse them one by one and to find out a free robot. If it has exhausted all robots, it goes back to the home host for administration of the robots, and updates its IP address list in the following cases: 1) some new robots have been added, and 2) some robots have been broken, where notice that the addition or the removal of a robot is recorded on the home host. However, these are so rare that the home host is made to go through in most cases.

Once an AA reaches a free robot, the AA controls the robot along the state transition as shown in Figure 2.

First, the AA drives the robot to the object to transport through two kinds of manners. Initially, the AA makes the robot randomly walk to search the object as shown by state *Search*. Once PA migrates to the same robot, the AA drives the robot to the object along the guidance of the PA as shown by state *Guided*. After that, once the AA finds the object, it prepares for transporting the object as shown by state *Prepare*. Actual transportation is performed by repetition of briefly pushing the object as shown by state *Push* and attracting other robots to the suitable location for balancing the object as shown by state *Attract*. The transportation task of each robot is limited in the specific time, and therefore, after the time-out, the robot is released from the task as shown by state *Leave*. The details of each state is as follows:

State Search. In this state, AA drives the robot at random as shown in Figure 3. If the robot contacts with other robots, it temporarily stop there, and then take an action for avoidance. Once the robot bumps onto the object to transport, the state of the AA is changed to the state Prepare as shown by *T3* in Figure 2. On the other hand, if a PA migrates to the robot, the state is changed to the state Guided as shown by *T1*.

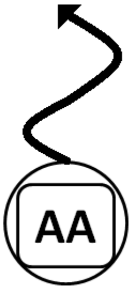

Figure 3: State Search of AA.

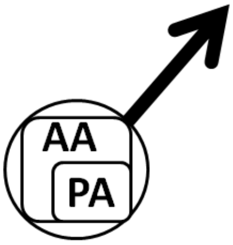

Figure 4: State Guided of AA.

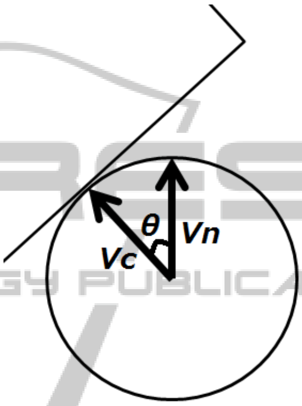

Figure 5: State Prepare of AA.

State Guided. In this state, AA drives the robot along the guidance of the PA that has migrated to the same robot as shown in Figure 4. The PA has the vector value pointing to the birth point of the PA, which is adjusted to always point to the birth point even if the robot moves or rotates. Since the PA can observe the rotations of wheels, the vector value is modified to neutralize the change caused by the rotation or the approach to the birth point of the robot. AA is guided for driving the robot in the direction of the vector value. Once the robot reaches the destination of the vector, the AA checks whether there are some objects around the robot or not. If some objects are found, the AA changes its state to the state Prepare for transporting the closest object as shown by *T4*; otherwise, the AA abandons the PA guiding it, and then changes its state to the state Search as shown by *T2*. At this time, the PA is killed.

State Prepare. Once an AA finds an object, the AA makes the robot approach to the object till contacting with it. After that, the AA measures the normal vector V_c against the surface of the object to know the

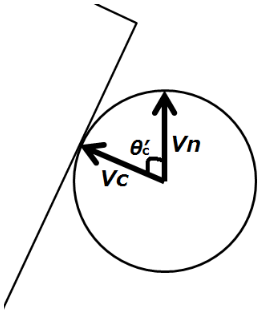

Figure 6: State Attract of AA.

degree of the inclination, as shown in Figure 5. Furthermore, since the AA always holds vector V_n to its nest to which the object is transported, it can calculate θ that is the angle between V_c and V_n. The agents including AA have the vector values in polar coordinate form. Therefore, using the angle θ_c of V_c and θ_n of V_n, θ is straightforwardly calculated as follows:

$$\theta = \theta_c - \theta_n \tag{1}$$

If the θ is more than $-\frac{\pi}{2}$ and less than $\frac{\pi}{2}$, the AA judges that robot has a suitable direction for efficient transportation, and changes its state to Push in order to start the transportation of the object as shown by $T6$. Otherwise, the AA changes its state to Leave in order to abandon the transportation, as shown by $T5$.

State Push. In this state, the AA just briefly pushes the object, and then changes its state to Attract in order to check the change of θ as shown by $T8$, and to attract another robot to the suitable location for balancing the object. Also, the AA releases the robot from the transportation task after specific time. The time-out functionality enables the finite robot resources to be reused for pushing another location, another object or another task, and it is achieved by changing its state to Leave, as shown by $T9$.

State Attract. Figure 6 shows the relation of a robot and an object after the robot pushed the object in the short period in the state Push, where V_c may have change. Therefore AA calculates θ_δ to check the angle θ'_c of new V_c as follows:

$$\theta_\delta = \theta'_c - \theta_c$$

If θ_δ is negative, it means that the object have rotated to the right. In this case, pushing the object at the neighbor location of the robot in the right hand side may suppress the rotation. Conversely, in the case where θ_δ is positive, pushing the object in the left hand side of the robot may suppress the rotation to the left. We call the location to push in order to balance the object *additional push point*. Once the AA knows the additional push point, it generates PA with the vector value pointing to the push point in order to attract another robot. After that, the PA repeatedly migrates some robots to find a free robot to attract. Notice that the AA generates no PA when θ_δ is correctly 0.

The AA generates some PAs in the predefined time, and then changes its state back to the Push, as shown by $T7$. The several transitions between the Push and the Attract achieves the transportation of the object in balance.

State Leave. In this state, the AA makes the robot turn to the back, and then, move it in the specific distance in order to release it from the transportation task.

3.3 Pheromone Agent (PA)

The PA has behaviors imitating the physical pheromone of ants, of which the main role is to attract the ants to preys or their nest. In a PA, the attraction is implemented as a guidance functionality based on the vector value to the birth point. As well as diffusion of physical pheromones, the PA searches the free robot to guide in two steps: *generation* and *propagation* of PA. First, in the generation step, the vector value to the birth point is set to the PA, and then, in the propagation step, it starts migrating to other robots. Finally, it is killed by the abandonment of AA. Also, when PA migrates to the robot on which another PA resides, they are composed in the *composition* step. In the remainder of the section, we describe the details of these steps.

Generation. The PA is generated by AA, of which the vector value is initialized to point to the neighbor location of the robot on which the AA resides. For example, when AA founds that the object rotates to the right in the state Attract, it generates a PA with the vector value as shown by V_r in Figure 7, where V_c is the normal vector against the object. Conversely, if the object rotates to the left, the vector is initialized as shown by V_l. The initial vector has the length corresponding with the size of a robot and 90 degrees against V_c.

Propagation. The purpose of the PA is to find a free robot, and then to lead it to the destination of the vector value. Therefore, as soon as it is generated or arrives at another robot, it checks around the current robot, so that if some robots are found, it generates its clone and makes the clone migrate to the closest one of them. The migration to the closest robot enables

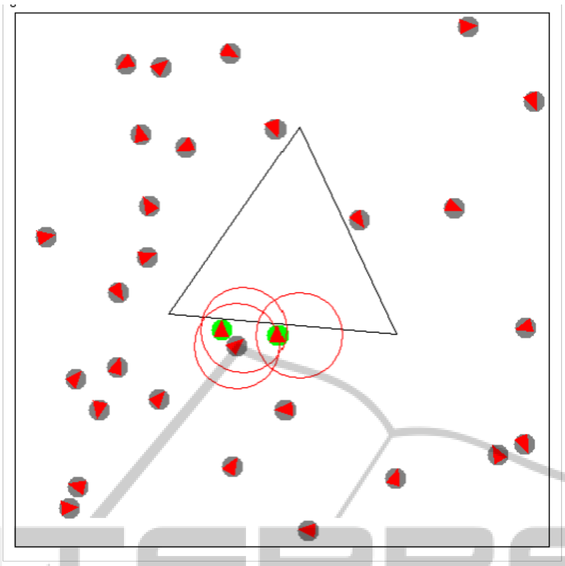

Figure 9: The snapshot image of the simulator.

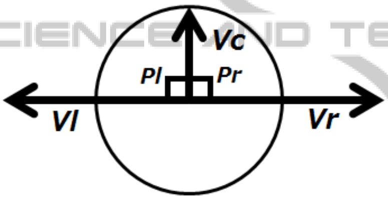

Figure 7: Generation of PA.

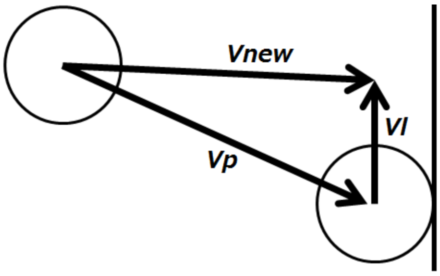

Figure 8: Propagation of PA.

the moving distance of the robot guided by the PA to be shorter. Notice here that in a migration process, the destination of the vector value of new PA after the migration has to be same as one of the old PA before the migration, which is achieved by synthesizing the old vector value V_{old} and the vector value V_p from the destination to source of the migration as follows:

$$V_{new} = V_{old} + Vp$$

Figure 8 shows the first migration of PA after its generation. In this case, V_{new} is set to $V_p + V_l$.

As mentioned above, a PA generates its clone, which migrates to another robot. The number of the generation of a clone is limited to five. A clone is generated one at a time. Also, each PA is finally abandoned by the AA in its state Leave, and therefore, the PA does not continue giving the old information to the system.

Composition. Once the PA migrates to the robot on which another PA resides, they have to be composed. In the case of physical pheromone, The effectiveness of the attraction is strengthened by the composition, whereas in PAs, either of them is alternatively selected. All the vector values of them are checked, so that the PA with shortest vector is selected.

This composition strategy enables decreasing the total distance that all robots move, so that it contributes to suppressing energy consumption of the entire robot system. In addition, the strategy also contributes to keeping the degree of the current dispersion, because it is profitable for robots to uniformly scatter considering the case where several objects are simultaneously transported.

4 EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of our transportation method,we have implemented a simulator. Figure 9 shows the output image of the simulator, through which we can observe the behaviors of agents and robots. Numerical statistic data are recorded apart from it.

Each small circle shows a robot, the moving direction of which is shown by a triangle inside it. Each robot is assumed to move from a grid to another grid

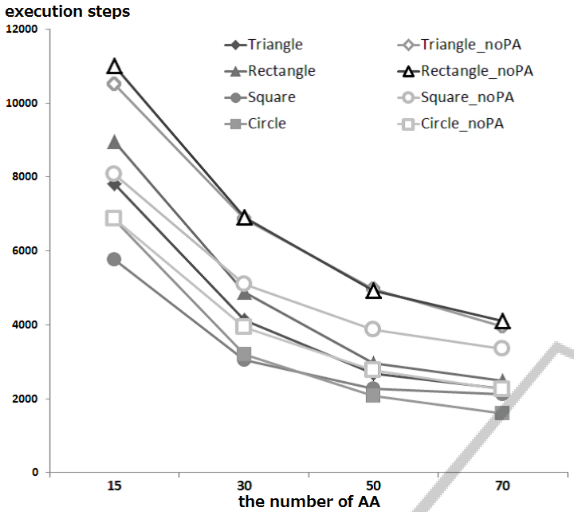

Figure 10: Transportation time.

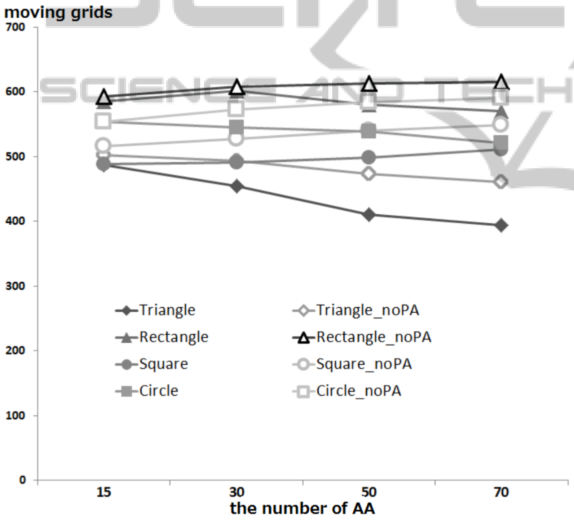

Figure 11: Transportation distance.

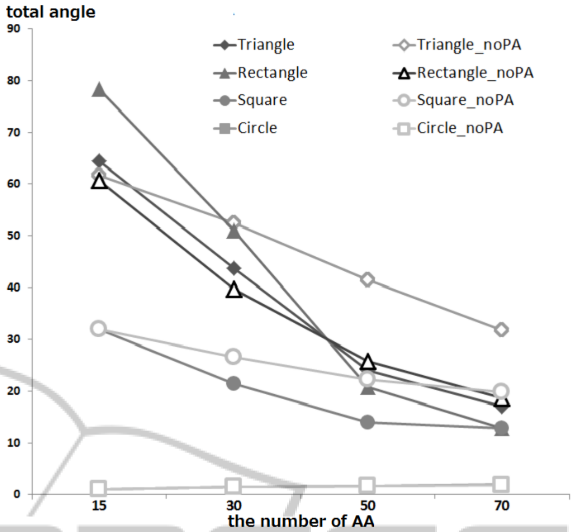

Figure 12: Rotation angle.

in 500×500 grids square field. Also, the field is surrounded by a wall, and therefore, robots cannot go out the field. A big red circle with a robot at its center shows the view range of the camera equipped on the robot, and it is also the range where AA or PA can migrate. In the figure, the object to transport is shown by an equilateral triangle with sides 240 grids long.

In order to confirm the accuracy and efficiency of our method, we have conducted eight experiments through applying two kinds of strategies, which are ones with PA and with no PA, to four kinds of objects with the shapes of triangle, square, rectangle, and circle. For each experiment, we have measured the total time and the total distance for transporting an object , and the total angle where the object rotates.

Figure 10 through 12 show the results of the transportation time, the transportation distance and the total angle of the rotation of the object, respectively. In all figures, the horizontal axes indicate the number of AA. The vertical axis of Figure 10 indicates the number of the execution steps of the simulator as the transportation time. One of Figure 11 shows the number of moving grids as the transportation distance. One of Figure 12 is the total angle in the transportation. In both figures, the thin line shows the results without PA, and the dark lines shows the result with PA.

Figure 10 shows the transportation time decreases as AAs increases. That just means that the more robots attend the transportation task, the more efficiently the task is performed, though the slop of the decrease becomes close to a flat around 70 AAs. Comparing the results with no PA and one with PA, the transportation with PA is found to be much more efficient. Besides, as observed from the fact that the efficiency of 30 AAs with PA is same as one of 70 AAs with no PA, the result shows that our method has effectiveness for suppressing the required number of robots.

The result of the transportation distance shows that the distance of the strategy with PA is less than with no PA for all object shapes, and furthermore, the distance of the no PA strategy increases as AAs increase except for just the square, while the PA strategy decreases for all the shapes, as shown in Figure 11.

That means PA contributes to suppressing energy consumption in the transportation task.

The result of Figure 12 shows that the rotation angle decreases as AA increases for both strategies. That is because the object with the symmetry shape naturally balances when a lot of robots uniformly contacts with one side. Even if such the fact is considered, the effectiveness suppressing the rotation in proportion to the number of AAs is remarkable. However, the degradation around 15 AAs is one of issues to solve. The degradation is marked for the rectangle, which is likely to increase for objects with a shape such as a stick. It is derived from the generation manner of PA. Since AA generates PA with the vector value to the neighbor location of it, the impact of each attraction is not so strong in the case of fewer AAs. In order to solve the issue, it would be effective to determine the destination of PA's vector value in proportion to the degree of the inclination occurred in pushing the object.

5 CONCLUSIONS

We have proposed an effective transportation method for multiple robots using mobile agents that imitate social insects. The transportation method enables robots scattering in a work filed to cooperate for pushing an object in balance, which suppresses the rotation of the object, so that it contributes to efficient transportation and suppressing energy consumption. In order to show the effectiveness of our transportation method, we have implemented a simulator, on which we have conducted some experiments. As a result, in most cases, our method shows remarkable effectiveness for the transportation task.

On the other hand, in some case where the number of Ant agents is small, the effectiveness of our method is restrictive. In order to solve this problem, the destination for Pheromone agents to guide would need to be determined depending on the degree of the rotation.

REFERENCES

Abe, T., Takimoto, M., and Kambayashi, Y. (2011). Searching targets using mobile agents in a large scale multi-robot environment. In *KES-AMSTA*, volume 6682 of *LNAI*, pages 211–220.

Binder, W., Hulaas, J. G., and Villazon, A. (2001). Portable resource control in the j-seal2 mobile agent system. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, pages 222–223. ACM.

Deneubourg, J., Goss, S., Franks, N. R., Sendova-Franks, A. B., Detrain, C., and Chreien, L. (1991). The dynamics of collective sorting: Robot-like ant and ant-like robot. In *Proceedings of the First Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–363. MIT Press.

Dorigo, M., Birattari, M., and T. Stützle (2006). Ant colony optimization–artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1(4):28–39.

Dorigo, M. and Gambardella, L. M. (1996). Ant colony system: a cooperative learning approach to the traveling salesman. *IEEE Transaction on Evolutionary Computation*, 1(1):53–66.

Fujisawa, R., Imamura, H., and Matsuno, F. (2010). Cooperative transportation by swarm robots using pheromone communication. In *Distributed Autonomous Robotic Systems - The 10th International Symposium, DARS 2010*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 559–570. Springer.

Gerkey, B. P. and Mataric, M. J. (2002). Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation 1*, pages 464–469.

Kambayashi, Y. and Takimoto, M. (2005). Higher-order mobile agents for controlling intelligent robots. *International Journal of Intelligent Information Technologies (IJIIT)*, 1(2):28–42.

Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R., and Casal, A. (1996). Vehicle/arm coordination and mobile manipulator decentralized cooperation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 546–553.

Kube, C. R. and Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101.

Lumer, E. D. and Faiesta, B. (1994). Diversity and adaptation in populations of clustering ants, from animals to animats 3. In *Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior*, pages 501–508. MIT Press.

Mataric, M. J., Nilsson, M., and Simsarian, K. T. (1995). Cooperative multi-robot box-pushing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 3*, pages 556–561.

Mizutani, M., Takimoto, M., and Kambayashi, Y. (2010). Ant colony clustering using mobile agents as ants and pheromone. In *Proceedings of the Second International Conference on Applications of Intelligent Systems*, pages 435–444. Lecture Notes in Computer Science 5990, Springer-Verlag.

Nagata, T., Takimoto, M., and Kambayashi, Y. (2009). Suppressing the total costs of executing tasks using mobile agents. In *Proceedings of Hawaii International Conference on System Sciences 42 CD-ROM*.

Rus, D., Donald, B., and Jennings, J. (1995). Moving furniture with teams of autonomous robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 235–242.

Shibuya, R., Takimoto, M., and Kambayashi, Y. (2013). Suppressing energy consumption of transportation robots using mobile agents. In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART 2013)*, pages 219–224. SciTePress.

Shintani, M., Lee, S., Takimoto, M., and Kambayashi, Y. (2011a). A serialization algorithm for mobile robots using mobile agents with distributed ant colony clustering. In *Knowledge-Based and Intelligent Information and Engineering Systems - 15th International Conference, KES 2011, Proceedings, Part I*, volume 6881 of *Lecture Notes in Computer Science*, pages 260–270. Springer.

Shintani, M., Lee, S., Takimoto, M., and Kambayashi, Y. (2011b). Synthesizing pheromone agents for serialization in the distributed ant colony clustering. In *ECTA and FCTA 2011 - Proceedings of the International Conference on Evolutionary Computation Theory and Applications and the Proceedings of the International Conference on Fuzzy Computation Theory and Applications [parts of the International Joint Conference on Computational Intelligence IJCCI 2011]*, pages 220–226. SciTePress.

Stilwell, D. J. and Bay, J. S. (1993). Toward the development of a material transport system using swarms of ant-like robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 766–771.

Takimoto, M., Mizuno, M., Kurio, M., and Kambayashi, Y. (2007). Saving energy consumption of multi-robots using higher-order mobile agents. In *KES-AMSTA*, volume 4496 of *LNAI*, pages 549–558.

Wand, T. and Zhang, H. (2004). Collective sorting with multi-robot. In *Proceedings of the First IEEE International Conference on Robotics and Biomimetics*, pages 716–720.

Wang, Z. D., Kimura, Y., Takahashi, T., and Nakano, E. (2000). A control method of a multiple nonholonomic robot system for cooperative object transportation. In *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems on Distributed Autonomous Robotic Systems 4*, pages 447–456.