

An Evolutionary Scheduling Paradigm Using Coordinated Autonomous Agents

T. Kaihara¹, S. Fujii² and S. Kunimasa²

*¹ University of Marketing and Distribution Sciences
Faculty of Information Science*

3-1, Gakuen-nishi, Nishi, Kobe 651-2188, JAPAN

Phone: +81-78-796-4402, Fax: +81-78-794-3054

E-mail: kaihara@umds.ac.jp

² Kobe University, Faculty of Engineering

Rokkoudai, Nada, Kobe 657, JAPAN

E-mail: fujii@seg.kobe-u.ac.jp, kunimasa@kaede.seg.kobe-u.ac.jp

Abstract

Nowadays de-centralized scheduling architecture with flexibility and robustness is required in manufacturing system. This paper studies the abilities of multi-agent paradigm with simple architecture and primitive algorithm to move appropriately in relation to several process machines. Efficient individual behaviour is obtained, and self-organized work flow is formulated with coordinated motions. An evolutionary mechanism is introduced into the self organized scheduling methodology and improves scheduling performance in terms of efficiency and robustness in our simulation model. The evolved work agents are capable of robustness and reliability and facilitate autonomous scheduling system.

Keywords

Scheduling, Evolution, Multi agent, Coordination, Genetic algorithm

1 INTRODUCTION

Agent-oriented programming is one of the latest topics in information technology (Shoham, 1993). It is well known in the field of science, engineering, social system, economics and so forth. The concept of agent has been mainly proposed in the area of artificial life (Langton, 1986).

The definition of agent varies in the application fields (Ishida, 1995). The definition is generally classified into four items as follows:

i) Autonomy

Agents solve their facing problems by using their own knowledge and recognizing their situations all by themselves.

ii) Pro-activeness

Agents join tasks actively towards their unified goal.

iii) Social ability

Agents solves their problems by reciprocal relation with other agents or human. Common protocol is implemented to make the communication and negotiation.

iv) Reactivity

Agents make an appropriate response against their environmental changes.

These characteristics facilitate a scheduling algorithm based on coordinated motions of autonomous agents.

Conventional scheduling approach in manufacturing system, however, is based on centralized control architecture, and it is quite difficult to get an universal solutions in all conditions due to the lack of flexibility and robustness(Kaihara, 1996a).

This paper presents a universal dynamic scheduling algorithm with multi-agent paradigm for manufacturing systems. The best known of natural collective multi-agent systems are social insects: they use simple components; they use many identical insects; they exhibit system reliability; they adapt to environmental changes; they are robust with respect to individual insect failures(Shone, 1984). The idea of making scheduling algorithm which are modelled on such systems is therefore very attractive.

The evolutionary concept must play an important role in the multi-agent based scheduling paradigm, because the coordination protocols implemented in each agent will be sophisticated gradually by the evolution mechanism. Genetic algorithm (GA) is a kind of evolutionary programming methods and a search procedure based on the mechanics of natural selection and natural genetics(12). GA allows computers to evolve solutions to difficult problems, such as function optimization and artificial intelligence.

An universal dynamic scheduling paradigm with evolutionary multi-agent system based on GA is proposed in this paper for job shop type manufacturing system. Our goal is to propose a decentralized universal scheduling concept which is robust against dynamic environmental changes despite its simple architecture. Multi-agent based manufacturing system is modelled and simulation technique is introduced to analyze its characteristics and prove its effectiveness.

2 SCHEDULING MODEL

Agent prerequisite

Each work moving around in manufacturing system is defined as agent in our model. Simple architecture which conducts self-organization and solves unified goal is one of our research topics. Work agent is defined by following assumptions:

i) minimal sensing

The simplest possible sensor will be sensitive to a single type of strictly local environmental variation. In our model each work agent is sensitive only the direction to its target process machine.

ii) minimal mobility

The movements of the work agents will be limited only 1)step forward 2)turn right 3)turn left 4)turn back. In case an obstacle is on their way, they will change their direction.

iii) minimal communication

Each work agent will communicate only to the agent in front. The information exchanged is limited to 1) agent type 2) moving direction.

iv) minimal computation

The only computation undertaken by the work agent will be the calculation of the change in direction, as a function of the value of the sensor input.

v) minimal environment

Our work agent will exist in the simplest possible environment to start with a plane, with a square boundary (100, 100) which agents bounce off but cannot sense.

Scheduling problem

Scheduling problem in this paper is formulated by following notation:-

<i>M</i> :	Total number of process machines
• :	Total number of agents
<i>TTY</i> :	Total number of agent type
<i>N_i</i> :	Total number of operations of agent <i>i</i>
<i>Fi</i> :	The number of finished operations of agent <i>i</i>
<i>O_{ij}</i> :	Operation index <i>j</i> of agent <i>i</i>
<i>TY_i</i> :	Type of agent <i>i</i>
<i>PR_{it}</i> :	Scheduling priority of agent <i>i</i> at time <i>t</i>
<i>To_{ij}</i> :	Operation time of <i>O_{ij}</i>
<i>PM_{oij}</i> :	Process machine index of <i>O_{ij}</i>
<i>X_{oij}</i> :	Total number of alternative machines of <i>O_{ij}</i>
<i>P_{it(x,y)}</i> :	Position(<i>x,y</i>) of agent <i>i</i> at <i>t</i>
<i>P_{io(x,y)}</i> :	Initial position of agent <i>i</i>
<i>P_{m(x,y)}</i> :	Position(<i>x,y</i>) of machine <i>m</i>
<i>D_{it}</i> :	Current direction of agent <i>i</i> at <i>t</i>
<i>D_{io}</i> :	initial direction of agent <i>i</i>
<i>MTBF_m</i> :	MTBF of machine <i>m</i>
<i>MTTR_m</i> :	MTTR of machine <i>m</i>
<i>T</i> :	Simulation finish time
<i>V_i</i> :	Velocity of agent <i>i</i> (step/ <i>t</i>)
<i>TG_{it}</i> :	Direction set to target of agent <i>i</i> at <i>t</i>

Internal condition of agent i at time t is defined as following notation:

$$WorkAgt\ i = \{Pit(x,y), Dit, PMoij, PRit\} \tag{1}$$

Performance evaluation parameter is average operation completion ratio (OCR), defined as follows:

$$OCR = \frac{\sum_{i=1, W} Fi}{W} \tag{2}$$

3 COORDINATED MOTION

Coordination algorithm

Coordination amongst work agents is established through their minimal communication. Generally coordination algorithms introduce the selection mechanism of several heuristic rules which have been equipped in each agent. An “entrainment” algorithm is proposed for agent protocol as a basic study of coordination phenomenon in our scheduling problem(Kaihara, 1996b).

The proposed coordinated algorithms are explained in Figure 1. Two types of work agent are considered to treat rapid product development problem. QTAT (Quick Turn Around Time) work agents, so called ‘Priority work’ or ‘Hot work’, have higher scheduling priority than normal ones(Kaihara, 1997).

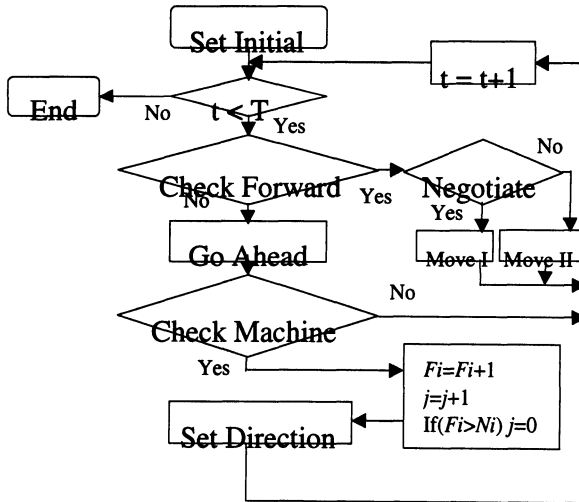


Figure 1 Coordination algorithm

where

Set Initial: $Pit(x,y) = Random((1, \dots, 100)(1, \dots, 100))$, $Di_0 = Random(N, S, E, W)$, $t=0, j=0$

Check Forward: $if(\sum WorkAgt_i(t) \cdot i) / (P_i(t(x,y)) - Pit(x,y)) = 1) = True$

Negotiate: $\text{if}(\bullet \text{ WorkAgent}(i) \bullet (Di \bullet TGit)) = \text{True}$
Move I: $(Pi(t+1)(x,y) = Pit(x,y) + Vi \bullet L(Di \bullet t)) \bullet (Di(t+1) = Di \bullet t)$
Move II: $(Pi(t+1)(x,y) = Pit(x,y)) \bullet (Di(t+1) = \text{Random}(Dit))$
Go Ahead: $(Pi(t+1)(x,y) = Pit(x,y) + Vi \bullet L(Dit)) \bullet (Di(t+1) = Dit)$
Check Machine: $\text{if}(|Pit(x,y) - Pm(x,y)| = 1 \bullet m = PMoij) = \text{True}$
Set Direction: $(Pi(t+1)(x,y) = Pit(x,y)) \bullet (Di(t+1) \bullet TGit)$

At first, initial internal conditions of the agents are set at '**Set Initial**'. Their initial positions are defined in the plane with a square boundary (100, 100) followed by uniform-random distribution. Their initial direction is selected from a set of (N: North, S: South, E: East, W: West) based on also uniform-random distribution.

After a work agent checks the existence of the other agents in front with its minimal sensor, minimal information about the priority and the direction is exchanged between the agents at '**Check Forward**'.

Minimal computation is undertaken in '**Negotiate**' followed by the entrainment algorithm, which is observed in the movement of several social insects. If the confronted two agents are equivalent in their moving direction, both never change their direction and one is entrained by the other ('**Move I**'). But otherwise they try to change their direction randomly to avoid the collision or deadlock ('**Move II**'). When a work agent reaches its target machine ('**Check Machine**'), it increments the internal conditions, and set a new direction toward the next target ('**Set Direction**').

QTAT work agents will cooperate only with other QTATs, that means QTAT steps into '**Negotiate**' only in the confrontation with other QTATs in the flow chart.

4 EVOLUTIONARY COMPUTATION

Optimization and robustness are generally under trade-off relations in scheduling problem. The proposed coordinated scheduling algorithm aims at the latter essentially. Well-trained groups of the work agents, however, will gain the better scheduling solutions without losing their robustness. Therefore the application of evolutionary computing into the work agents should be promising.

Genetic algorithm (GA) is one of the most popular evolutionary computations and derives its behaviour from a metaphor of the processes of evolution in nature. We applied GA approach to reinforce the coordinated work agents.

A genotype information, which defines the possible movements in negotiation, of the work agent is shown in Figure 2. Each agent has 6 bit genotype in length. For example, the agent which selects a genotype "011010" can change its direction into left or back with probability 50% in the '**MOVE II**' as negotiated action.

Two types of fitness functions and reproduction rules are examined in this paper as follows:

(1) Fitness function

- Fitness=0: The number of finished operations of agent i ($= Fi$)

•Fitness=1: The number of moved steps closer toward the target machine
 (2) Reproduction rule

•Elite=0: reproduce all the individuals (Chromosomes)

•Elite=1(Elitism): reproduce worst half of the individuals in fitness value
 As a selection rule of the individuals, we adopted roulette selection strategy, which selects the individuals in fitness-proportionate manner. The other experimental parameters are as follows:

Crossover rate = 50%

Mutation rate = 1%

The time step for evolution is given as a parameter in the experiment.

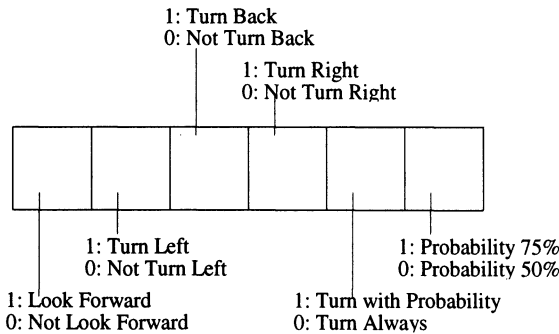


Figure 2 A genotype Information

5 EXPERIMENT RESULTS AND DISCUSSIONS

A simulation model is developed to make basic analysis of evolutionary self-organized scheduling paradigm with respect to its effectiveness and robustness.

Simulation conditions

Default simulation conditions are as follows:

$$M=3, W=100, TTY=2 (TYi=1: 50, TYi=2: 50), Ni=3, Toij=1, Vi=1$$

Followings are the process machine sequence definitions:

$$\text{if}(\bullet \text{ TAgtit}(TYi=1)=\text{True}) \text{ then } PMoi1 =1, PMoi2 =2, PMoi3 =3 \quad (3)$$

$$\text{if}(\bullet \text{ TAgtit}(TYi=2)=\text{True}) \text{ then } PMoi1 =3, PMoi2 =2, PMoi3 =1 \quad (4)$$

Robustness of self-organization scheduling

Scheduling robustness under various dynamic environment is our major concern about cooperative movement of multi-agents. One of the most popular non-linear noise about job shop type manufacturing system, QTAT work, is focused in this paper.

Figure 3 shows the simulation result about the robustness against QTAT works. The denominator and numerator in QTAT parameter means the number of total work agent and QTAT work agent in the simulation respectively.

This experiment indicated us several interesting characteristics of the proposed paradigm. At first, we observed the robustness against QTAT work disturbance. Deadlock never happened no more than 20 in the number of QTAT works. Secondly not only the robustness against QTAT work but also noteworthy 'leadership' role of social society were observed in our work agent movement. Until the number of QTATs reaches 20, OCR becomes higher, contrary to our initial expectations. The work movement animation showed us the interesting fact; a QTAT work agent attended several normal work agents and conducted them to appropriate direction as a leader. QTAT works play an important role to form a global agent flow quickly. We call it 'leader-ship phenomenon' of the autonomous work agents, and the phenomenon is related to the mass decision making process in our society. Once the number of QTAT is over the threshold(>20), deadlocks occur and OCR progress stops as we expected, because QTAT work agents never negotiate with normal ones and act selfishly. Too many selfish agents can't formulate cooperative global work flow.

Finally, all the agents are QTAT works, OCR progress recovers. QTAT work agents can resolve interactions, because the negotiation amongst QTAT is admitted.

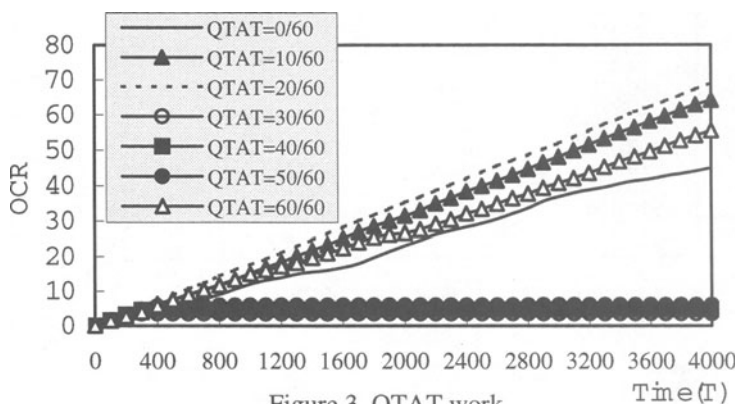


Figure 3 QTAT work
($W=60, PRit=0,1, X_{oij}=1, T=4000$)

Evolutionary effects

The effect of GA approach is shown in Figure 4. The parameters in the simulation experiment were described in chapter 4. The performance in the proposed evolutionary cooperative scheduling has increased up to 40% compared with the conventional cooperative scheduling. The final genotype information in the parameters (Fitness=1, Elite=1) at 4,000 simulation time is shown in Figure 5. It is observed that the individuals converge into 2 types, (010001) and (010011), in this experimental conditions. The comparison of scheduling robustness between non-evolution and evolution algorithms is shown in Figure 6. As agent population

increases, it generally becomes difficult to formulate autonomous agent flow. Finally Figure 7 shows the evolutionary time span effects on robustness.

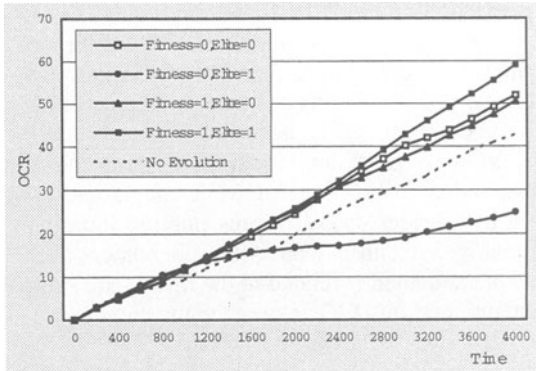


Figure 4 Effectiveness of evolution

WorkID	Chromosome	WorkID	Chromosome	WorkID	Chromosome	WorkID	Chromosome
0	OPPOP	15	OPPOP	30	OPPOP	45	OPPOP
1	OPPOP	16	OPPOP	31	OPPOP	46	OPPOP
2	OPPOP	17	OPPOP	32	OPPOP	47	OPPOP
3	OPPOP	18	OPPOP	33	PPPOP	48	OPPOP
4	OPPOP	19	OPPOP	34	OPPOP	49	OPPOP
5	OPPOP	20	OPPOP	35	OPPOP	50	OPPOP
6	OPPOP	21	OPPOP	36	OPPOP	51	OPPOP
7	OPPOP	22	OPPOP	37	OPPOP	52	OPPOP
8	OPPOP	23	OPPOP	38	OPPOP	53	OPPOP
9	OPPOP	24	OPPOP	39	OPPOP	54	OPPOP
10	OPPOP	25	OPPOP	40	OPPOP	55	OPPOP
11	PPPOP	26	OPPOP	41	OPPOP	56	OPPOP
12	OPPOP	27	OPPOP	42	OPPOP	57	OPPOP
13	OPPOP	28	OPPOP	43	OPPOP	58	OPPOP
14	OPPOP	29	OPPOP	44	OPPOP	59	OPPOP

Figure 5 Genotype information

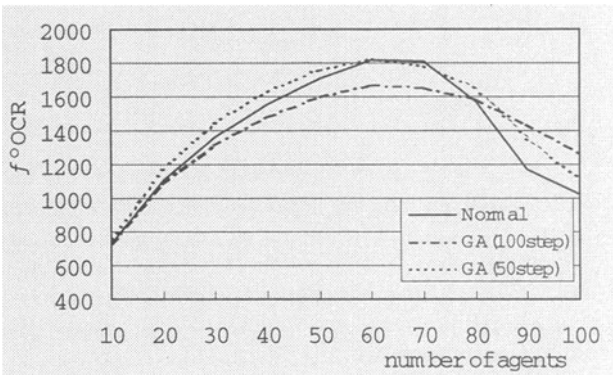


Figure 6 Robustness



Figure 7 Time span of evolution

Several interesting characteristics are pointed out about the evolutionary cooperative scheduling as follows:

i) General characteristic:

The evolutionary computation based on GA enhances the efficiency of cooperative scheduling despite its simple architecture. Work agents evolve gradually and offspring with elite chromosome are produced and increase in every generation(Figure 5).

ii) Fitness function:

Scheduling performance is always improved under the condition (Fitness=1), whereas the improvement is not always expected under the condition (Fitness=0). That is because the sensitivity of fitness function (Fitness=0) is too rough to converge the chromosomes. A detailed level of the sensitivity is essential to the fitness function and the function (Fitness=1) is reasonable in this scheduling model(Figure 4, Figure 5)

iii)Reproduction rule:

Elitism(Elite=1) increases the scheduling performance only if the sensitivity of the fitness function is secured (Figure 4). Elitism requires the sensitive fitness function to produce elite offspring by nature. The chromosomes aren't converged under rough fitness function (Fitness=0) in Elitism, because it is difficult to take the chromosomes in order.

iv) Robustness (Figure 6):

The evolutionary mechanism based on GA has been proved to enhance scheduling robustness under any population density. The appropriate convergence of chromosomes, which varies the experimental conditions, are obtained in the evolutionary scheduling paradigm.

v) Evolutionary frequency (Figure 7):

Scheduling robustness is relatively sensitive against evolutionary time span. Too short time span decreases the robustness, because the fitness value in each agent is

quite similar and that prevents appropriate convergence of chromosome. On the other hand, evolutionary effects on the agents is weak under too long time span.

Discussions

Our implementation and experiments successfully demonstrated that the mechanisms for conflict resolution that we derived from our concept were able to cooperate work agents in a systematic manner. These work agents proved capable of robustness and reliability coping with the scheduling demands of a simple environment, and seem to perform pretty well. The proposed scheduling paradigm is robust enough to treat general dynamic environmental changes in manufacturing system, such as priority work, despite its simple architecture. We found 'Leadership phenomenon' that is quite distinctive in the proposed scheduling. Evolutionary computation is effective to autonomous agents and improved scheduling performance up to 40%. Several characteristics of the evolutionary self-organization scheduling were clarified.

6 CONCLUSION

A new paradigm of evolutionary self organization scheduling using cooperative autonomous agents has been proposed in this paper. The architecture is quite simple but suitable to fully de-centralized scheduling system. The effectiveness of the proposed paradigm has been investigated by using simulation model. As the result, our work agents have proved capable of robustness and reliability coping with the several scheduling demands. Evolutionary algorithm has played an important role to enhance scheduling performance and robustness.

7 REFERENCES

- Ishida, T. (1995) Discussion on Agents, *Journal of Japan AI Society*, Vol.10 No.5, 663-667.
- Kaihara, T. and Imai, H.(1996a) A Scheduling Methodology for Large Scaled Job Shop Manufacturing Line, *Journal of the University of Marketing and Distribution Sciences -Information, Economics & Management Science-*; Vol.5, No.1, 99-108.
- Kaihara, T. (1996b) A Study on Multi Agent Scheduling, *Journal of the University of Marketing and Distribution Sciences -Information, Economics & Management Science-*; Vol.5, No.2, 113-120.
- Kaihara, T. and Fujii, S.(1997), A self-organization scheduling paradigm using coordinated autonomous agents, *Rapid Product Development*, Chapman & Hall, London, 489-498.
- Langton, C.G.(1986) Studying artificial life with cellular automata, *Physica D*, Vol.22, 120-149.
- Shoham, Y. (1993) Agent-Oriented Programming, *Artificial Intelligence.*, Vol.60, No.1, 51-92.
- Shone, H. (1984) The Spatial Orientation of Behaviour in Animals and Man, *Princeton Series in Neurobiology and Behaviour*, Princeton University Press.

8 BIOGRAPHY

Toshiya Kaihara is an Associate Professor at University of Marketing and Distribution Sciences. He received the B.S. and M.S. degrees in precision engineering from Kyoto University, Kyoto, Japan, in 1983 and 1985, respectively. He received the Ph.D. degree in mechanical engineering from Imperial College, University of London, London, UK, in 1994. He is a member of JSPE, JSME, IEEJ, IEEE, and others.

Susumu Fujii is a Professor of Computer and Systems Engineering at Kobe University, Kobe, Japan. He received a B.S. in Mechanical Engineering and M.S. in Precision Engineering from Kyoto University and Ph.D. in Mechanical Engineering from the University of Wisconsin, Madison. He is a member of various academic societies such as JSPE, JSME, ORSJ, ISCIE, SICE, INFORMS and others.

Satoshi Kunimasa is a graduate student of Computer and Systems Engineering at Kobe University, Kobe, Japan. He received a B.S. in Computer and Systems Engineering at Kobe University, Kobe, Japan in 1997. His current interests are on multi-agent paradigm, evolutionary computing and reinforce learning method.