

Research on Mechanism of Real-Time Mas Based Dynamic Intelligent Manufacturing Systems

Haihua ZHU, Yingcong WANG, Ming CHEN

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, E-mail: zhuhh@nuaa.edu.cn

crossref <http://dx.doi.org/10.5755/j01.mech.24.1.18631>

1. Introduction

Scheduling on shop floor layer is fundamental in the controlling of the manufacturing system, which directly influences the efficiency of the whole manufacturing system. The single massive production mode in traditional manufacturing companies reduces the complexity of the scheduling problems, which enables traditional static (offline) to achieve acceptable scheduling result. However, the weakness of static scheduling gradually appears when the complexity of manufacturing system, the sorts of the products and the uncertain factors in the manufacturing process increase [1]. For one thing, high stability will be required in the static scheduling problems of the manufacturing with respect to its structure, running, and orders. For the other thing, this sort of hierarchical rigid structure often restricts the ability of self-adaptability and reconstruction required when production environment varies dynamically [2]. It can be seen that traditional static scheduling mode can hardly adapt to current dynamic environment nowadays.

Therefore, it has been one of the most significant research to look for a scheduling mechanism in manufacturing system with responsiveness, real time performance and robustness. In the past years, new-born dynamic scheduling methods have attracted many relevant scholars' attention. Shen [3] mainly concentrates on multi-objective dynamic schedule, Freitag's [4] work is about scheduling algorithms in large-scale and multi-restrict manufacturing system, Zhang [5] combines genetic algorithm with taboo search to solve dynamic scheduling problems. However, there exists a shortcoming that the authors above mostly use a heuristic search algorithm with long calculating time, high coupling degree which has along calculating time and is hard to deploy.

Different with algorithms mentioned above, multi-agent system (MAS) completes complex tasks with cooperation between sub-cells [6], which can improve system's flexibility and re-configurability. Multi-agent system is a distributed artificial intelligent model. In this model, single Agent's perception is limited, but with communication, they can reach out large-scale overall goals in an interactive way [7].

Many scholars make progress in their research of MAS. Brennan [8] raises a reconfigurable control rule which is based on the agent. Zaeh [9] researches on a system control method based on perceive agent. Colombo [10] builds an intelligent platform based on agent.

However, these applications mostly centralize on the upper system control of the manufacturing system, while researches on bottom system are relatively lesser.

This paper proposes a dynamic real-time event-triggered based scheduling mechanism. This mechanism is aimed at the purpose of realizing real-time control and dynamic schedule, which can make decision according to the system's running state. Meanwhile, this mechanism compatible with the hierarchical hardware and software in the manufacturing system, which is convenient for system to reconfigure dynamically.

2. Problem description

This paper regards the situation that the multiple processing cells in manufacturing system are arranged at a certain position. Each processing cell can carry out one or more turning, milling, planning, and grinding processes. Output buffers, output buffers, processing zones with limited size and a manipulator makes up a manufacturing cell. The processing zone can only hold a job per time. In the whole system, there exists an Automated Storage and Retrieval System (AS/RS), which is used to deposit raw materials or end products. In AS/RS only one job can be delivered at one time. Between AS/RS and manufacturing cells, there are one-way roads that connect to each other, which make up of a fully connected directed graph. Automated Guided Vehicle (AGV) is used to transit jobs between every single facility. As a result of the limit of power, we set up several charging points where AGV will go to when it is out of power. And while being charged, AGV will not response to system's scheduling request.

For the convenience of our research, this paper makes such assumptions to restrict this real-time Internet of Things (IoT) -based manufacturing system :

1. All orders use the same material;
2. We have already been aware of the manufacturing procedures (including processing sequences and processing methods), processing time and transport time of AGV between any two facilities;
3. The time of conveying and clamping can be negligible;
4. The jobs in input buffer are processed in a first-in first-out sequence, while those in output buffer can leave in any sequence;
5. There are only three kind of disturbing events that rush orders, breakdown in processing equipment and low-power states of AGV will be considered, so time error accumulations between any two scheduling will not trigger the reschedule.

3. A real-time dynamic schedule in manufacturing system

3.1. Event-trigger based rolling horizon dynamic scheduling strategy

There are two main scheduling strategies in manufacturing systems, dynamic schedule and static schedule respectively. Static schedule means before manufacturing systems is running, all orders are scheduled by specific scheduling algorithms in advance. However, when it goes to a practical production problem, everything will be different. Sometimes, factors such as urgent orders and equipment failure are ignored, the production will be at a standstill. In a dynamic scheduling situation, production status could be monitored in real time while the manufacturing system is running. Once disturbing event occurred, system could take action to deal with it immediately. Effectively handling disturbing events, dynamic scheduling strategy has been the major scheduling mechanism in current manufacturing system.

There are three main dynamic scheduling models: artificial intelligence (expert system), simulation experiment and rolling horizon. Expert system is a model that first set up a decision-making knowledge base, then will search the knowledge base for corresponding algorithms, strategies to generate dynamic scheduling solutions. However, the outcome of this scheduling model highly depends on whether there is an effective, all-round scheduling knowledge base whose developing cost and debugging cost are high. Simulation experiment is a model to simulate all kinds of situations in the production process of discrete-event. But this model is complicated and has a long development cycle. Taking the thought of greedy algorithm, rolling horizon divide global scheduling space into several subspaces to make sure the outcome of every subspace is a local optimum. So, it is very suitable to use this model to operate dynamic schedule in an intelligent manufacturing system and to ensure the system can real-time responses to all kinds of disturbance events.

To summarize, this paper uses an event-trigger based rolling horizon dynamic scheduling strategy, divides dynamic events in manufacturing system into basic events (predictable events during system's operation, such as the completion of the process and the completion of the transit) and disturbing events (unforeseen events of urgent orders, equipment failure, etc.). As is shown in Figure 1, after split into procedures, original orders get into the task set. The rolling horizon moves and puts procedures of each orders into MAS as a scheduling task. In MAS, every agent negotiates with each other through communication network to generate local scheduling schemes. When there is a disturbing event, MAS will immediately start a reschedule in response to the event. In this model, the entire scheduling object of the system will be split into several sub-objects that are calculated by multiple distributed agents in a synergistic way. Owing to this model, manufacturing systems are able to generate rescheduling schemes in response to disturbing events in a few cpu time. With regard to a single production task, if the system is in a good condition, then when basic events trigger, it will only schedule its next transit and process procedure. But when there appear some disturbing events,

then the system will interrupt current task and reschedule immediately. For the entire manufacturing system, each agent is able to communicate and negotiate with other ones to form a dynamic scheduling scheme for the manufacturing system. Not only the responsiveness to environment, but also the completeness of the algorithms when transiting and machining is ensured by this mechanism.

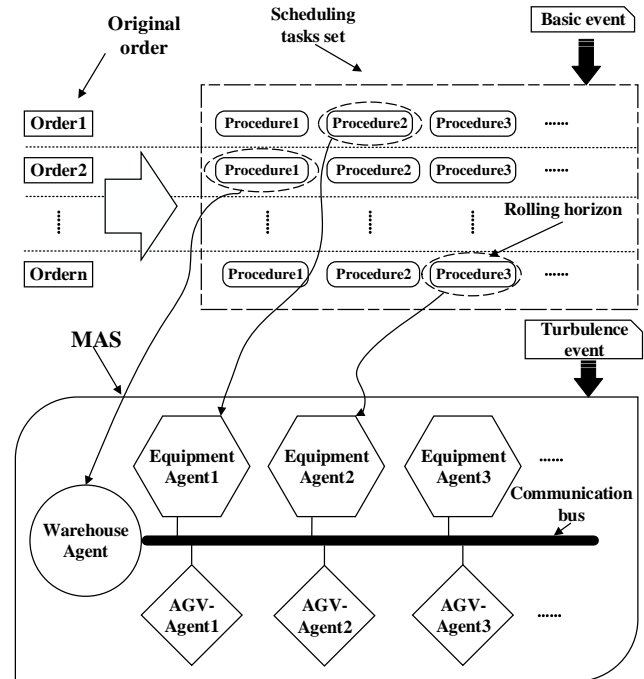


Fig. 1 Sketch of Event-trigger based rolling horizon dynamic scheduling strategy

3.2. Contract net based multi-agent cooperative manipulating algorithm

Dynamic scheduling mechanism mentioned above schedules only with specific process in single productive task as its fundamental unit. For scheduling procedures are limited by mechanism's own knowledge base, we don't know whether or not this mechanism may ensure the global optimality of the system. Moreover, different scheduling schemes generated by different tasks may affect with each other due to the limit of the resources. (E.g. Multiple schemes may require a same processing equipment.) Therefore, when dynamic schedules are carrying on, we need corresponding mechanisms to negotiate agents involved in scheduling schemes.

Contract net mechanism is usually used to solve problems for distributed systems. This method is simple, efficient and practical, so it is widely applied to negotiation in multi-agent systems. This paper designed an MAS negotiating mechanism designed by contract net mechanism. Fig. 2 shows the process in which agents of entities use improved contract net to negotiate with each other when they are scheduled from AS/RS to processing cells. (In this figure, dashed line means another option for current Agent unit to implement.)

The steps based on improved contract net are as follows:

Step 1. At each scheduling time, $A_{AS/RS}$ and A_M with jobs to be processed become initiator agents.

Step 2. Each initiator chooses a job's next procedure from its own job queue as a task and broadcasts the bidding information to all other participant agents (Other $A_{AS/RS}$, A_M and A_{AGV}) in the system.

Step 3. Any participant agent who is able to take

the task will participate in the bid by sending package including information such as its own state, current tasks and cache tasks back to the initiator. Those without the ability to take the task will instead send back a refuse information.

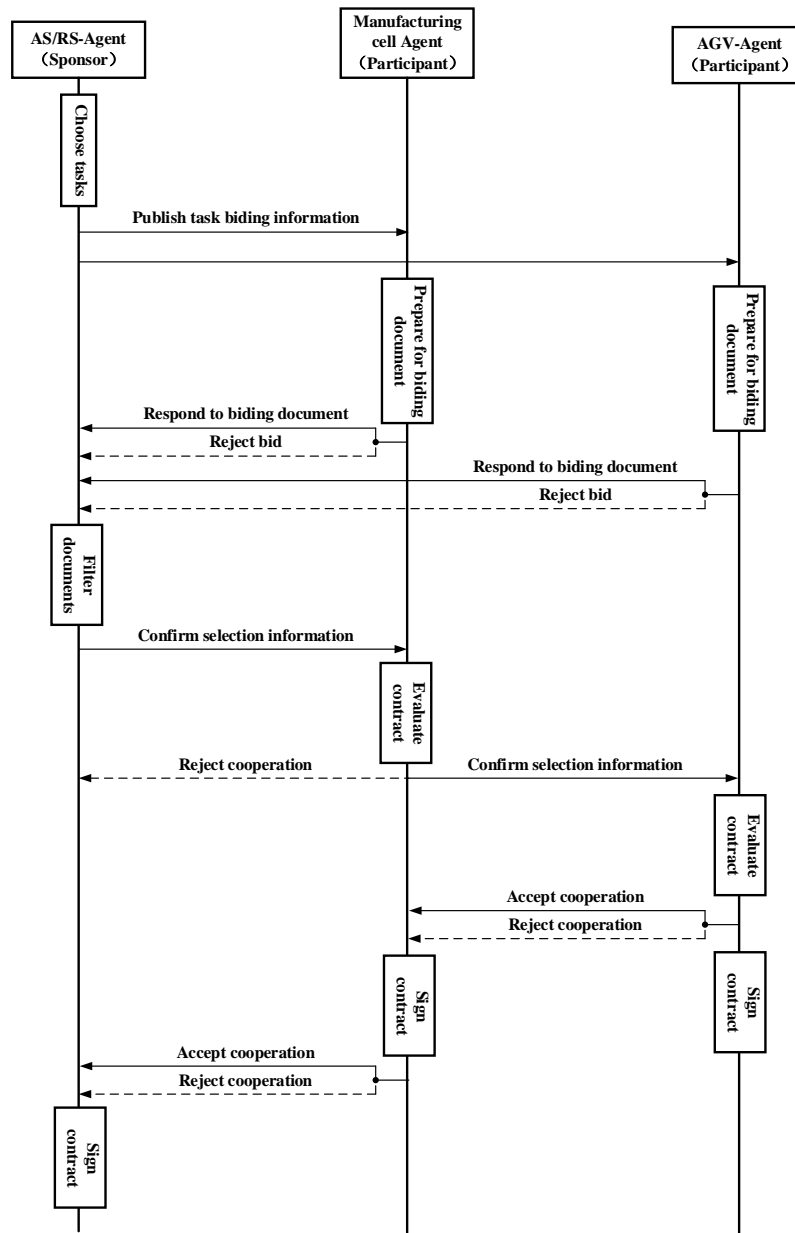


Fig. 2 Mechanism of distributed multi-agent negotiated control

Step 4. Initiator agents combine n_{AGV} number of A_{AGV} s with n_M number of A_M s (For in storage tasks, there will be $n_{AS/RS}$ number of $A_{AS/RS}$) to generate $n_{AGV} \cdot n_M$ (or $n_{AS/RS} \cdot n_M$) candidate sets.

Step 5. Initiator agents call evaluate functions to evaluate candidate groups and compare the evaluating outcomes with corresponding agent's rating value of cache tasks. After that, those with higher evaluating outcomes than corresponding task rating value will be chosen.

Step 6. Arrange the filtered combinations in order of the evaluating outcomes. Then send information including evaluation values, procedures to the optimal one in candidate set to confirm selection.

Step 7. After the participant A_M (or $A_{AS/RS}$) receives information from initiator agent, decide whether to accept take the contract according to its own state. If refuses, then it will reject the contract.

Step 8. After participant A_{AGV} receives the confirm information, it will also decide whether or not to take the task and sign the contract according to its current states and respond to corresponding participant A_M (or $A_{AS/RS}$). If the contract is signed, participant A_{AGV} 's cache contract will be replaced by the task in the contract.

Step 9. Participants A_M s (or $A_{AS/RS}$) sign their own contracts according to whether their corresponding participants A_{AGV} s sign their contracts. If contract is

signed, A_{MS} (or $A_{AS/RS}$) will respond the decision to initiator agent. If the contract is signed, participant A_{AGV} 's cache task will be replaced by the current task in the contract.

Step 10. If the information sent back is that contract is signed, the negotiating process of contract net will be completed. Otherwise, delete groups including participants that reject the contract, turn to Step6 and choose other candidate groups until the candidate set is empty to complete this negotiating process.

Step 11. After negotiating process of current procedure is completed, initiator chooses other jobs in job queue, turn to Step2 and negotiates contract net until the queue is empty to complete current schedule.

There are some tips to be underlined and illustrated further towards the negotiation in the process of improved contract network:

1. In contract network, consider the “taking-transit-machining” process as the basic unit of scheduling task of procedure. Each A_M (or $A_{AS/RS}$) of corresponding cell will be negotiated in a distributed parallel scheduling way.

2. A_M and $A_{AS/RS}$ can not only be initiator agents, but also participant agents. Jobs that A_{AGV} contains are all in the process of task execution and are all not jobs to be processed. So, A_{AGV} only serves as participant.

3. The processes of negotiation and execution of each agent are operated in a multiple thread parallel way, which increases the system's operating efficiency.

4. Sponsor agent will negotiate on the basis of principle “Bidding simultaneously, confirming step-by-step”. Compared with the way in which transit process and machining process bid individually, sponsor agent can evaluate the bidding combinations to reduce the traffic of invalid bidding combination.

5. In the case of no influence from distributions, cache tasks in agent cell can only be replaced by other tasks with better outcomes at scheduling points. But if the evaluating outcome of the new task is equal to that of the cache tasks, the replacement will not take place. So, sponsor agent is able to sift out the candidate combination set by comparing the evaluating outcome of the candidate bidding combination and those of corresponding cache task in the bidding combination.

6. In the multi-agent system, multiple interactive negotiating processes run simultaneously. At the same scheduling point, the entity state of the agent cell will not change and cache task could be replaced repeatedly. So, participant will decide whether or not to sign the contract according to the latest cache task when it receives confirming message.

7. When some participant agent's cache task is replaced, it will confirm cache task's sponsor and another participant to cancel the task. Sponsor agent will pick other combination from the combination set and will renegotiate the contract network. Meanwhile, the participant agent will broadcast the message that its task's has been canceled. And once other sponsor agents receive the information, it will add this participant's combination to the candidate set of corresponding task and then will renegotiate. Regarding one certain task, if the new combination has been including in other contract, sponsor

agent will confirm corresponding participant agents to cancel the contract, and it will renegotiate.

8. At the same scheduling point, the physical state of the system will not change. So, the evaluating outcomes of all combinations in each procedure will not change. By saving these outcomes, the system can avoid collecting bid documents redundantly when renegotiating the task.

Improved contract network makes sure that the system will be able to get to the optimal in current state after multiple negotiations by replacing cache task. It optimizes the negotiation by “bidding simultaneously”, “comparing cache tasks” and “keeping the evaluating outcomes”.

3.3. Real-time dynamic scheduling control algorithms

Considering a dynamic control system, its basic goals are keeping production system stable and continuous and reducing the waiting time of equipment. Ideally, there are two aspects following that reflect system's goal of continuous running. For equipment agent, we would like jobs to be transited to AGV immediately after machining task has completed to make sure it can receive tasks from other jobs to realize continuous manufacturing. For AGV-Agent, we wish that jobs would be processed after transited to object equipment to make sure it can receive tasks from other jobs to realize continuous transit. So, suitable scheduling rules must be designed to estimate bidding agent combinations in order to effectively connect machining processes to transit process of each jobs.

In this paper, we take three different processes picking up, transit and machining as one independent scheduling subsection, namely the shortest transport time. Its time consumption of picking up and transit is the shortest. It can assign rules and evaluate bidding agents.

In the picking up process, the shortest time consumption should be the longer one between the time that AGV takes when going to the cell where a job is located at and the rest time consumption of the job in the current cell. So, the expected time $T(p_x, p_y)$, that is the time consumption of picking up, can be calculated in such a formula:

$$T(P_{A_j}, P_{O_i}) = \max\{(L(A_j) + t(P_{A_j}, P_{O_i})), L(O_i)\}, \quad (1)$$

where: PA_j and Po_i mean the locations where AGV_j is and where job_i is respectively. Function $t(P_x, P_y)$ means the transport time while function $L(x)$ means the rest time of current task in cell x .

In the total process of picking up and transiting, the shortest time consumption will be the longer one between the time that AGV takes when going to the object cell and the rest time consumption of the task in the object cell. So, the expected time $T(P_x, P_y, P_z)$, that is the total time consumption of transiting, can be calculated in such a formula:

$$\begin{aligned} T(P_{A_j}, P_{O_i}, P_{M_k}) &= \\ &= \max\{T(P_{A_j}, P_{O_i}) + t(P_{O_i}, P_{M_k}), L(M_k)\}, \end{aligned} \quad (2)$$

where: P_{Mk} means where object cell k is.

The agent which releases tasks will choose an available bidding agent combination with a minimum transport time to hand out a task in order to maximize utilization rate.

4. Simulation to justify

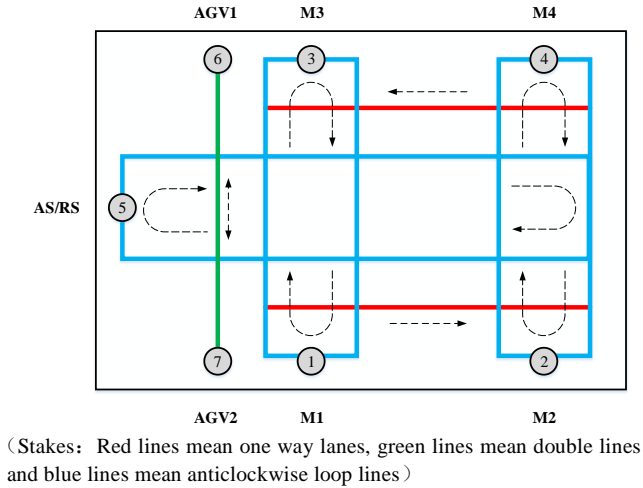


Fig. 3 Arrangement of the simulation platform

To verify the performance of the control system and the responsiveness to disturbance event, this paper constructs corresponding simulation platform to test, the arrangement is as shown in Fig. 6. In the system, spot 1, 2, 3, 4 mean the location of M1, M2, M3, M4 respectively; Spot 5 is the location of AS/RS; and spot 6, 7 mean the initial point of stop. Red lines mean one way lanes, green lines mean double lines and blue lines mean anticlockwise loop lines. The direction of traffic is shown in the figure as dashed lines. Between every two points, there is only one path. The transport time of AGV between each point is shown in Table 1. A manufacturing cell possesses a public cache shared with in and out storage and a processing position that is each manufacturing cell possesses at most two jobs at the same point. The processing capacity of each cell is shown in Table 2. The procedure and consumption time of each job are shown in Table 3.

Fig.4 shows the Gantt chart of the system when it is running normally, the total time consumption is 157 seconds.

There are two simulating experiments of scheduling designed in this paper. The first experiment is influenced by turbulence of equipment failure that manufacturing cell M3 will arise failure at 60 s and it will take 30s to fix this failure. The second one is to simulate AGV's low power turbulence. We set AGV1 to arise a low-power alarm when it is at 60s, then it need to return to initial point to be charged, and it will cost 30 s. The experiment's results are shown in Fig. 5 and Fig. 6 respectively.

In the turbulence experiment of manufacturing cell, M3 will encounter a failure at 60 s. After failure, all turning processes will be transited to M4. The final total schedule of this system takes 188 s. And in the turbulence experiment of AGV low power, AGV1 will get a mistake at 60 s. At this moment, AGV1 is still taking the

transmit task from order4, and it needs to be charged after finishing the current task. As the red slash block is shown in Fig. 9, AGV1 will return to the initial point immediately and to be charged at 83s. When AGV is being charged, all transit tasks will be assigned to AGV2. The final total schedule of this system takes 203seconds.

The results of the experiments show that this IoT-based manufacturing system can realize the dynamicity and ensure the stability of the system. From that, we can see that this is a practical system.

Table 1 Transport time (unit: second)

Starting point	End destination						
	P1	P2	P3	P4	P5	P6	P7
P1	N/A	10	8	12	6	8	6
P2	10	N/A	12	8	10	12	10
P3	8	12	N/A	8	10	12	10
P4	12	8	10	N/A	14	16	14
P5	10	14	6	10	N/A	4	6
P6	10	14	6	10	6	N/A	N/A
P7	12	16	8	14	4	N/A	N/A

Table 2

Capability of each cell

Procedure	Capability
M1	mill
M2	drill
M3	turn
M4	turn, grind

Table 3

The table of procedures (unit: second)

Order of jobs	Arrival time	Order of procedures	Manufacturing methods	Machining time (s)
J1	0	1	Out storage	5
		2	Turn	12
		3	Drill	6
		4	In storage	5
J2	0	1	Out storage	5
		2	Mill	16
		3	Grind	20
		4	Drill	12
		5	In storage	5
J3	0	1	Out storage	5
		2	Mill	14
		3	Grind	8
		4	In storage	5
J4	10	1	Our storage	5
		2	Turn	14
		3	In storage	5
J5	20	1	Out storage	5
		2	Turn	6
		3	Mill	16
		4	Grind	8
		5	In storage	5
J6	30	1	Out storage	5
		2	Turn	12
		3	Mill	4
		4	Drill	2
		5	In storage	5

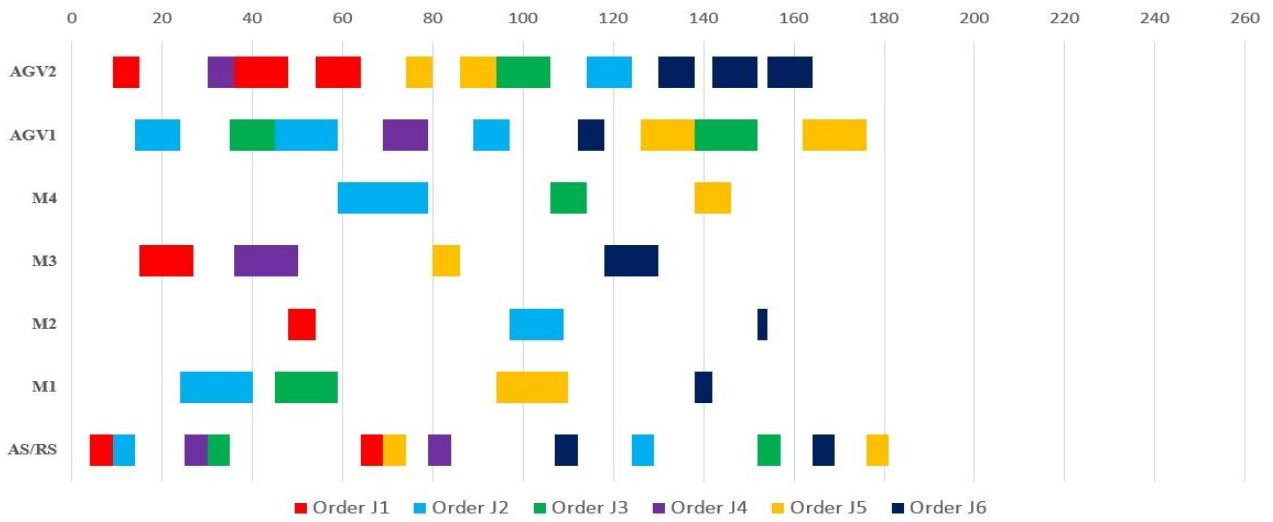
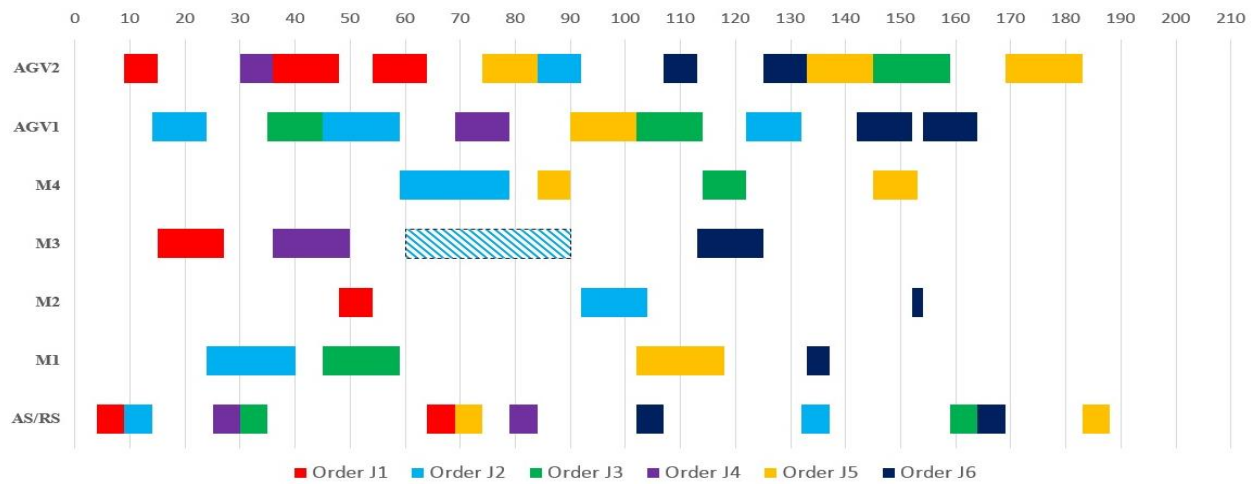
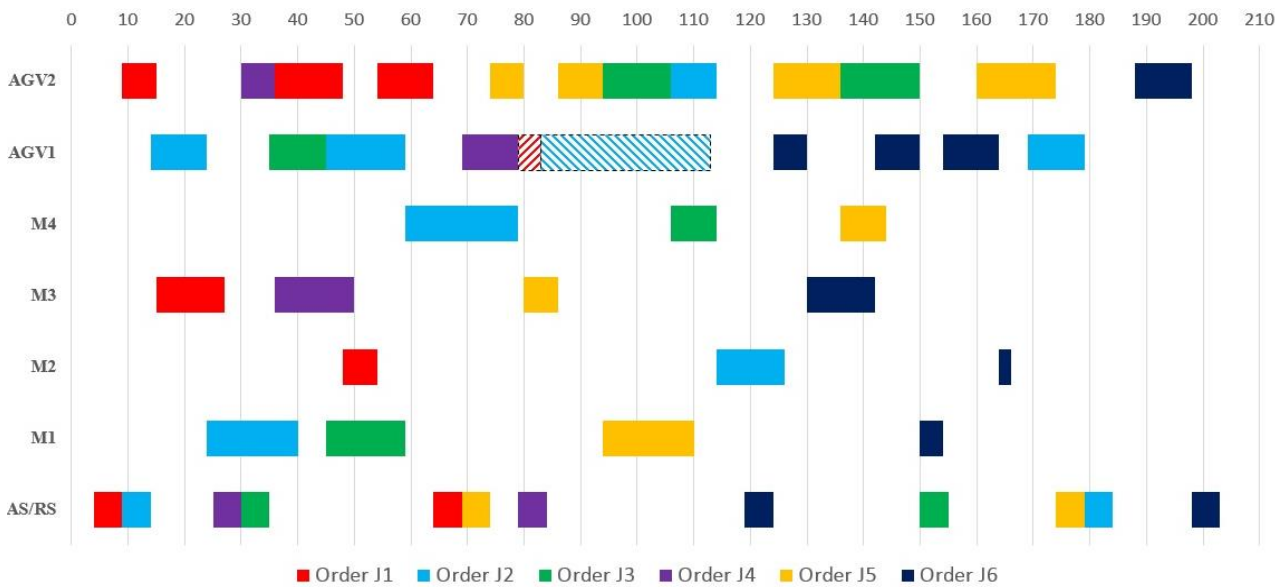


Fig. 4 The Gantt chart with no turbulence



(Stakes: the blue slash block represents the repair time of the manufacturing cell)

Fig. 5 The Gantt chart with turbulence of manufacturing cell



(Stakes: the red slash block represents the time of returning initial stopping point and the blue slash block represents the charging time of the AGV)

Fig. 6 The Gantt chart with turbulence of AGV low power

5. Conclusions

A model of intelligent control manufacturing system is proposed in this paper. According to the real time state in manufacturing shop floor, this system can realize a distributed control and a dynamic schedule based on MAS. We build a dynamic real-time control model of manufacturing system, and explain the rolling horizon mechanism, the MAS negotiation mechanism based on contract network and the real-time dynamic control algorithms. At last, this experiment is proven to get a good anti-disturbance ability. Some Real-time scheduling algorithm will be taken into consideration in the further work.

Acknowledgements

This research is supported by National Nature Science Foundation of China under Grant U1637211, and Qing Lan Project.

References

1. **Scholz-Reiter, B.; Freitag, M.** 2007. Autonomous processes in assembly systems, *Annals of the CIRP*, 56:712–729.
<http://dx.doi.org/10.1016/j.cirp.2007.10.002>.
2. **Frazzon, E.M.; Hartmann, J.; Makuschewitz, T.** 2013. Towards socio-cyber-physical systems in production networks, *Procedia Cirp*, 7(5): 49–54.
<http://dx.doi.org/10.1016/j.procir.2013.05.009>.
3. **Shen, X. N.; Yao, X.** 2015. Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems, *J. Information Sciences*, 298: 198-224.
<https://doi.org/10.1016/j.ins.2014.11.036>.
4. **Freitag, M.; Hildebrandt, T.** 2016. Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization, *CIRP Annals-Manufacturing Technology*, 65(1):433-436.
<http://dx.doi.org/10.1016/j.cirp.2016.04.066>.
5. **Zhang, L.; Gao, L.; Li, X.** 2013. A hybrid genetic algorithm and taboo search for a multi-objective dynamic job shop scheduling problem, *International Journal of Production Research*, 51(12): 3516-3531.
<http://dx.doi.org/10.1080/00207543.2012.751509>.
6. **Xiang, W.; Lee, H.P.** 2008. Ant colony intelligence in multi-agent dynamic manufacturing scheduling, *Engineering Applications of Artificial Intelligence*, 21(1):73-85.
<http://dx.doi.org/10.1016/j.engappai.2007.03.008>.
7. **Brennan, R.W.; Fletcher, M.; Norrie, D.H.** 2002. An agent-based approach to reconfiguration of real-time distributed control systems, *IEEE transactions on Robotics and Automation*, 8(4): 444-451.
<http://dx.doi.org/10.1109/TRA.2002.802211>.
8. **Brennan, R.W.; Fletcher, M.; Norrie, D.H.** 2002. An agent-based approach to reconfiguration of real-time distributed control systems, *IEEE Trans. Robot. Autom*, 18(4): 444–451.
<http://dx.doi.org/10.1109/TRA.2002.802211>.
9. **Zaeh, M.F.; Reinhart, G.; Ostgathe, M.; Geiger, F.; Lau, C.** 2010. A holistic approach for the cognitive control of production systems, *Advanced Engineering Informatics*, 24: 300–307.
<https://doi.org/10.1016/j.aei.2010.05.014>.
10. **Colombo, A.W.; Schoop, R.; Neubert, R.** 2006. An agent-based intelligent control platform for industrial holonic manufacturing systems, *IEEE Transactions on Industrial Electronics*, 53(1):322-337.
<http://dx.doi.org/10.1109/TIE.2005.862210>.

Haihua Zhu, Yingcong Wang, Ming Chen

RESEARCH ON MECHANISM OF REAL-TIME
MAS BASED DYNAMIC INTELLIGENT
MANUFACTURING SYSTEMS

S u m m a r y

With the improvement of consumption concept, customer's demand is becoming more and more dynamic, diversified and personalized, which requires the manufacturing system to be more and more robust and real-time. To achieve such purpose, in this paper, a hierarchical heterogeneous hybrid multi-agent system (MAS) based control architecture is proposed, a MAS-based shop floor scheduling model is designed, an event-driven rolling horizon scheduling mechanism is selected as the scheduling mechanism of the system and an improved contract network protocol is proposed to act as the system's scheduling algorithm. According to the simulation experiments based on the ever built experimental platform, Internet of Things (IoT) based manufacturing real-time scheduling system is proven to be feasible.

Keywords: IoT-based manufacturing, real-time scheduling, MAS, rolling horizon, contract network.

Received July 21, 2017

Accepted February 15, 2018