A Basic Study on Highly Distributed Production Scheduling

Eiji Morinaga, Eiji Arai, Hidefumi Wakamatsu

A Basic Study on Highly Distributed Production Scheduling

Eiji Morinaga, Eiji Arai and Hidefumi Wakamatsu

Division of Materials and Manufacturing Science, Osaka University
{morinaga,arai,wakamatu}@mapse.eng.osaka-u.ac.jp

Abstract. Recent manufacturing systems are required to be flexible to cope with variable situation. This requirement has driven development of distributed methods of production simulation and scheduling. Recent advances in computer network technology is achieving *highly distributed manufacturing systems (HDMSs)* where each facility is computerized and manages itself autonomously by communicating with other facilities. A distributed simulation method for HDMSs was proposed. To take the full advantage of this method, scheduling problem should be also discussed. Conventional distributed scheduling methods would be inappropriate for HDMSs, since some elements perform processes for information integration and decision making and are therefore subject to heavy computational load when those methods are applied to HDMSs. This paper proposes a distributed scheduling method based on a dispatching rule where the processes for decision making are not performed by any elements but indirectly by a communication protocol.

Keywords: highly distributed manufacturing systems, distributed production scheduling, dispatching rules, agile manufacturing

1 Introduction

Production scheduling is a key issue in realizing sophisticated manufacturing system management/operation, which is indispensable to achieve competitive manufacturing. Numerous studies have been conducted in this area for a long time in various approaches such as mathematical programming formulation[1, 2], artificial intelligence[3, 4], Petri nets[5] and so on. In these activities, the mainstream has been the centralized approach where the whole manufacturing system is modeled and then the optimal or a suboptimal schedule for the whole system is searched. To evaluate feasibility of the schedule in detail, production simulation is required to be performed based on the schedule. The centralized approach has been the mainstream also in production simulation area.

Recent diversified customers' needs have promoted high-mix low-volume manufacturing, and then manufacturing systems should be flexible to cope with variable situation. This requires production simulation/scheduling to be performed immediately when the situation has changed. The centralized approach is inappropriate from this point of view, since re-modeling the whole system under the

new situation involves a heavy load. It is desirable to take the distributed approach in which the simulation/scheduling is performed by a set of local models and coordination among them. Recent development of computer network technologies is achieving *highly distributed manufacturing systems (HDMSs)*, that is, makes it achievable to equip each facility with an independent simulation model and run the models on a network so that they can communicate with each other. Production simulation for HDMSs, termed as *highly distributed (HD) simulation*, has been discussed. A framework and a time management method for this simulation with a set of models based on a master schedule was proposed[7].

To make this method beneficial enough, distributed production scheduling is also required for modification of the master schedule based on the simulation result. There have been many different kinds of distributed scheduling methods, such as auction[6] and active database scheduling. However, these methods would be inappropriate for HDMSs, since some elements such as the launcher and the database perform information integration and decision making which result in overload of the elements. Furthermore, it is desirable to carry out the scheduling in the similar framework to the HD simulation. In this paper, such type of distributed scheduling, termed as *highly distributed (HD) scheduling*[8], is discussed. This paper considers scheduling by the shortest processing time (SPT) dispatching rule, and proposes a communication algorithm for dispatching based on the architecture and mechanism of the HD simulation.

2 Highly Distributed Simulation

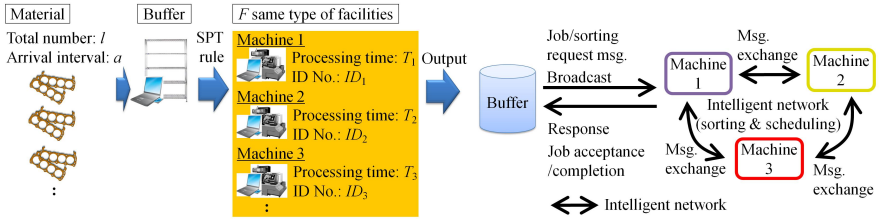
HD simulation[7] is carried out by a collection of autonomous on-line facility models brought together to respond to dynamic manufacturing system changes. They offer the potential of increased responsiveness and robustness compared to centralized methods. In this simulation method, each facility is equipped with a computer installed an identical model with its own data and event list, while all the computers are connected by network. The entire simulation runs by message exchanges between the models, in which each computer represents a different facility independently. By these means, engineers can save time of building models and, in addition, simulation architecture or model layout can be changed flexibly according to the real situation. As to the time management aspect, each computer manages their own data (processing time of jobs, current condition, etc) and job event list (which record the starting time of each job and the total number of jobs processed on this facility). Simulation sequence is decided by sorting of the event times within the network.

3 Highly Distributed Production Scheduling

This section describes a method of distributed scheduling based on the framework and mechanism of the HD simulation, which we call HD scheduling. The rationale behind this approach is to provide the function to modify the original master schedule immediately based on the result of the HD simulation. As a

preliminary research, this paper describes a scheduling method using the SPT dispatching rule within the concept of the HD scheduling.

This research deals with a simple manufacturing scenario shown in Fig. 1(a). Materials are transported to an area one-by-one at time period a. The area is composed of a buffer and multiple machines which can do the same process but have different performance. The material arrived is assigned by the buffer to one of the machines according to the SPT dispatching rule. In the conventional method using auction technique or active databases, decisions of dispatching are made by the launcher or the database; while in the method described below, these decisions are made by message exchanges among the machines (Fig. 1(b)).



(a) Considered scenario. (b) Message exchanges.

Fig. 1. Architecture of HD scheduling

The computer of each machine has a model of the real machine and its own processing time. At first, the models of the machines are sorted by the processing times. This phase called initial sorting is carried out by the processing flow shown in Fig. 2. Each message consists of each machine's processing time T_i, ID number ID_i (e.g. IP address in the network) and message type (in this case, 0 means that the message is sent for initial sorting). After receiving another's message, each computer analyses its message type. If its value is 0, the ith computer compares the processing time T_j with its own processing time T_i. If T_i is larger than T_j, the priority sequence number S_i (at the beginning, its value has been initialized to 1) is added by one, which means the priority decreases. If T_i is smaller than T_j, S_i remains unchanged, which results in a higher priority. If T_i equals to T_j, ID_i is compared to ID_j and then the priority is updated in the same manner. This comparison goes through every machine. When this sorting phase finished, the initial sequence of the machines for scheduling has been decided.

After the initial sorting phase, the dispatching phase starts. First of all, the buffer sends a request for processing the arrived material with its current time stamp WC to all of the machines, as shown in Fig. 3. This time the message type is 1. After sending the request, the buffer updates quantity of arrived materials Q, whose initial value is 0. Then the buffer waits for the response from all the machines and the clock of the buffer WC proceeds by one time scale a at which the next material arrives.

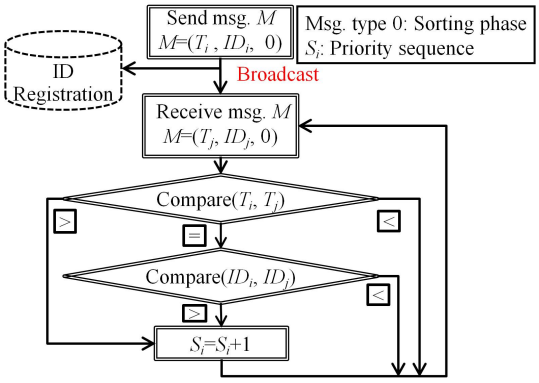


Fig. 2. Initial sorting flow

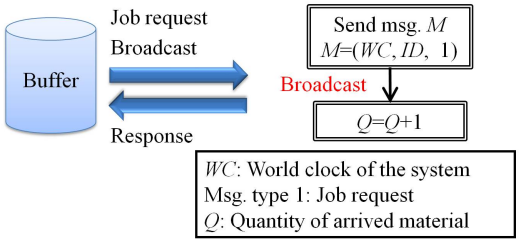


Fig. 3. Processing request from buffer

When receiving the request from the buffer, the machine having the highest priority (i.e. $S_i = 1$) sends an acceptance message back unless it is unavailable due to processing another material. Fig. 4 shows the flowchart of machines' reactions to the request. At first, each machine saves the initial priority sequence number S_i in B_i (the reason for introducing this variable will be explained later). After receiving a request message from the buffer, S_i is checked whether it equals to 1. If this is false, then the computer replies to the buffer with message type 6 which means this machine is unavailable to process the material. If S_i equals to 1, then $Flag$ is checked whether it equals to 0 which means the machine is idle and therefore available to process the material. If the $Flag$ is 1 which means the machine is working now, the computer replies to the buffer with message type 6, waits until $T_iE \geq WC \geq T_iE - a$ holds (the time at which the current processing job will finish after one time period), broadcasts its initial priority sequence number B_i and ID number ID_i with message type 3 for re-sorting among the machines, and then turns $Flag$ into 0. If $Flag$ equals to 0, the computer broadcasts B_i and ID_i with message type 2. After that this machine starts the processing, and the current time is substituted to the start time and the finished time is set to the start time added by the processing time T_i.

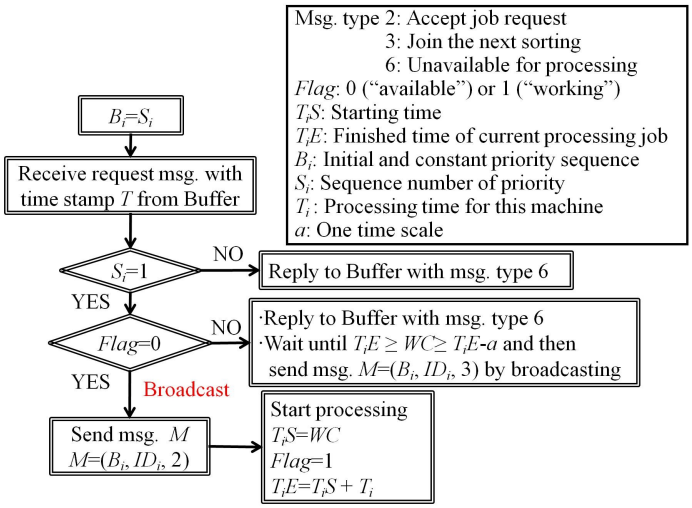


Fig. 4. Process flow when request message is received

In this dispatching phase, a lot of messages are broadcasted by machines. After receiving these messages, the buffer has different reaction according to the message type as shown in Fig. 5. If the message type is 2 or 6, the buffer waits until all the machines send replies (the number of replies equal to total number of the machines), increases the world clock WC by one time scale a, and then checks whether there is still any materials for processing (whether quantity of arrived materials is smaller than the total number of materials l, which has been set at the beginning of scheduling). If there is still a coming material, the buffer sends a processing request by message type 1 and increase Q by 1. If there is no more material, then the buffer sends a message of type 8 which means it is out of material now. If the message type is 7, which means one processing job has finished, then the total number of finished processing jobs N is increased by 1. After that, it is checked whether N is equal to l. If this is true, which means all the material has been processed, the buffer broadcasts a message of type 4 to notify every machine, and then this scheduling is finished.

After receiving these broadcasted messages, machines also have different operations according to the message type, as shown in Fig. 6. If the message type is 2 meaning another machine accepted the job request and started processing, the machine checks whether current value of S_i equals to 1. If this is false, the machine decreases S_i by 1 to increase its own priority sequence. By this mean, machines update their priority sequence and get chance to process a material. If the message type is 3, which means another machine joins the re-sorting, the machine compares the sender's B_j with its own B_i. If B_i is smaller than B_j, then

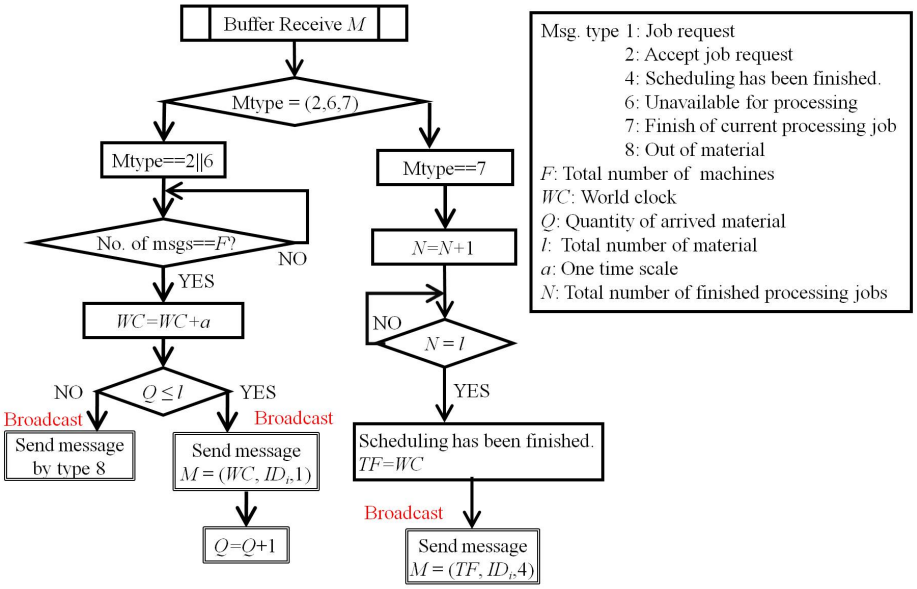


Fig. 5. Buffer's operations according to received message type

the machine decreases the priority by subtracting 1 from S_i and send message type5 with P set as 0. If B_i is larger than B_j, then the priority sequence remains the same and the machine sends a message of type 5 with P set to 1 (to lower the rival machine's priority). If the message type is 4, which means the simulation for scheduling has finished, this machine stops running. If the machine receives a message of type 5, which is the result of re-sorting, then it updates its priority sequence and set $Flag$ to 0 (now is available for processing), waits until WC turns to T_iE (the finished time of the current processing job) and sends a message of type 7 to the buffer notifying one material has been processed. If the message type is 8, which means there is no more material for processing, the machine replys to the buffer by message type 6 (not available for processing).

These flows were applied to the scheduling problem shown in Fig. 1(a) with $l = 10$, $a = 1$, $T_1 = 4$, $T_2 = 2$ and $T_3 = 3$. The SPT rule brings about the schedule shown in Fig. 7. The flows were implemented on a computer, where the models communicate with each other by the local network in the computer, via coding with the C# language. Executing this code, the same schedule was successfully output, which shows fundamental feasibility of the proposed method.

4 Conclusion

Recent manufacturing systems are required to be flexible to cope with variable situation. This requires production simulation/scheduling to be performed im-

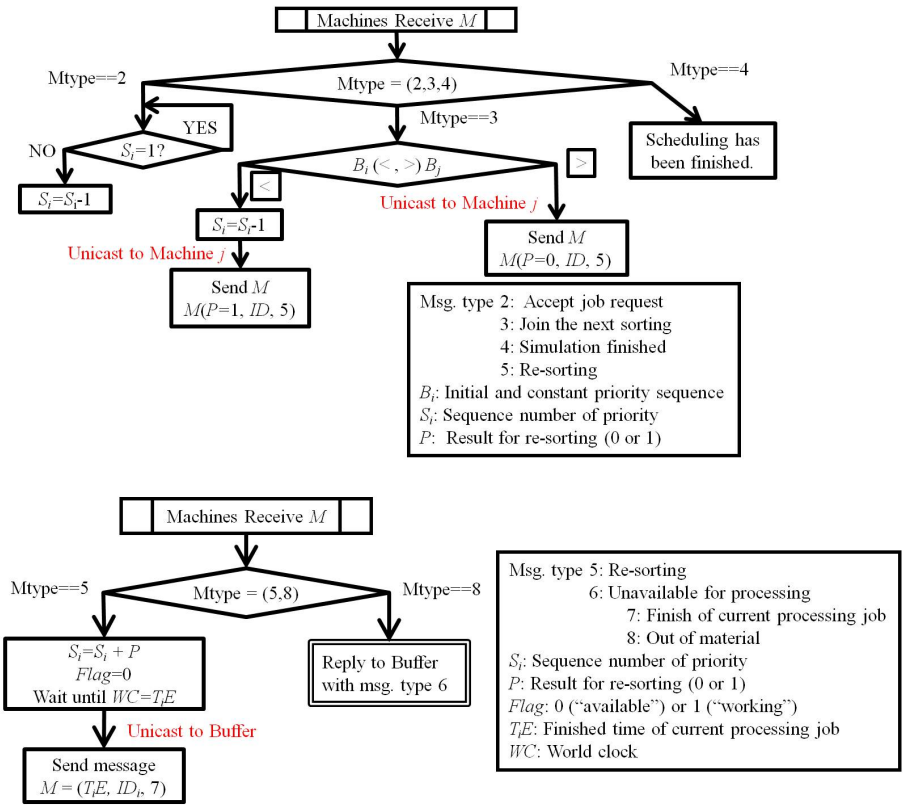


Fig. 6. Machines' operations according to received message type

mediately when the situation has changed. The HD simulation/scheduling, where each facility is equipped with a simulation/scheduling model and can communicate with other facilities via a network, is expected to be a key technique for realization of such flexible manufacturing systems.

In this paper, a method for HD scheduling has been proposed. The simple scenario was considered where materials are transported to an area one-by-one at a constant time period and then processed by one of the facilities having different performance. In this scenario, a dispatching problem based on the SPT rule has been discussed. A framework and algorithms to achieve the dispatching by a set of elementwise models and message exchanges among them were proposed, and their feasibility was shown by a simple example.

Future works include the following issues: one is quantitive evaluation of the proposed method compared to conventional distributed and centralized methods. Another is enhancement of the proposed method to the more complex scenarios where multiple kinds of materials are transported to the area randomly, and

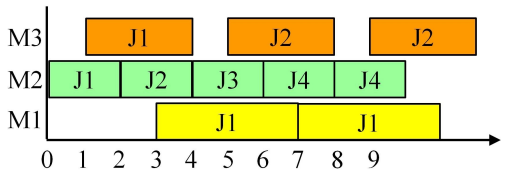


Fig. 7. The schedule obtained by the SPT rule

the materials require multiple kinds of processes with transportation among the facilities. The other is development of algorithms for other dispatching rules such as the earliest-due-date rule, the minimum slack time rule, and so on.

Acknowledgement

We would like to thank Ms. Hejun Pan, who is currently with Panasonic Corporation, for her tremendous contribution, and also the members of Technical Committee on Manufacturing and Management Knowledge, The Japan Society for Precision Engineering, for their invaluable advice.

References

1. Graves, S.C.: A Review of Production Scheduling, Operations Research, 29, 646-675 (1981)
2. Mendez, C.A., Cerda, J. and Grossmann, I.E.: State-of-the-art review of optimization methods for short-term scheduling of batch processes, Computers and Chemical Engineering, 30, 913-946 (2006)
3. Noronha, S.J. and Sarma, V.V.S.: Knowledge-Based Approaches for Scheduling Problems: A Survey, IEEE Transactions on Knowledge and Data Engineerng, 3, 160-171 (1991)
4. Akyol, D.E. and Bayhan, G.M.: A Review on Evolution of Production Scheduling with Neural Networks, Computers & Industrial Engineering, 53, 95-122 (2007)
5. Tuncel, G. and Bayhan, G.M.: Applications of Petri Nets in Production Scheduling: A Review, International Journal of Advanced Manufacturing Technology, 34, 762-773 (2007)
6. Kaihara, T., Fujii, N., Toide, S., et al.: Optimization Method using Combinatorial Auction for Production Scheduling with Batch Processing, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 4, 588-596 (2010)
7. Fujii, S., Fujii, N., Iwamura, K., et al.: A Basic Study on a Highly Distributed Simulation of Manufacturing Systems under the Ubiquitous Environment, Proceedings of ASME/ISCIE 2012 International Symposium on Flexible Automation, ISFA2012-7208 (2012)
8. Pan, H., Morinaga, E., Wakamatsu, H. and Arai, E.: A Scheduling Method for Highly Distributed Manufacturing System, Proceedings of JSME Manufacturing Systems Division Conference, 21-22 (2012)