

Location of Intelligent Carts Using RFID

Yasushi Kambayashi and Munehiro Takimoto
*Nippon Institute of Technology & Tokyo University of Science
Japan*

1. Introduction

This chapter addresses optimizing distributed robotic control of systems using an example of an intelligent cart system designed to be used in common airports. This framework provides novel control methods using mobile software agents. In airport terminals, luggage carts used by traveler are taken from a depot but are left after use at arbitrary points. It would be desirable that carts be able to draw themselves together automatically after being used so that manual collection becomes less laborious. In order to avoid excessive energy consumption by the carts, we employ mobile software agents and RFID (Radio Frequency Identification) tags to identify the location of carts scattered in a field and then cause them to autonomously determine their moving behavior using a clustering method based on the ant colony optimization (ACO) algorithm.

When we pass through terminals of an airport, we often see carts scattered in the walkway and employees manually collecting them one by one. It is a laborious task and not a fascinating job. It would be much easier if carts were roughly gathered in any way before the laborers begin to collect them. Multiple robot systems have made rapid progress in various fields, and the core technologies of multiple robot systems are now easily available (Kambayashi & Takimoto, 2005). Employing such technologies, it is possible to give each cart minimum intelligence, making each cart an autonomous robot. We realize that for such a system cost is a significant issue and we address one of those costs, the power source. A big, powerful battery is heavy and expensive; therefore such intelligent cart systems with small batteries are desirable to save energy (Takimoto, Mizuno, Kurio & Kambayashi, 2007; Nagata, Takimoto & Kambayashi, 2009; Oikawa, Mizutani, Takimoto & Kambayashi, 2010; Abe, Takimoto & Kambayashi, 2011).

Travelers pick up carts at designated points and leave them in arbitrary places. It is desirable that intelligent carts (intelligent robots) draw themselves together automatically. A simple implementation would be to give each cart a designated assembly point to which it automatically returns when free. That solution is easy to implement, but some carts would have to travel a long way back to their own assembly point, even though they are located close to other assembly points. That strategy consumes unnecessary energy.

To improve efficiency, we employ mobile software agents to locate carts scattered in a field, e.g. an airport, and enable them to determine their moving behavior autonomously using a clustering algorithm based on ant colony optimization (ACO). ACO is a swarm intelligence-based method and a multi-agent system that exploits artificial stigmergy for the solution of combinatorial optimization problems. Preliminary experiments yield a favorable result. Ant

colony clustering (ACC) is an ACO specialized for clustering objects. The idea is inspired by the collective behaviors of ants, used by Deneubourg to formulate an algorithm that simulates the ant corps gathering and brood sorting behaviors (Deneubourg, Goss, Franks, Sendova-Franks, Detrain & Chretien, 1991).

We have studied the base idea for controlling mobile multiple robots connected by communication networks (Kambayashi, Tsujimura, Yamachi, Takimoto, & Yamamoto, 2010; Kambayashi & Takimoto, 2005). Our framework provides novel methods to control coordinated systems using mobile agents. Instead of physical movement of multiple robots, mobile software agents can migrate from one robot to another so that they can minimize energy consumption for aggregation. In this chapter, we describe the details of implementation of the multi-robot system using multiple mobile agents and static agents that implement ACO as well as the location system using RFID. The combination of the mobile agents augmented by ACO and mobile multiple robots with RFID opens a new horizon of efficient use of mobile robot resources. We report here our experimental observations of our robot cart system.

Quasi-optimal cart collection is achieved in three phases. The first phase collects the positions of the carts. One mobile agent issued from the host computer visits scattered carts one by one and collects the positions of them. The precise coordinates and orientation of each robot are determined by interrogating RFID tags under the floor carpet. Upon the return of the position collecting agent, the second phase begins wherein another agent, the simulation agent, performs the ACC algorithm and produces the quasi-optimal gathering positions for the carts. The simulation agent produces not only the assembly positions of the carts but also the moving routes and waiting timings for avoiding collisions; i.e. the entire behaviors of all the intelligent carts. The simulation agent is a static agent that resides in the host computer. In the third phase, a number of mobile agents, the driving agents are issued from the host computer. Each driving agent migrates to a designated cart, and drives the cart to the assigned quasi-optimal position that was calculated in the second phase.

The behaviors of each cart are determined by the simulation agent. It is influenced, but not determined, by the initial positions and the orientations of scattered carts, and is dynamically re-calculated as the configuration of the field (positions of the carts in the field) changes. Instead of implementing ACC with actual carts, one static simulation agent performs the ACC computation, and then mobile agents distribute the sets of produced driving instructions. Therefore our method eliminates unnecessary physical movement and provides energy savings.

The structure of the balance of this paper is as follows. In the second section, we review the history of research in this area. In the third section, we describe the controlling agent system that performs the arrangement of the intelligent carts. The agent system consists of several static and mobile agents. The static agents interact with the users and compute the ACC algorithm and the simulation of the intelligent carts' behaviors. The other mobile agents gather the initial positions of the robots and drive the carts to the assembly positions. The fourth section describes how each robot determines its coordinates and orientation by sensing RFID tags under the floor carpet. The fifth section describes the ACC algorithm we have employed to calculate the quasi-optimal assembly positions and moving instructions for each cart. Finally, in the sixth section, we summarize the work and discuss future research directions.

2. Background

Kambayashi and Takimoto have proposed a framework for controlling intelligent multiple robots using higher-order mobile agents (Kambayashi & Takimoto, 2005). The framework helps users to construct intelligent robot control software using migration of mobile agents. Since the migrating agents are of higher order, the control software can be hierarchically assembled while they are running. Dynamic extension of control software by the migration of mobile agents enables the controlling agent to begin with relatively simple base control software, and to add functionalities one by one as it learns the working environment. Thus we do not have to make the intelligent robot smart from the beginning or make the robot learn by itself. The controlling agent can send intelligence later through new agents. Even though the dynamic extension of the robot control software using the higher order mobile agents is extremely useful, such a higher order property is not necessary in our setting. We have employed a simple, non-higher-order mobile agent system for our intelligent cart control system. We previously implemented a team of cooperative search robots to show the effectiveness of such a framework, and demonstrated that that framework contributes to energy savings for a task achieved by multiple robots (Takimoto, Mizuno, Kurio & Kambayashi, 2007; Nagata, Takimoto & Kambayashi, 2009; Oikawa, Mizutani, Takimoto & Kambayashi, 2010; Abe, Takimoto & Kambayashi, 2011). Our simple agent system should achieve similar performance.

Deneuburg formulated the biologically inspired behavioral algorithm that simulates the ant corps gathering and brood sorting behaviors (Deneuburg, Goss, Franks, Sendova-Franks, Detrain, & Chretien, 1991). His algorithm captured many features of the ant sorting behaviors. His design consists of ants picking up and putting down objects in a random manner. He further conjectured that robot team design could be inspired from the ant corps gathering and brood sorting behaviors (Deneuburg, Goss, Franks, Sendova-Franks, Detrain, & Chretien, 1991). Wang and Zhang proposed an ant inspired approach along this line of research that sorts objects with multiple robots (Wang & Zhang, 2004).

Lumer improved Deneuburg's model and proposed a new simulation model that was called Ant Colony Clustering (Lumer, & Faieta, 1994). His method could cluster similar object into a few groups. He presented a formula that measures the similarity between two data objects and designed an algorithm for data clustering. Chen et al have further improved Lumer's model and proposed Ants Sleeping Model (Chen, Xu & Chen, 2004). The artificial ants in Deneuburg's model and Lumer's model have considerable amount of random idle moves before they pick up or put down objects, and considerable amount of repetitions occur during the random idle moves. In Chen's ASM model, an ant has two states: active state and sleeping state. When the artificial ant locates a comfortable and secure position, it has a higher probability of being in the sleeping state. Based on ASM, Chen has proposed an Adaptive Artificial Ants Clustering Algorithm that achieves better clustering quality with less computational cost.

Algorithms inspired by behaviors of social insects such as ants that communicate with each other by the stigmergy are becoming popular (Dorigo & Gambardella, 1996). Upon observing real ants' behaviors, Dorigo et al found that ants exchanged information by laying down a trail of a chemical substance (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 & Gambardella, 1996). Our ACC algorithm employs pheromone, instead of using Euclidian distance to evaluate its performance.

Retrieving accurate location information of an object is a key to enabling a robot to perform any task. Recently, to solve the localization problem in indoor environment, various RF-based techniques have been developed (Hightower & Borriello, 2001; Finkenzeller, 2003). One of the often used methods is to use reference stations with well-known locations, where the transponder position is reported by nearby stations (Werb & Lanzl, 1998). A criticism of this approach is that the number of stations and the interval between the stations affects the accuracy (Kim & Chong, 2007). In order to deal with the problem of object localization, Kim and Chong have developed a location sensing system incorporating 315 MHz active RFID devices. Their system enables a robot to gather information and understand the context of the environment using the RFID transponder's location as well as stored information. They use a directional antenna to determine the arrival direction of the RF signal. Even though their system successfully provides guidance to mobile robots to a stationary target, the RFID transponder fails to provide precise coordinate of a robot. Also the system is costly because of the use of active RFID devices.

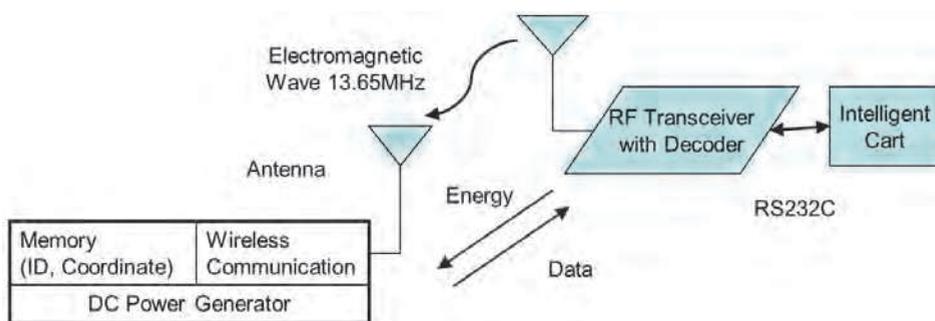


Fig. 1. Architecture of a passive RFID tag

In order to get precise coordinates for the robot, we utilized another well-known and well-analyzed approach. We embedded a very large number of passive RFID devices that emit very weak signals under the floor carpet of our test environment so that only robots extremely near the RFID tag can receive the positional signal. Since 13.56 MHz passive RFID devices are very inexpensive, we believe our rather primitive method is practical enough to retrieve the location of a robotic intelligent cart. Fig. 1 shows the architecture of a passive RFID tag.

3. Controlling agents

Our system model consists of carts and a few kinds of static and mobile software agents. All the controls for the intelligent carts as well as ACC computation performed in the host computer are achieved through the static and mobile agents. They are: 1) user interface agent (UIA), 2) operation agents (OA), 3) position collecting agent (PCA), 4) clustering simulation agent (CSA), and 5) driving agents (DA). All the software agents except UIA and CSA are mobile agents. A mobile agent traverses carts scattered in the field one by one to collect their coordinates. After receiving the moving instructions computed by a static agent, many mobile agents migrate to the carts and drive them to the assembly positions. Fig. 2 shows the interactions of the cooperative agents to control an intelligent cart.

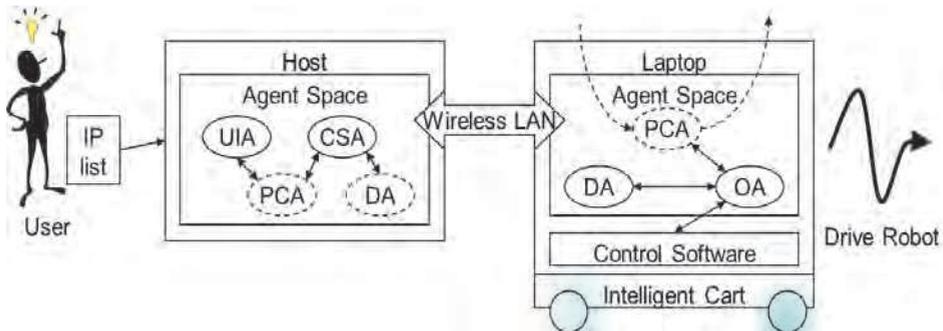


Fig. 2. Cooperative agents to control an intelligent cart

The followings are details of each agent:

1. User Interface Agent (UIA): The user interface agent (UIA) is a static agent that resides on the host computer and interacts with the user. It is expected to coordinate the entire agent system. When the user creates this agent with a list of IP addresses of the mobile robots, UIA creates PCA and passes the list to it.
2. Operation Agent (OA): Each intelligent cart has at least one operation agent (OA). It has the task that the cart on which it resides is supposed to perform. Each intelligent cart has its own OA. Currently all operation agents (OA) have a function for collision avoidance and a function to sense RFID tags embedded in the floor carpet to detect its precise coordinates in the field.
3. Position Collecting Agent (PCA): A distinct agent called position collecting agent (PCA) traverses intelligent carts scattered in the field one by one and collects their coordinates. PCA is created and dispatched by UIA. Upon returning to the host computer, it hands the collected coordinates to the clustering simulation agent (CSA) for ACC.
4. Clustering Simulation Agent (CSA): The host computer houses the static clustering simulation agent (CSA). This agent actually performs the ACC algorithm by using the coordinates collected by PCA as the initial positions, and produces the quasi-optimal assembly positions of the carts, and then performs yet another simulation to produce instructions each cart follows to reach its assigned goal position. Upon terminating the simulation and producing the procedures for all the intelligent carts, CSA creates a number of driving agents (DA).
5. Driving Agent (DA): The quasi-optimal arrangement coordinates, as well as procedures to reach them, produced by the CSA are delivered by driving agents (DA). One driving agent is created for each intelligent cart, and it contains the set of procedures for the cart. The DA drives its intelligent cart to the designated assembly position. DA is the intelligent part of the intelligent cart.

OA detects the current coordinate of the intelligent cart on which it resides. The coordinate information is retrieved from the RFID tag under that cart. Each cart has its own IP address and UIA hands in the list of the IP addresses to PCA. First, PCA migrates to an arbitrary cart and starts hopping between them one by one. It communicates locally with OA, and writes the coordinates of the cart into its own local data area. When PCA gets all the coordinates of the robots, it returns to host computer. Upon returning to the host computer, PCA creates CSA and hands in the coordinate data to CSA which computes the ACC algorithm.



Fig. 3. RFID tags under a carpet tile

Each intelligent cart employs RFID (Radio Frequency Identification) to get precise coordinates. We set RFID tags in a regular grid shape under the floor carpet tiles (Fig. 3). The tags we chose have a small range so that the position-collecting agent can obtain fairly precise coordinates from the tag. Also, the cart itself has a basic collision avoidance mechanism using infrared sensors.

CSA is the other static agent and its sole role is ACC computation. When CSA receives the coordinate data of all the carts, it translates the coordinates into coordinates for simulation, and then performs the clustering. When CSA finishes the computation and produces a set of assembly positions, it then creates the set of procedures for intelligent cart movements by simulating the behaviors of all the carts. The simulator produces one set of moving procedures for each cart including not only the course it takes but also timing to avoid collisions so that each cart can know exactly what it should do just for a few steps. The reason why the procedures are for a few steps is that it is impractical to forecast the entire behaviors of all the carts from the initial positions to the final destinations. Therefore the system repeats the cycle of collecting positions, performing ACC, and driving robots for short distance for several times to accomplish the clustering task.

Then CSA creates enough DA so that there is one for each cart. The DA convey the sets of procedures to the intelligent carts. Each DA receives its destination IP address from PCA, and the set of procedures for the destination cart, and then migrates to the destination cart. Each DA has a set of driving procedures that drives its assigned cart to the tentative destination, while it avoids collision. OA has the basic collision detection and avoidance procedures, and DA has task-specific collision avoidance procedures.

We have implemented the entire multiple mobile software agents for mobile robot control using Agent Space (Satoh, 1999). Agent Space is a library for constructing mobile agents developed by Satoh. By using its library, the user can implement a mobile agent environment with the Java language.

4. Determining the positions and directions

In order to get the coordinate of each robot, each robot employs an RFID reader/writer module ARW13T-RS01 by Niigata Seimitsu Co. Ltd. (Niigata Seimitsu, 2011). The module is based on ISO 15693 and uses carrier frequency 13.56MHz. Fig. 4 shows the effective scope of

sensing. As the diagram shows the effective scope of the RFID module is quite narrow so that the robot can sense the RFID tag exactly under its body.

For driving carts on a quasi-optimal route, one needs not only the precise coordinates of each robot but also the direction each cart faces. In order to determine the direction that it is facing, each cart moves straight ahead in the direction it is currently facing and obtains two positions (coordinates) from RFID tags under the carpet tiles. Determining current orientation is important because there is a high cost for making a cart rotate through a large angle. It is desirable that each cart be assigned rather simple forward movements rather than complex movement with several direction-changes when there are obstacles to avoid. Therefore whenever OA is awake, it performs the two actions, i.e. obtaining the current position and calculating the current direction.

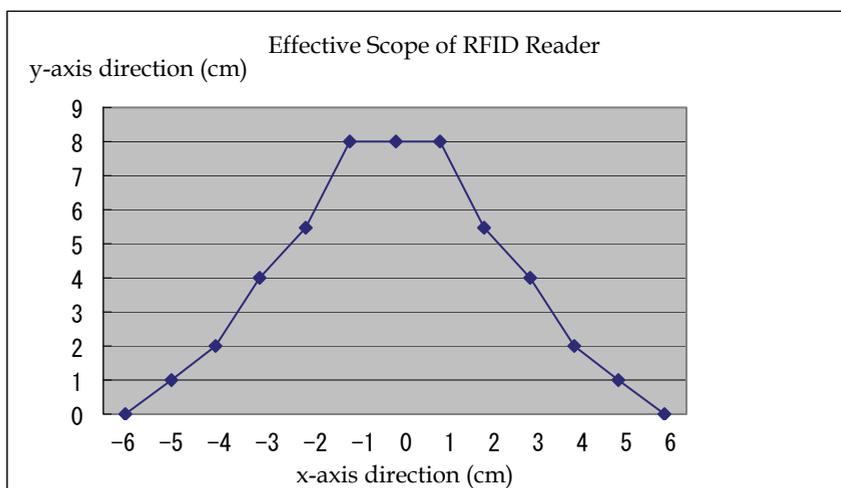


Fig. 4. Effective scope of the RFID reader/writer module

Since each carpet tile has nine RFID tags, as shown in Fig. 3, the cart is supposed to obtain the current position as soon as OA gets the sensor data from the RFID module. If OA can not obtain the position data, it means the RFID module can not sense a RFID tag, OA moves the cart forward until the RFID module senses a RFID tag. Once OA obtains the current position, it drives the cart a short distance until the RFID module detects the second RFID tag. Upon obtaining two positions, it is a simple computation to determine the direction to which the cart is moving, as shown in Fig. 5.

In a real situation, we may find the cart close to the wall and facing it. Then we can not use the simple method just described above since the cart would move forward and collide with the wall. In order to accommodate such situations, we gave RFID tags near the wall a special signal to the cart that tells it that it is at the end of the field (near the wall) so that the cart that senses the signal can rotate to the opposite direction. This is only required in our experimental implementation because the current collision detection mechanism does not otherwise work well enough to avoid collision with the wall. When the cart finds wall while moving to obtain the direction, it arbitrarily changes the direction, and starts obtaining two positions again.

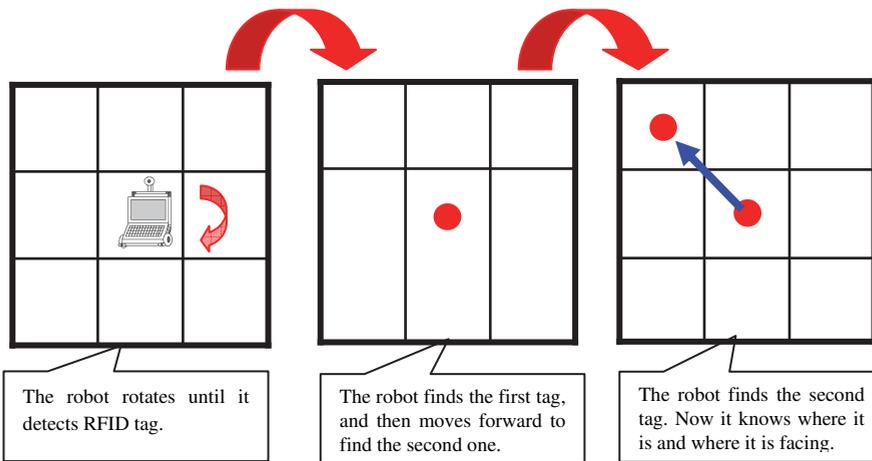


Fig. 5. Determining orientation from two RFID tag positions

5. Ant colony clustering

PCA collects the positions of the carts obtained by OA on the carts, and passes them to CSA that performs the ACC algorithm to calculate assembly points to which the carts are directed. In traditional ACO systems, artificial ants leave pheromone signals so that other artificial ants can trace the same path (Deneuburg, Goss, Franks, Sendova-Franks, Detrain, & Chretien, 1991). In our ACC system, however, we made objects have a pheromone such that more objects are clustered into a place where strong pheromone is sensed. Our randomly walking artificial ants have high probability to pick up an object with weak pheromone, and to put the object where they sense strong pheromone. They are not supposed to walk long distances so the artificial ants tend to pick up a solitary object and produce many small clusters of objects. When a few clusters are generated, they tend to grow.

Since the purpose of traditional ACC is clustering or grouping objects into several different classes based on selected properties, it is desirable that the generated chunks of clusters grow into one big cluster such that each group has distinct characteristics. In our system, however, we want to produce several roughly clustered groups of the same type, and move each cart a minimal distance. (We assume we have one kind of cart, and that we do not want carts to move long distances.) Therefore our artificial ants have the following behavioral rules:

1. An artificial ant's basic behavior is random walk, and when it finds an isolated object, it picks up the object.
2. When the artificial ant finds a cluster with certain number of objects, it tends to avoid picking up an object from the cluster. This number can be updated later.
3. When the artificial ant with an object finds a cluster, it put down the object so that the object is adjunct to one of the objects in the cluster.
4. If the artificial ant cannot find any cluster with certain strength of pheromone, it just continues a random walk.

By the restrictions defined in the above rules, the artificial ants tend not to convey objects long distances, and produce many small heaps of objects at the first stage. In order to implement the first feature, the system locks objects with certain number of adjoining objects, and no artificial ant can pick up such a locked object. The number for locking will be updated later so that artificial ants can bring previously locked objects in order to create larger clusters. When the initially scattered objects are clustered into a small number of heaps, then the number of objects that makes objects locked is updated, and the further activities of the artificial ants re-start to produce smaller number of clusters. We have implemented a simulator to evaluate our ACO algorithm, and succeeded in producing not only the quasi-optimal gathering positions but also the precise behaviors of all the carts to reach the calculated gathering positions. Fig. 6 shows the behavior of an artificial ant. The details of the ACO algorithm we have designed and implemented are reported in (Kambayashi, Tsujimura, Yamachi, Takimoto & Yamamoto, 2010).

Even though the ACC algorithm achieves some quasi-optimal clustering, it is hard to have confidence that we can make all the carts autonomously move to the gathering positions so that they form the quasi-optimal clusters. As the carts move toward the assigned positions, the configuration of the entire cart system changes and also each cart may perform unexpected behaviors, such as slipping tires over-stirring as well as under-stirring. Therefore we need to dynamically re-perform ACC to re-calculate the new goal position for each cart based on the current position after all the carts move independently. On the other hand, we found from the preliminary experiments that excessive re-computation of ACC might produce one large cluster, and that was not what we desired. In this section, we discuss how frequently we perform ACC to guide carts so that they form quasi-optimal clusters. In order to give each cart not only the goal position but also the procedure to reach it, we have implemented the simulator to execute a simulation of all the carts so that we can assign one precise behavior for each cart as well as perform ACC algorithm, and confirmed that it is feasible to produce the behavioral instructions for all the carts so that ultimately they can reach the quasi-optimal positions.

Upon receiving the positions of all the carts, the CSA immediately starts ACC simulation and produces the quasi-optimal clusters. Fig. 7b show the calculated clusters the CSA proposed from the initial cart positions in Fig. 7a. At this moment, none of the carts know how to behave, i.e. which direction and how far each should go, because each cart has not get been assigned its goal position. Upon obtaining the goal clusters, CSA performs yet another simulation. At this time, the simulation imitates the behaviors of all the carts from the initial positions to the tentative goal positions. This simulation produces the moving routes and wait timing for avoiding collisions. Fig. 7c shows one of the best samples of the simulated clusters that can be actually formed by the moving carts. Surprisingly the simulated clusters are quite similar to the calculated clusters by the ACC.

This second simulation produces the precise coordinate and wait timing for each cart, thus generating a set of moving procedures for each cart. One procedure consists of not only a route for the cart but also the timing when the cart stops and how long it waits to avoid collision against other colleague carts.

Upon constructing all the instructions for all the carts, a number of DAs are created to convey the procedures. Each DA drives corresponding cart to the simulated gathering

positions. At a certain time, all the carts move toward the assigned positions through the instructions given by DA. After that period, the configuration of the field changes, then we need to re-perform the ACC again so that it reflects the current configuration (positions of all the carts).

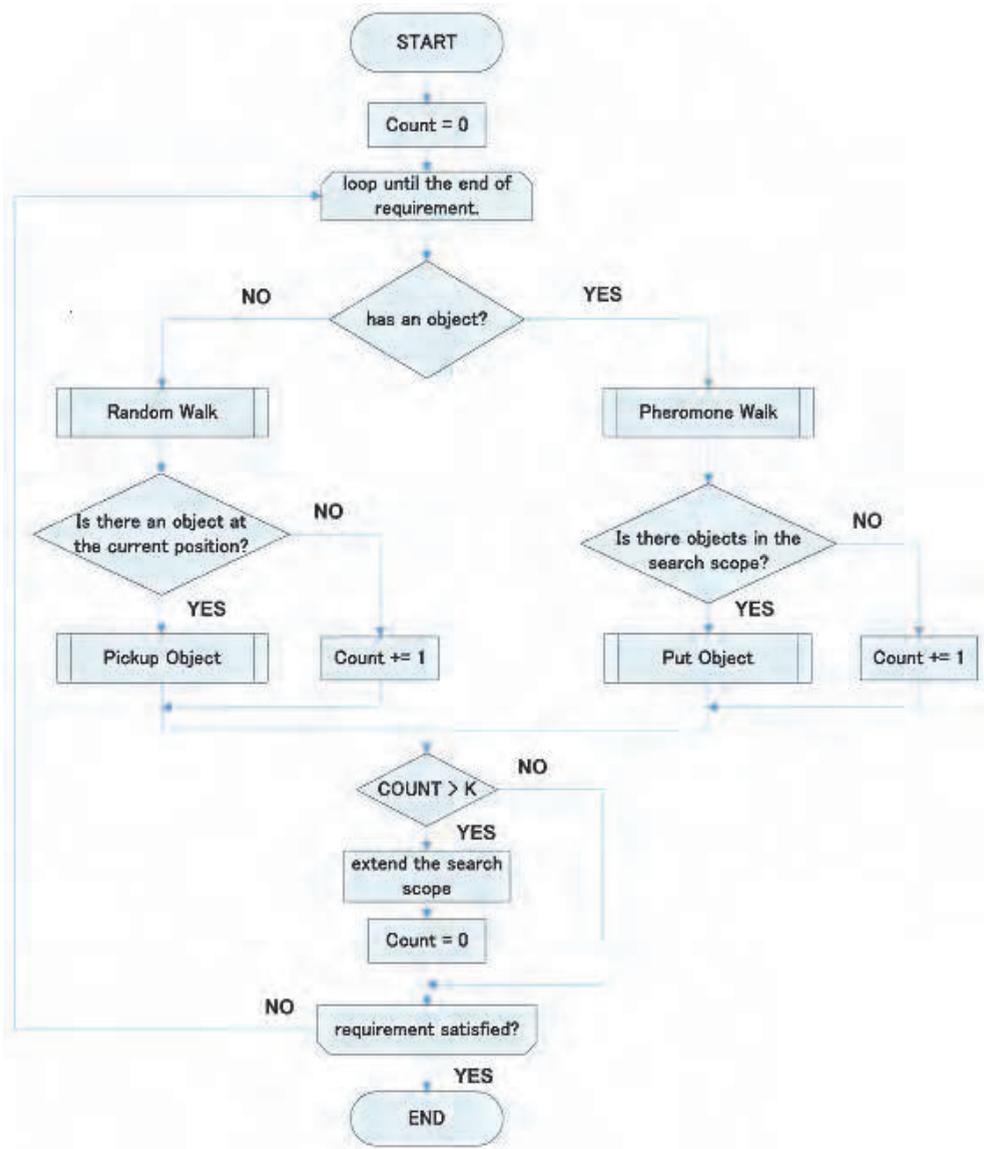


Fig. 6. The behavior of the artificial ant

Table 1 shows the summary of the numerical experiments. We have set the field size to be 100 times 100, and performed three trials of the number of ACC we perform to achieve final clustering, i.e. 1, 3, and 5. We can observe that performing ACC for five times produced the best result, i.e. the least moving distance of aggregate of all the carts. One means the simulator performs ACC only once, then all the carts try to form the given clusters. They form clusters anyway, but the number of clusters is relatively larger than in the case of larger numbers of repetitions of ACC. For 300 carts, about four clusters seem to be optimal number of clusters (see Fig. 6b). The figures Fig. 6a through 6c are typical simulation results obtained in our 300-cart example.

Performing the ACC three times produces a near optimal number of clusters, but the moving total distance is not optimal. It may be that small number (one and three in our experiments) of ACC performances can only give each cart rough idea of what to do and the carts execute futile movements. Performing the ACC five times drastically improves efficiency. We confirmed our conjecture that repetition of the ACC produces better results. The average moving distance becomes optimal. Fig. 6c shows such an optimal clustering case. The lines denote the trace; we can see each carts moves almost optimal route to form the clusters.

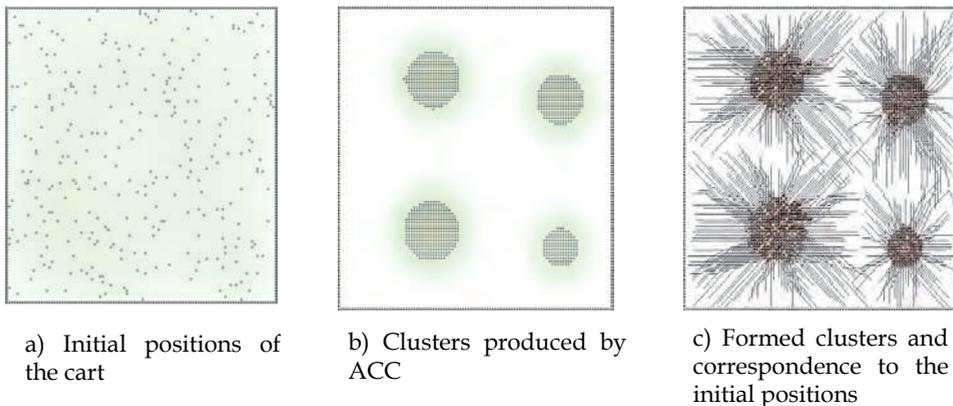


Fig. 7. Simulated results of the intelligent cart behaviors

6. Conclusion and future directions

We have presented a framework for autonomous intelligent carts connected by communication networks. Mobile and static software agents collect the coordinates of scattered carts and implement the ant colony clustering (ACC) algorithm in order to find quasi-optimal positions to assemble the carts. Making mobile multiple robots perform the ant colony optimization is enormously inefficient. Therefore a static agent performs the ACC algorithm in its simulator and computes the quasi-optimal positions for the intelligent carts. Then other mobile software agents carrying the requisite set of procedures migrate to the carts, and drive the carts using the sequence of control commands that is constructed from the computed set of procedures.

No. of ACC	Average Cluster Size	Average Moving Distances
1	6.3	11.63
3	3.3	11.83
5	3.7	2.92

Table 1. Averages of Calculated Moving Distances and Simulated Moving Distances

Since our control system is composed of several small static and mobile agents, it shows an excellent scalability. Our control framework can be applied not only intelligent cart system but also any general purpose multiple mobile robot systems. Then the number of mobile robots increases, we can simply add the increases number of mobile software agents to direct the mobile robots. The user can enhance the control software by introducing new features as mobile agents so that the multiple mobile robot system can be extended dynamically while the robots are working. Also mobile agents decrease the amount of the necessary communication. They make mobile multiple robot applications possible in remote sites with unreliable communication. In unreliable communication environments, the multiple mobile robot system may not be able to maintain consistency among the states of the robots in a centrally controlled manner. 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. In the minimal case, a mobile agent requires that the connection be established only when it performs migration (Binder, Hulaas & Villazon, 2001). The concept of a mobile agent also creates the possibility that new functions and knowledge can be introduced to the entire multi-agent system from a host or controller outside the system via a single accessible member of the intelligent multiple mobile robot system (Kambayashi & Takimoto, 2005). While our current application is simple cart collection, the system should have a wide variety of applications.

We have implemented a team of mobile robots that simulate intelligent carts to show the feasibility of our model (see Fig. 8.) In the current implementation, an agent on the robot can obtain fairly precise coordinates of the robots from RFID tags.

The ACC algorithm we have proposed is designed to minimize the total distance intelligent carts move. We have analyzed and demonstrated the effectiveness of our ACC algorithm through simulation, performing several numerical experiments with various settings. Although we have so far observed favorable results from the experiments in the simulator, we must apply the results of the simulation to a real multiple mobile robot system, and we are aware of its difficulty.

Although the intelligent carts are roughly gathered, if they are not serially aligned, the human laborers would have to align them one by one. The work is still hard and must be facilitated.

We are now re-implementing the ACC algorithm to use not only the sum of moving distances but also the orientation of each robot so that the mobile robots that are facing similar direction tend to get together. This can be achieved through employing vector values for pheromone values to compute ACC simulation.

Even though ACC computation with robots' orientations make the calculation more complex, compared with the time for robot movements, the computation time for the ACC algorithm is negligible. Even if the number of artificial ants increases, the computation time will increase linearly, and the number of objects should not influence the computation's complexity. Because any one step of each ant's behavior is simple, we can assume it takes constant execution time. Even though apparently obvious, we need to confirm this with quantitative experiments. As we mentioned in the previous section, we need to design the artificial ants to have certain complex features that change their ability to adapt to circumstances. We defer this investigation to our future work.



Fig. 8. A team of mobile robots work under control of mobile agents

For another investigation, we are designing a completely different intelligent cart assembly system where entire multiple mobile robot system performs the ACC by using mobile software agents (Oikawa, Mizutani, Takimoto & Kambayashi, 2010; Abe, Takimoto & Kambayashi, 2011). We call the system distributed ant colony clustering where two new mobile software agents are introduced to control the driving agents. They are ant agents and pheromone agents. The ant agents represent the artificial ants. They see the mobile robots and influence the driving agents to the quasi-optimal positions. The pheromone agents represent pheromone and diffuse the effects by migrations. In general, making mobile multiple robots perform the ant colony optimization has been impossible due to enormous inefficiency. Our approach, however, does not need the ant-like robots and other special

devices, because those agents are just software agents and do not require any physical movements. So far we are not aware of any multiple robot system that integrates pheromone as a control means as Deneuboug envisaged in his seminal paper (Deneuboug, Goss, Franks, Sendova-Franks, Detrain & Chretien, 1991).

By using pheromone agents, we can implement the serialization of clustered carts (Abe, Takimoto & Kambayashi, 2011). In many ways, we have room to improve our automatic cart collection system before integrating everything into one working multiple robot system.

7. Acknowledgment

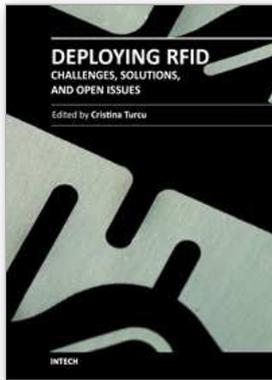
We acknowledge our colleagues Yasuhiro Tsujimura and Hidemi Yamachi. We enjoyed fruitful discussions with them. We appreciate Kimiko Gosney who gave us useful comments. This work is partially supported by Japan Society for Promotion of Science (JSPS), with the basic research program (C) (No. 20510141), Grant-in-Aid for Scientific Research.

8. References

- Abe, T., Takimoto M. & Kambayashi, Y. (2011). Searching Targets Using Mobile Agents in a Large Scale Multi-robot Environments. *Proceedings of the Fifth KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence 6682*, Berlin, Heidelberg, New York Springer-Verlag, pp. 211-220, ISBN 978-3-642-21999-3, Manchester, UK, Jun. 2011
- Binder, W.J., Hulaas, G. & Villazon, A. (2001). Portable Resource Control in the J-SEAL2 Mobile Agent System. *Proceedings of International Conference on Autonomous Agents*, pp. 222-223, ISBN 1-58113-326-X, Montreal, Canada, May 2001
- Chen, L., Xu, X. & Chen, Y. (2004). An adaptive ant colony clustering algorithm, *Proceedings of the Third IEEE International Conference on Machine Learning and Cybernetics*, pp. 1387-1392, ISBN 0-7803-8403-2, Shanghai, China, Aug. 2001
- Deneuburg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. & Chretien, L. (1991). The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot, *Proceedings of First Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pp. 356-363, MIT Press, Cambridge, MA, U.S.A., ISBN 0-262-63138-5,
- Dorigo, M. & Gambardella, L.M. (1996). Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman, *IEEE Transaction on Evolutionary Computation*, (Apr. 1997), pp. 53-66, Vol.1, No.1, ISSN 1089-778X
- Evolution Robotics Inc. Homepage (2008). <http://www.evolution.com/>.
- Finkensteller, K. (2003). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, (2nd ed.), John Wiley & Sons Ltd., ISBN 0-470-84402-7, Chichester, U.K.
- Hightower, J. & Borriello, G. (2001). Location Systems for Ubiquitous Computing, *IEEE Computer Magazine*, Vol. 34, No. 8, Aug. 2001, pp. 57-66, ISSN 0018-9162

- Kambayashi, Y. & Takimoto, M. (2005). Higher-Order Mobile Agents for Controlling Intelligent Robots, *International Journal of Intelligent Information Technologies*, Vol.1, No.2, pp. 28-42, ISSN 1548-3657
- Kambayashi, Y., Tsujimura, Y., Yamachi, H., Takimoto M. & Yamamoto, H. (2009). Design of a Multi-Robot System Using Mobile Agents with Ant Colony Clustering, *Proceedings of the 42nd Hawaii International Conference on System Sciences*, IEEE Computer Society, ISBN 0-7695-3450-3, CD-ROM, Hawaii, U.S.A., Jan. 2008
- Kambayashi, Y., Tsujimura, Y., Yamachi, H., Takimoto M. & Yamamoto H. (2010). Integrating Ant Colony Clustering Method to Multi-Robots Using Mobile Agents, In: *Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies: Intelligent Techniques for Ubiquity and Optimization*, S. H. Chen, Y. Kambayashi & H. Sato, (Ed.), pp. 174-191, IGI Global, ISBN 953-7619-81-7, Hershey, PA, U.S.A.
- Kim, M. & Chong, N. Y. (2007). RFID-based mobile robot guidance to a stationary target. *Mechatronics*, Vol.17, No.4-5, May-Jun. 2007, pp. 217-229, ISSN 0957-4158
- Lumer, E. D. & Faieta, B. (1994). Diversity and Adaptation in Populations of Clustering Ants, From Animals to Animats 3: *Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior*, pp. 501-508, Cambridge, MA, U.S.A., MIT Press, ISBN 0-262-53122-4
- Nagata, T., Takimoto, M. & Kambayashi, Y. (2009). Suppressing the Total Costs of Executing Tasks Using Mobile Agents. *Proceedings of the 42nd Hawaii International Conference on System Sciences*, IEEE Computer Society, ISBN 0-7695-3450-3, CD-ROM, Hawaii, U.S.A., Jan. 2008
- Niigata Seimitsu Co. Ltd. Homepage (2011).
http://www.niigata-s.co.jp/products/module/reader_writer/arw13t.htm
- Oikawa, R., Mizutani, M., Takimoto M. & Kambayashi, Y. (2010). Distributed Ant Colony Clustering Using Mobile Agents and Its Effects. *Proceedings of the 14th KES International Conference on Knowledge-based and Intelligent Information and Engineering Systems: Part I, Lecture Notes in Artificial Intelligence 6276*, Berlin, Heidelberg, New York Springer-Verlag, pp. 198-208, ISBN 978-3-642-15386-0, Cardiff, UK, Sep. 2010
- Satoh, I. (1999). A Mobile Agent-Based Framework for Active Networks. *Proceedings of IEEE Systems, Man, and Cybernetics Conference*, pp. 161-168, ISBN 0-7803-5731-0, Tokyo, Japan, Dec. 1999
- Takimoto, M., Mizuno, M., Kurio, M. & Kambayashi, Y. (2007). Saving Energy Consumption of Multi-Robots Using Higher-Order Mobile Agents, *Proceedings of the First KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence 4496*, Berlin, Heidelberg, New York Springer-Verlag, pp. 549-558, ISBN 3-540-72830-6, Wroclaw, Poland, May, 2007
- Wang T. & Zhang, H. (2004). Collective Sorting with Multi-Robot. *Proceedings of the First IEEE International Conference on Robotics and Biomimetics*, pp. 716-720, ISBN 0-7803-8614-8, Shenyang, China, Aug. 2004

Werb, J. & Lanzl, C. (1998). Designing a Positioning System for Finding Things and People Indoors. *IEEE Spectrum*, Vol.35, No. 9, pp. 71-78, ISSN 0018-9235



Deploying RFID - Challenges, Solutions, and Open Issues

Edited by Dr. Cristina Turcu

ISBN 978-953-307-380-4

Hard cover, 382 pages

Publisher InTech

Published online 17, August, 2011

Published in print edition August, 2011

Radio frequency identification (RFID) is a technology that is rapidly gaining popularity due to its several benefits in a wide area of applications like inventory tracking, supply chain management, automated manufacturing, healthcare, etc. The benefits of implementing RFID technologies can be seen in terms of efficiency (increased speed in production, reduced shrinkage, lower error rates, improved asset tracking etc.) or effectiveness (services that companies provide to the customers). Leading to considerable operational and strategic benefits, RFID technology continues to bring new levels of intelligence and information, strengthening the experience of all participants in this research domain, and serving as a valuable authentication technology. We hope this book will be useful for engineers, researchers and industry personnel, and provide them with some new ideas to address current and future issues they might be facing.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yasushi Kambayashi and Munehiro Takimoto (2011). Location of Intelligent Carts Using RFID, Deploying RFID - Challenges, Solutions, and Open Issues, Dr. Cristina Turcu (Ed.), ISBN: 978-953-307-380-4, InTech, Available from: <http://www.intechopen.com/books/deploying-rfid-challenges-solutions-and-open-issues/location-of-intelligent-carts-using-rfid>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.