Modeling and Development of an Autonomous Pedestrian Agent – As a Simulation Tool for Crowd Analysis for Spatial Design

Toshiyuki Kaneda, Nagoya Institute of Technology, (kaneda@nitech.ac.jp)
Yanfeng He, Hitachi Systems & Services, Ltd., (he_yanfeng@hotmail.com)

Abstract: At present, it is expected that pedestrian agent simulation will be applied to not only accident analysis, but also spatial design; ASPF (Agent-based Simulator of Pedestrian Flows) has already been developed as a simulator for such purposes. However, in the present version ASPFver.3 a pedestrian agent merely walks straight ahead and simply avoids other agents, and it had been impossible to analyze crowd flows on a large-scale space with a complicated shape, a function is required that enables an agent to walk along a chain of visible target 'waypoints' to each destination, as well as a function the agent keeps the direction to the target. The study introduces newly a target maintaining (Helmsman) function, a concept of waypoint, and update mechanism of targets, and develops the simulator ASPFver4.0 that models an autonomous pedestrian agent on ArtiSoc(KKMAS). The performances tests of these additional functions of ASPFver4.0 are shown. Especially, to successfully model pedestrians' shop-around behavior in a Patio-style shopping mall at Asunal Kanayama, Nagoya, ASPFver4.1 has been also developed by introducing an optimization function of routes by Dijkstra method, and implemented several parameters based on data for survey of the pedestrians' behaviors in this mall. Through the test of four simulation cases; (1) weekday case, (2) weekday double case, (3) holiday case, and (4) at time of a music event in holiday case, the performance of ASPFver4.1 was also verified. Due to a series of these version-ups, we can conclude that ASPF is now available for analyzing crowd flows and density in space with complicated shapes.

1. Research Background and Objectives

At present, it is expected that pedestrian agent simulation will be applied to not only accident analysis, but also spatial design; ASPF (Agent-based Simulator of Pedestrian Flows) has already been developed as a simulator for such purposes.[5-9] However, in the present version ASPFver.3 a pedestrian agent merely walks straight ahead and simply avoids other agents, and it had been impossible to carry

out a simulation on a large-scale space with a complicated shape.[9] In particular, to successfully model pedestrians' behavior in a Patio-style mall and in order for an agent to reach their destination by moving from point to point, a function is required that enables an agent to move along a route made up of a chain of visible waypoints linking each target destination.[4,12]

The study introduces newly a target maintaining function, concepts of waypoint, update of targets and optimization of routes, and develops the simulator ASPFver4 that models an autonomous pedestrian agent with a function to move toward a target destination on ArtiSoc(KKMAS)[i]. More specifically, characteristics of this simulator are described through the analysis of crowd density in a simulation of pedestrian behavior within the Patio-style mall at Asunal Kanayama, Nagoya.

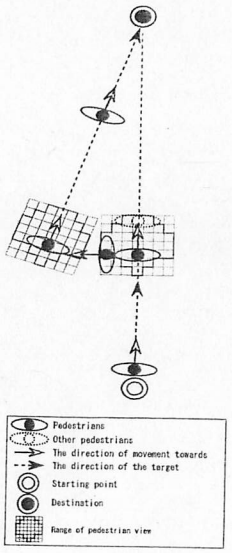

Fig.1 Target maintaining function

2. APSFver.4.0: Modeling an Autonomous Pedestrian Agent with a Function to Move toward a Target Destination

2.1 Improvement of Pedestrian Agent toward ASPFver4.0

In ASPFver4.0, in order to install a movement to destination function into a pedestrian agent, a target maintaining (Helmsman) function, the concept of waypoint and walking routes were introduced,

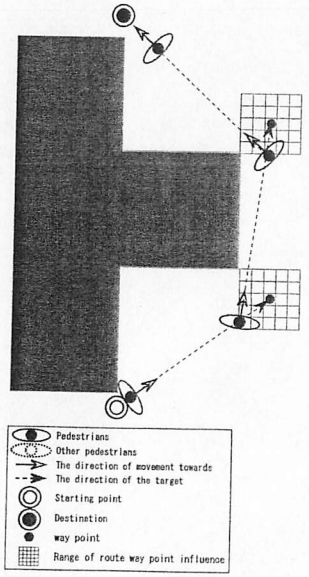

Fig.2 Movement to destination

and a target update algorithm at waypoints was implemented.[4, 12]

The target maintaining function (Fig. 1) refers to a function that firstly determines the direction of movement towards a given (visible) target and secondly by regularly reconfirming the location of the target corrects any difference of directions between the movement and the target; such differences may occur because of behavior to avoid other pedestrians and walls while moving towards the target. In addition, in a large-scale and complicated shaped space, there is no guarantee that a destination can be always confirmed visually; in this case, a list of waypoints that satisfy visual confirmation conditions from the starting point to the destination is given in advance and a pedestrian agent walks along the list of the waypoints (Fig. 2). The target mentioned above is either the final target destination or a waypoint. Update of the target means that the agent regularly confirms whether they are closing on their target and when they arrive in the neighborhood of the target, they update their target to the next one.

Furthermore, pedestrian behavior is affected by not only other pedestrians, but also walls and both these factors have different characteristics; therefore, version 4.0 was designed to allow the installation of wall agents at any location in the form of a unit of cells.

2.2 Structure of ASPFver.4.0

The general structure of the simulator followed the basic design of ASPF3.0. [7,8] The spatial scale is represented by 40 square cm cells and the time scale is set at one step per 0.5 seconds. In ASPFver4.0, with the introduction of a target maintaining function, target update and walls, in order to maintain the integrity of these new elements, additions and changes were made to the walking behavior rules.

Behavior rules were applied to an agent in the following order: (1) set a route; (2) maintain the target; (3) walking behavior rules; and (4) update the target (Fig. 3). The following parameters were set: confirmation to maintain a target was carried out every 10 steps, target update was confirmed every 2 steps by checking whether they were located within the 2 cells from a waypoint. Due to the introduction of a wall agent, walking behavior rules increased to a total of 36 rules comprising 14 new wall avoidance rules and 25 rules taken from the previous version, comprising 6 basic behavior rules, 8 slow-down rules, 4 avoidance rules, 3 high density flow rules and 1 pattern cognition rule (Fig. 4). Wall avoidance rules were designed to avoid a wall by changing the direction of movement. The existence of other agents was basically judged by using a relative coordinate system, but in order to standardize the differences in unit distances caused by each agent's individual progress, the existence of a wall agent was judged on an absolute coordinate system.

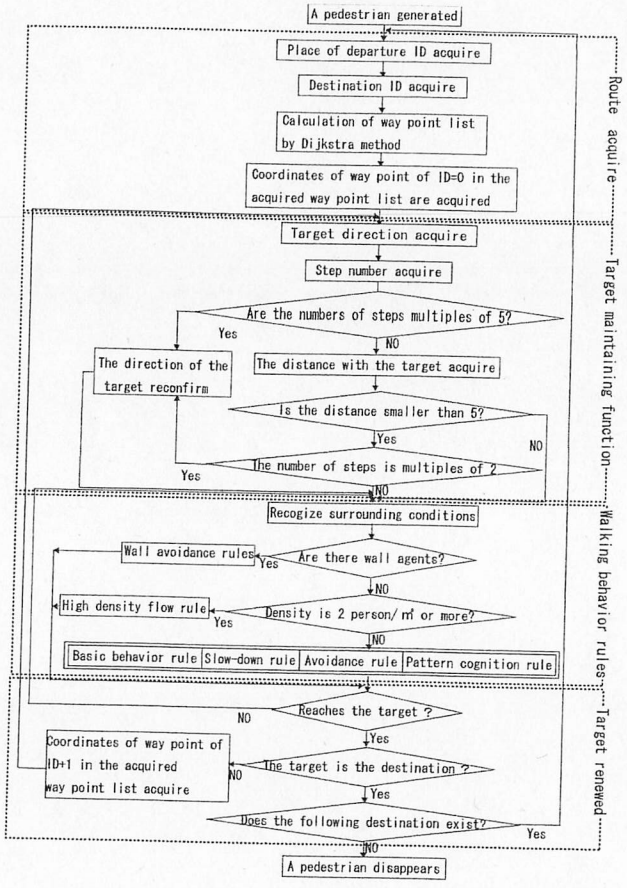

Fig. 3 Pedestrian agent's algorithm

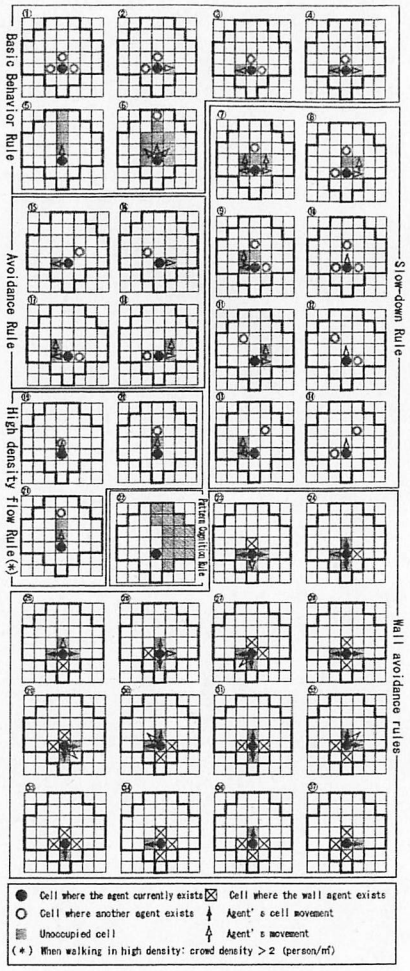

Fig. 4 Pedestrian behavior rules

2.3 Example of 'Crowd Flows-Crossing' Simulation

Here, the target maintaining function was demonstrated. In a cross-shaped space, 40 cells in road width, opposing flows with a flow coefficient of 0.5 person/m · sec were generated from both the right and left sides, and after the number

of pedestrians had become steady, three crossing pedestrians were generated from the lower part and the loci of these pedestrians were examined (Fig. 5). Target maintaining behavior by agents crossing a pedestrian flow from two different directions was confirmed. We can see that the agents kept holding their target destination by correcting their direction though he/she had drifted by the flows. Moreover, in this study, simulations were tested the relationship between density and speed with a straight movement flow, and in an L-shaped corridor, and the same results as in the previous version were confirmed.

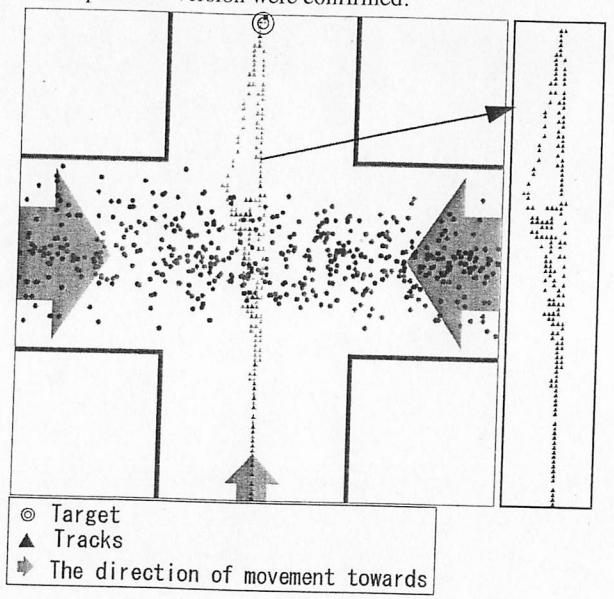

Fig5 Target maintaining behavior

3 ASPFver4.1: Autonomous Pedestrian Agent Simulation with an Introduction of a Route Optimization Function

3.1 Characteristics of ASPFver.4.1

By improving ASPFver.4.0, ASPFver.4.1 was developed, which deals with pedestrian behavior in the Patio-style mall standing next to Kanayama railway

Station, Asunal Kanayama, and an attempt was made to apply this updated version to the analysis of crowd density. Firstly, by using the data of pedestrian behavior research that was conducted in the previous year, an agent behavior algorithm shown in Fig. 6 was established. This version has the following characteristics: first, create a list of shop-around facilities from research data by attributes; then by using the Dijkstra method for movement between facilities, find and follow the shortest route. Moreover, in order to express the crowd density at the time of an event, an event routine was added; when an music event starts, the 80% of the agents within the facilities begins to gather in the event square and at the end of the event, these agents continue shop-around behavior again.

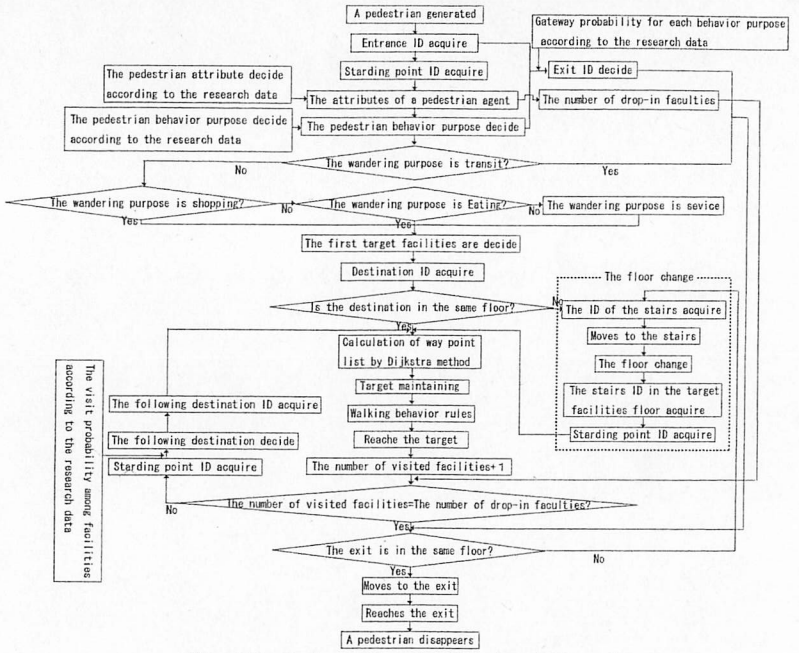

Fig.6 Wandering action algorithm by ASPFver.4.1

3.2 Crowd Density Analysis of Pedestrian Behavior in a Patio Style Mall

This section describes a simulation experiment of an area of 150m x 100m on the ground floor of Asunal Kanayama. The target area covered 300 × 250 cells and comprised 11 gateways (10 gateways for the ground floor, 1 gateway for 2nd

and 3rd floors), 19 stores for each store type (18 stores for the ground floor, 1 store for 2nd and 3rd floors). 105 waypoints (87 points for ground floor, 18 points for 2nd and 3rd floors) were set up: all the waypoints formed a route network and conformed to visual confirmation conditions. When any starting point and destination were given, one of 7,656 set shortest routes, found by using the Dijkstra method, was used.

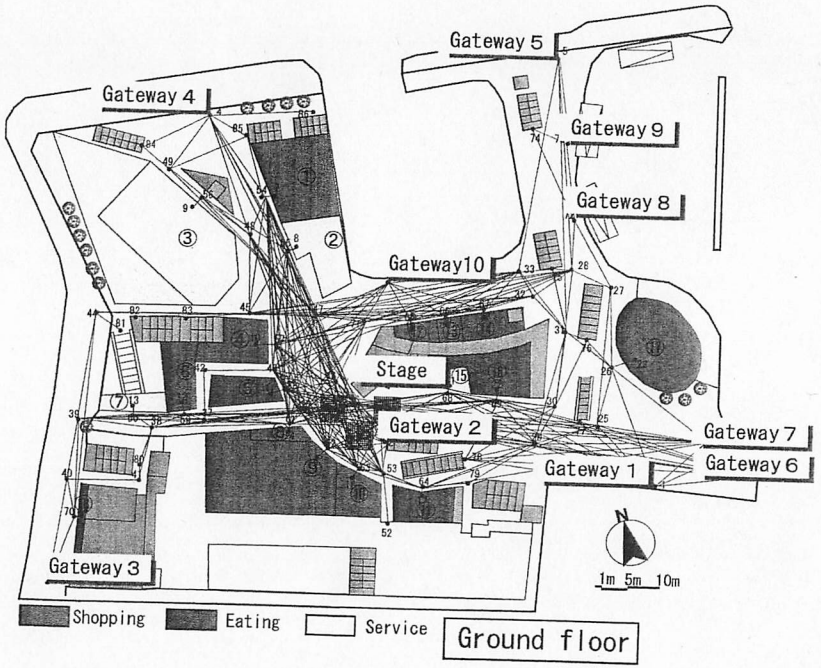

Fig.7 Layout of Asunal Kanayama and waypoints and route network

The attributes of a pedestrian agent were set by sex. Four categories of behavior purpose were set: shopping, eating and drinking, service and transit. For the number of drop-in faculties, 0 was given for the case of transit; for all other cases, based on research data, a random number value was obtained with a Poisson distribution using a minimum value of 1, a maximum value of 13 and an average value by attribute. From the research data, the pedestrian inflow rate at the entrances was obtained for weekdays and holidays.

Exit points were set for each behavior purpose. In the case of transit, the exit point was determined by a random number and for other purposes, 90% of pedestrians left the facilities from the same gateway as they entered, and 10% were determined by a random number. The first drop-in store was set on the same floor as

the entry gateway for 90% of pedestrians and for all behavior purposes. The visit probability among facilities was set according to the research data.

In the simulation, as shown in Fig. 8, three areas were set for density measurement and a simulation was carried out for the following four cases: (a) weekday case; (b) weekday double case – the number of visitors on a weekday was doubled; (c) holiday case; and (d) holiday event case. In the simulation experiments, the value at the time of 600 steps, when the number of agents generally became steady for all cases, was measured.

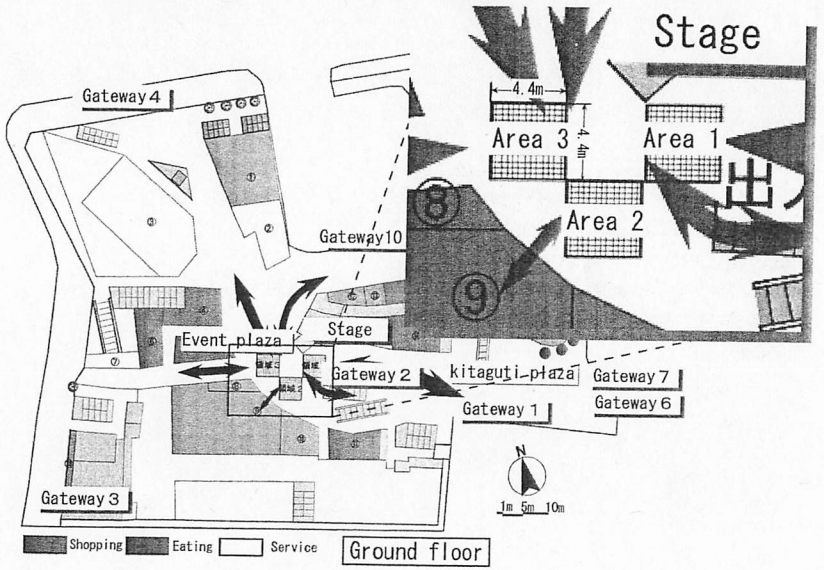

Fig.8 Density measurement areas

3.3 Analysis of Crowd Density Simulation Results

Fig. 9 shows the simulation results. In both the weekday and holiday cases, the density in the passage that leads from the event square to the west (Area 3) was relatively high (0.98 person/m^2 for both weekday and holiday). In the holiday case, the density in the passage that leads from the square to the square at the north entrance (Area 1) is high (1.6 person/m^2). This result conforms to the passing rate trends obtained from the previous year's research. In the weekday double case, as shown in Areas 2 and 3, the density increased in a different location from the

weekday case. Apart from Area 1 of the weekday double case (2.22 person/m²); it was found that the density did not exceed 2.0.

Fig. 10 shows changes of density in Area 1 in the case of an event. After an event occurred, the density increased and after the end of the event, the density continued to increase for a time before declining to the same level as the holiday case. When the duration of the event time was set at 150 seconds, which is 1.5 times more than the norm, it was found that the density at peak time also went up. The conditions set were extreme; however, these results suggest that crowd control would be required at the time of an event.

4. Conclusion

The study developed a pedestrian flow simulator ASPFver4, in which functions of a high order such as movement to a destination and drop-in facilities – a chain of movement – were implemented by introducing a change in direction using a target maintaining function and waypoints into an agent, in addition to the basic functions such as avoidance, following and overtaking; performance was confirmed. Due to these improvements, we think it is now possible to apply this simulator to analysis of crowd density in space with complicated shapes, although we need to study much more cases of spaces and clarify the limitation of this version.

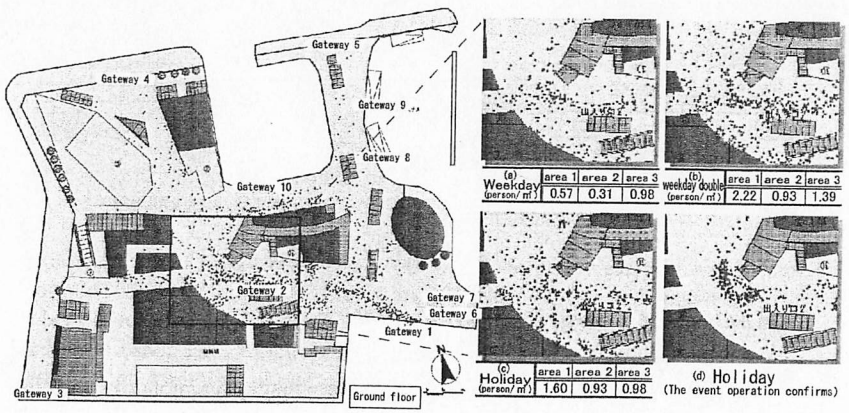

Fig.9 Simulation results in four cases

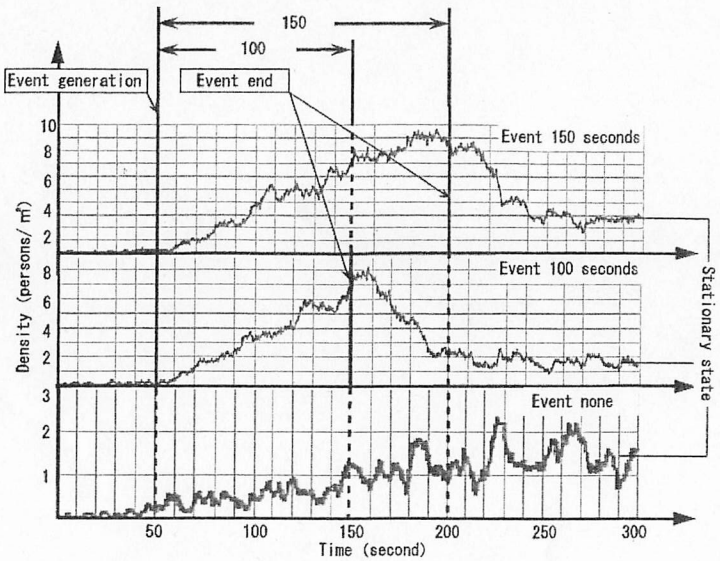

Fig.10 Density rise at event (holiday area 1)

References

1. Antonini G, Bierlaire M (2007) A Discrete Choice Framework for Acceleration and Direction Change Behaviors in Walk Pedestrian. In: Waldau W et al (eds) Pedestrian and Evacuation Dynamics 2005. Springer-Verlag.
2. Batty M, DeSyllas J et al (2002) Discrete Dynamics of Small-Scale Spatial Events: Agent-Based Models of Mobility in Carnivals and Street Parades. Working Paper 56, Centre for Advanced Spatial Analysis. University College London.
3. Borgers A, Timmermans HA (1986) A Model of Pedestrian Route Choice and Demand for Retail Facilities within Inner-City Shopping Areas. Geographical Analysis 18: 115-128.
4. Haklay M, Thurstain-Goodwin MO et al (2001) "So Go Downtown " :Simulating Pedestrian Movement in Town Centres. Environment and Planning B 28: 343-359.
5. Kaneda T, Yano H et.al (2003) A Study on Pedestrian Flow by Using an Agent Model - A Simulation Analysis on the Asagiri Overpass Accident, 2001--. In: Terano T, Deguchi H et al (eds), Meeting the Challenge of Social Problems via Agent-Based Simulation. Springer.

6. Kaneda T (2004) Agent Simulation of Pedestrian Flows. Journal of the Society of Instrument and Control Engineers 43/12: 950-955. (in Japanese)
7. Kaneda T, Suzuki T (2005) A Simulation Analysis for Pedestrian Flow Management. In: Terano T et al (eds) Agent-Based Simulation From Modeling Methodologies to Real-World Applications. Springer.
8. Kaneda T (2007) Developing a Pedestrian Agent Model for Analyzing an Overpass accident. In: Waldau.W et al (eds) Pedestrian and Evacuation Dynamics 2005. Springer-Verlag.
9. Kaneda T, Okayama D (2007) Pedestrian Agent Model Using Relative Coordinate Systems. In: Terano T et al (eds) Agent-Based Simulation: From Modeling Methodologies to Real-World Applications. Springer.
10. Lovas GG (1994) Modeling and Simulation of Pedestrian Traffic Flow. Transportation Research B 28B/6: 429-443.
11. Penn A, Turner A (2002) Space Syntax Based Agent Simulation. In: Waldau.W et al (eds) Pedestrian and Evacuation Dynamics. Springer-Verlag.
12. Schelhorn T et al (1999) STREETS: An Agent-Based Pedestrian Model. Working Paper 9, Centre for Advanced Spatial Analysis. University College London.

[i] Developed by Kozo-Keikaku Engineering Inc.

Agent-Based Adaptive Production Scheduling – A Study on Cooperative-Competition in Federated Agent Architecture

Jayeola Femi Opadiji [1] and Toshiya Kaihara [2]

1) Graduate School of Science and Technology, Kobe University, Japan.
2) Graduate School of Engineering, Kobe University, Japan.
femi@kaede.cs.kobe-u.ac.jp and kaihara@cs.kobe-u.ac.jp

Abstract. An increasingly popular method of improving the performance of complex systems operating in dynamic environments involves modeling such systems as social networks made up of a community of agents working together based on some basic principles of social interaction. However, this paradigm is not without its challenges brought about by the need for autonomy of agents in the system. While some problems can be solved by making the interaction protocol either strictly competitive or strictly cooperative, some other models require the system to incorporate both interaction schemes for improved performance. In this paper, we study how the seemingly contradictory effects of these two behaviours can be exploited for distributed problem solving by considering a flexible job shop scheduling problem in a dynamic order environment. The system is modeled using federated agent architecture. We implement a simple auction mechanism at each processing center and a global reinforcement learning mechanism to minimize cost contents in the system. Results of simulations using the cooperative-competition approach and the strictly competitive model are presented. Simulation results show that there were improvements in cost objectives of the system when the various processing centers cooperated through the learning mechanism, which also provides for adaptation of the system to a stream of random orders.

1 Introduction

The need for very robust and flexible models for complex system simulation explains the growing trend in exploiting theories of agent interactions from the social sciences. These theories range from market-based theories in economics to social contract and culture formation theories in sociology. The main impact of these theories is in the construction of interaction protocols for a society of agents co-habiting in a specified environment. In order to employ such protocols however, it is necessary to clearly define the nature and behaviour of each type of agent existing in the environment, as well as the environmental conditions under which they exist.

One important area of application of Multiagent system (MAS) paradigm is in solving problems relating to the supply network of organizations. Problems such as manufacturing resources allocation, production system scheduling, distribution network planning, increase in complexity with increase in the size of a supply network.