A Method of Distributed Production Management for Highly-Distributed Flexible Job Shops

Daiki Yasuda, Eiji Morinaga, Hidefumi Wakamatsu

▶ **To cite this version:**

Daiki Yasuda, Eiji Morinaga, Hidefumi Wakamatsu. A Method of Distributed Production Management for Highly-Distributed Flexible Job Shops. IFIP International Conference on Advances in Production Management Systems (APMS), Aug 2020, Novi Sad, Serbia. pp.485-492, 10.1007/978-3-030-57997-5_56 . hal-03635610

HAL Id: hal-03635610
https://inria.hal.science/hal-03635610

Submitted on 8 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A Method of Distributed Production Management for Highly-Distributed Flexible Job Shops

Daiki Yasuda[1], Eiji Morinaga[2], Hidefumi Wakamatsu[1]

[1] Graduate School of Engineering, Osaka University, Japan
{daiki.yasuda, wakamatu}@mapse.eng.osaka-u.ac.jp
[2] Graduate School of Humanities and Sustainable System Sciences, Osaka Prefecture University, Japan
morinaga@kis.osakafu-u.ac.jp

Abstract. Recent developments of computer technology and information and communication technology are realizing highly-distributed manufacturing systems (HDMSs) in which each machine is computerized and can communicate with other machines. For this type of manufacturing system, a distributed method of discrete event simulation and a distributed method of job shop production scheduling were proposed, respectively. Because these methods use a common distributed sequencing algorithm for HDMSs, they can be integrated and an integrated method of distributed simulation and scheduling was also proposed for job shops. This paper describes an extension of the method to flexible job shops, in which an operation is processed by one of multiple machines. A distributed algorithm for selecting the machine which processes the next operation of an intermediate job based on a given dispatching rule was proposed. By incorporating this algorithm, the conventional method can be applied to flexible job shops. In addition, a method of optimizing the machine selection by adjusting the ratio of each of multiple dispatching rules was proposed. The feasibility of the proposed method was shown by computer experiments.

Keywords: Distributed Production Management, Highly-Distributed Manufacturing Systems, Flexible Job Shop, Dispatching Rules, Simulated Annealing, Internet of Things

1 Introduction

Diversified customers' needs have caused the transition from low-mix high-volume production to high-mix low-volume production and agile manufacturing in this half century. In these styles of production, it is important to flexibly respond to fluctuations in production conditions, and it is required to develop a method for real-time production management. To achieve this flexible manufacturing, it is necessary to immediately generate, evaluate, modify and determine an operation plan which is suitable for the

new condition. As for the immediate generation of an operation plan, there have been proposed many methods for reactive scheduling that utilize meta-heuristics [1], knowledge-based systems [2], neural networks [3], multi-agent systems [4-6] and so on. As for the evaluation of an operation plan, manufacturing simulation has been discussed and many commercial simulators have been developed. However, those simulators are not used for the purpose of immediate evaluation because of the heavy load of model construction [7].

Recent developments of computer technology and information and communication technology are realizing highly-distributed manufacturing systems (HDMSs) in which each machine is computerized and can communicate with other machines. With this type of manufacturing systems, distributed discrete event simulation can be performed by providing each machine with a function to simulate its behavior by constructing a simulation sub-model on its own computational resource and with a function to exchange information about itself for executing events in chronological order in the entire system. Based on this concept, highly-distributed manufacturing system simulation has been discussed [8,9].

Along with such a distributed method for evaluating an operation plan, it is desirable to also develop a distributed method for generating an operation plan. As for distributed production scheduling, methods using combinatorial auction [10], mediator architecture [11] and active database [12] have been proposed. However, from the point of view of performing production scheduling together with the immediate evaluation of the generated schedule by the highly-distributed manufacturing system simulation, it is desirable to develop another distributed scheduling method which is based on the same architecture that the simulation method has. For this reason, highly distributed scheduling methods utilizing the time management mechanism of the highly-distributed manufacturing system simulation have been proposed [13-16].

These distributed methods of scheduling and simulation for HDMSs can be integrated because they are based on the same time management mechanism. The integration makes it possible to generate better operational plans efficiently. It is also possible to obtain good plans by incorporating an optimization method in the integrated method. For this aim, a method to integrate them and perform optimization was given for job shops [17]. In actual production, it is common that there are multiple machines which can process a certain operation [18,19]. This paper describes an extension of the integrated method to flexible job shops.

2 Highly-Distributed Manufacturing Management System for Job Shops [17]

This paper considers a manufacturing system composed of an automated storage/retrieval system (AS/RS) and machining centers (MCs). Each job (product) is initially stored in the AS/RS, transported to MCs which process its required operations, and finally returned to the AS/RS. Each MC has a buffer of finite capacity for storing pending jobs and processed jobs, and processes the pending jobs stored in it one by one. The

processed job is passed to the MC which is responsible for the next operation by automated guided vehicles (AGVs). Each job is temporarily returned to the AS/RS when the buffer of the recipient MC is full. Production ends when the final operation of all jobs is completed and the AS/RS receives all finished products.

Each of the AS/RS and MCs, which is a component of this production system, has a calculation function and a communication function, and can perform autonomous decision-making about itself and exchange information by communication with other components. In transport using AGVs, it is important to avoid collision between AGVs. In production sites, a practical method for avoiding the collision is adopted. In the method, the transport path is divided into plural sections (called *zones* in this paper) with a switch for controlling an incoming AGV based on the presence or absence of an AGV in the section. In this paper, the zones are dealt with components of the production system. Each zone is assumed to have a calculation function and a communication function also as well as the AS/RS and MCs. The role of a zone is to simulate transport of intermediate products and finished products between machines by passing information about an AGV between zones by communication. There is a constraint that an AGV cannot enter a zone with another AGV until the preceding AGV goes out from the zone. To deal with uncertainty caused by this constraint, it is necessary to perform simulation.

The highly-distributed manufacturing system simulation [8,9] executes a discrete event simulation in the highly distributed environment described above. Events that cause a change in the system status occur at discrete points in time. The execution of the simulation is realized by repeating the following steps. Step 1: Advance the simulation time (simulation clock) to the next discrete time point; Step 2: Extract all events that can occur at the new time; Step 3: The process associated with the occurrence of each event is executed to change the state of the system. In order to execute this in a distributed environment, it is necessary to generate events that occur one after another in various places in the system, while maintaining temporal consistency. For this purpose, the machine is ordered based on the earliest event occurrence time of each machine, and the right to execute simulation is given to the machine with the earliest time.

There are four types of events to be executed on the machine which has acquired the right for simulation: job reception, starting processing, finishing processing, and a transport. Each process is as follows: In the case of an event of job reception, the events of starting processing and finishing processing for the next process of the received job are generated. If the machine is idle, the occurrence time of the generated event of starting processing is determined to the time when the job is received. If the machine is processing another job, the occurrence time of the generated event is determined to the time at which the current process is finished.

In the case of an event of starting processing, first, the event list is checked whether there are plural earliest events of starting processing of stored jobs having the same occurrence time. If there are such events, one of the jobs is selected using pregiven dispatching rules (SPT, LWKR, etc.), the weights of which are optimized using similar technique as that for flexible job shops described in Section 3.2. The event of starting processing for the selected job is executed, and then the occurrence time of the event of finishing processing of the job is determined. The occurrence time of the events of

starting processing of the unselected jobs are changed to the time at which the processing of the selected job finishes. If there are not plural earliest events of starting processing with the same occurrence time, the earliest event of starting processing is executed, and the occurrence time of the event of finishing processing is determined.

In the case of an event of finishing processing, a request to transport the job to the machine that processes the next operation of the job is assigned to an AGV.

In the case of an event of transport, information of the zone having the right for simulation is passed to the zone in the direction of travel of the AGV and vice versa, and the information about the AGV and the object being transported is transmitted to the recipient zone. The zone receiving the AGV generates an event of transport within itself.

In this system, sequencing is performed using two types of algorithms. One is that called *initial sorting algorithm*, which is mainly used in the initial stage of simulation. This sorting algorithm orders the priorities of all machines by having each machine inform all other machines of its time and adjust its priority based on received information. The other is that called *re-sorting algorithm*, which is mainly used for performing sequencing in intermediate stages of simulation. This algorithm reorders the priority of all machines by having the machine the time of which has changed inform all other machines of the change and the related machines adjust their priorities based on received information.

3 Highly-Distributed Manufacturing Management System for Flexible Job Shops

3.1 Processing Machine Selection and Prioritization

In job shops, an operation of a job is processed by a pre-specified machine. But, in flexible job shops, there are a plurality of machines that can process the operation. In job shop scheduling, the MC responsible for each operation of each job is determined in advance, so the transfer of jobs between operations can be simple processing. However, in flexible job shop scheduling, before processing a job for delivery, it is necessary to select an MC to be in charge of the next operation of the job. A common dispatching rule for performing this selection is PT rule that gives priority to a high-performance MC, and a highly distributed scheduling method using this rule has been proposed [16]. In this approach, before assigning a job, each machine autonomously determines the initial value of the priority of the machine in accordance with the PT rule that prioritizes high-performance MC. This process is called *initial sorting*. Each machine broadcasts the required processing time PT_i and the ID number ID_i to other machines. Thereafter, each machine receives a message from another machine (ID_j) and analyzes the magnitude relationship with the required time PT_j of the sender machine. If the required time PT_i is larger, the value of S_i (initial value is 1), which represents the own priority of the machine for each product, is incremented by one to lower the priority. If they are equal, the one with the smaller ID number has priority. By performing this set of processes, the correct priority sequence of all machine is obtained.

An example of processing is shown below. First, AS/RS assigns each job to one of the MCs that can take charge of processing its first operation. At this time, the priorities of the MCs based on the PT rule is calculated and the job is assigned to the MC with the highest priority. A transportation event and a job reception event are then generated by job allocation. Next, by the initial sorting algorithm, a machine (AS/RS, MC or zone) with the earliest event is determined, and the machine executes its earliest event. After that, the priorities of the generated events are ordered by the re-sorting algorithm, so that a machine with the earliest event is determined, and the machine executes its earliest event. When the job arrives at the MC, the MC executes the event of starting processing for the job. After that, when the event of finishing processing for the job is executed, the MC responsible for the next operation of the job is determined by performing the machine selection based on the PT rule. If the selected MC has multiple jobs in its buffer when it reaches the idle time, the priorities of those jobs is calculated and the event of starting processing for the job with the highest priority is executed.

3.2 Machine Selection Optimization

If only the PT rules is used, jobs may be excessively concentrated on high-performance machine, resulting in a poor schedule. Thus, if only one dispatching rule is adopted, the shortcomings of the rule will have a significant effect on the schedule. Therefore, it can be expected that the shortcomings of each rule can be compensated by using some dispatching rules together. There are various rules for machine selection other than the PT rule. In this study, the excessive concentration of jobs to a machine is suppressed by using the NINQ rule, which can be expected to reduce the machines that become idle, together with the PT rule. The NINQ rule is a rule that, when assigning jobs, assigns to the MC that has the least number of jobs in the buffer among the assignable MCs. The evaluation value of the PT rule is time, and the evaluation value of the NINQ rule is the number of jobs. Therefore, when these are used together, it is necessary to perform normalization to make the value range uniform. The following shows how to normalize the PT rule and the NINQ rule respectively.

$$V_{PT} = \frac{PT - PT_m}{PT_M - PT_m} \tag{1}$$

PT: Processing time (If the MC is unavailable, a large value is set to this variable.)
PT_m: Minimum processing time for one operation in all machine
PT_M: Maximum processing time for one operation in all machine

$$V_{NINQ} = \frac{NumberofJobs}{BufferSize} \tag{2}$$

NumberofJobs：Number of jobs in the buffer
BufferSize：Buffer capacity of each MC

The evaluation value V_i is calculated for each rule and multiplied by the rule weight W_i. The sum of them is regarded as the priority P_{MC} of each job as follows.

$$P_{MC} = W_{PT} * V_{PT} + W_{NINQ} * V_{NINQ} \tag{3}$$

Since each V_i is defined by the rule that a small value is good, the MC with the smallest P_{MC} value is preferentially selected. Using the weight W_i of each rule in this equation as a solution, the weight is optimized by simulated annealing (SA). Since W_i is a continuous variable, it generates a neighborhood solution using normal random numbers. Normal random numbers are generated by the Box-Muller method. If the random variables X and Y are independent of each other and both follow a uniform distribution on [0,1], then, z_{i1} defined by the following formula becomes independent random variables according to the standard normal distribution $N(0,1)$ with mean 0 and variance 1.

$$z_{i1} = \sqrt{-2\log X} * \cos 2\pi Y \tag{4}$$

The random variable x following the normal distribution $N(\mu,\sigma)$ is expressed as follows.

$$x = \mu + \sigma z_{i1} \tag{5}$$

Since the neighborhood solution $\widetilde{W}_i$ of the solution W_i is a normal random number according to $N(W_i,\sigma)$, the neighborhood solution is obtained by the following equation.

$$\widetilde{W}_i = W_i + \sigma z_{i1} \tag{6}$$

Weight adjustment is performed by the above method.

4 Numerical Example

Processing flows based on the proposed algorithms and concept were coded in the C language and the proposed method was implemented on a generic workstation in semblance by interprocess communication with sockets due to machine constraints. The method was applied to an example of flexible job shops of 10 jobs which require 10 operations using 3 machining centers (MCs), 1 automated storage/retrieval system (AS/RS) and 2 AGVs shown in Fig. 1.

As a result of optimizing the makespan with the number of iterations in SA set to 200, a schedule with a makespan of 281 was obtained. Table 1 shows the numbers of message exchange which occurred in the simulation. Assuming that the number of machines is n, $n(n\text{-}1)$ times of communication are required for the initial sorting algorithm, and $2(n\text{-}1)$ times of communication are required for the re-sorting algorithm. When the number of jobs is o, the number of operations is p and the maximum transport distance is q zone, the maximum number of events from the start to the end of production is $op(q+3)$. By these characteristics of the communication algorithms, the number of message exchange tends to be large. Assuming that a message is sent based on TCP and its size is 80byte, the total size is about 792MB. If the communication is performed using a standard of wireless LAN (IEEE 802.11n), it can be completed in about 9.9sec. This result shows a potential of the proposed method for practical use.

5 Conclusions

This paper described an extension of a highly-distributed manufacturing management system to flexible job shops. A method for handling multiple dispatching rules and performing optimization has been applied to a set of processing flows of the management system for flexible job shops. Numerical experiments implied that the proposed

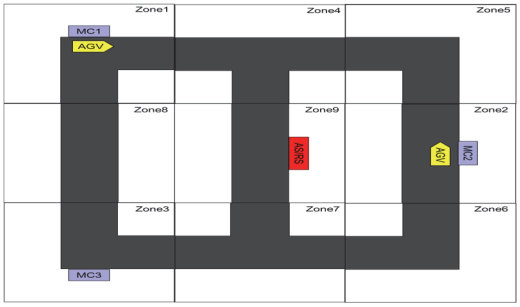

Fig. 1. Example used for case study.

Table 1. Number of message exchange which occurred in the simulation.

Machine type	Number of message exchange
AS/RS	547837
MC1	1086559
MC2	1071049
MC3	865399
Zone1	639457
Zone2	589978
Zone3	944737
Zone4	829597
Zone5	592567

method can generate an optimal operation plan with a practical number of message exchanges.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP18K03872.

References

1. Zheng, L., Gao, L.: A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem. International Journal of Production Research, 51(12), 3516-3531 (2013)
2. Szelke, E., Kerr, R.M.: Knowledge-based reactive scheduling. Production Planning & Control, 5(2), 124-145 (1994)
3. Garetti, M., Taisch, M.: Using neural networks for reactive scheduling. Artificial Intelligence in Reactive Scheduling, 146-155 (1995)
4. Archimede, B., Coudert, T.: Reactive scheduling using a multi-agent model: The SCEP framework. Engineering Applications of Artificial Intelligence, 14(5), 667-683 (2001)
5. Lou, P., Liu, Q., Zhou, Z., Wang, H.: Multi-agent-based proactive–reactive scheduling for a job shop. International Journal of Advanced Manufacturing Technology, 59(1-4), 311-324 (2012)

6. Sabuncuoglu, I., Kizilisik, O.B.: Reactive scheduling in a dynamic and stochastic FMS environment. International Journal of Production Research, 41(17), 4211-4231 (2003)
7. Fujii, S., Hibino, H., Iwamura, K., Tsumaya, A., Sashio, K.: Simulation of Manufacturing System under Ubiquitous Environment. Journal of the Japan Society for Precision Engineering, 74, 1016-1019 (2008) (In Japanese)
8. Fujii, S., Fujii, N., Iwamura, K., Morinaga, E., Tsumaya, A., Inoue, T., Mariyama, T.: A Basic Study on A Highly Distributed Simulation of Manufacturing Systems under The Ubiquitous Environment. In Proceedings of the ASME/ISCIE 2012 International Symposium on Flexible Automation, ISFA2012-7208, 321-324 (2012)
9. Morinaga, E., Yasuda, D., Imagawa, Y., Wakamatsu, H., Tsumaya, A., Inoue, T., Iwamura, K., Ishibashi, M., Fujii, N., Arai, E., Fujii, S.: A Study on Highly-Distributed Manufacturing System Simulation. Procedia Manufacturing, 39, 50-57 (2019)
10. Kaihara, T., Fujii, N., Toide, S., Ishibashi, H., Nakano, T.: Optimization Method using Combinatorial Auction for Production Scheduling with Batch Processing. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 4(3), 588-596 (2010)
11. Matsumoto, T., Kato, Y., Nagafune, S., Wakamatsu, H., Shirase, K., Arai, E.: Advanced Autonomous Distributed Manufacturing System Using Active Database. Transactions of the Japan Society of Mechanical Engineers, 65(630), 837-843 (1999) (In Japanese)
12. Tönshoff, H-K., Seilonen, I., Teunis, G., Leitão, P.: A mediator-based approach for decentralized production planning, scheduling, and monitoring. In Proceedings of the CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, 113-118 (2000)
13. Morinaga, E., Takagi, A., Sakaguchi, Y., Wakamatsu, H., Arai, E.: Basic study on production scheduling method for highly-distributed manufacturing systems. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 8(5), JAMDSM0072 (2014)
14. Morinaga, E., Sakaguchi, Y., Wakamatsu, H., Arai, E.: A Distributed Production Scheduling Method for Highly-Distributed Manufacturing Systems. Advances in Production Management Systems; Innovative and Knowledge-Based Production Management in a Global-Local World, Part I, Springer, 531-538 (2014)
15. Morinaga, E., Nakamura, T., Wakamatsu, H., Arai, E.: A method for highly-distributed manufacturing systems. In Proceedings of the 23rd International Conference on Production Research, ICPR23, Manila; Philippines; (31 July 2015-6 August 2015) 1100
16. Morinaga, E., Nakamura, T., Wakamatsu, H., Arai, E.: Flexible Job-Shop Scheduling Method for Highly-Distributed Manufacturing Systems. Journal of Smart Processing, 6(5) 181-187 (2017)
17. Yasuda, D., Morinaga, E., Wakamatsu, H.: Integrated production scheduling and simulation for highly-distributed manufacturing systems. In Proceedings of 2020 JSPE Spring Conference, 345-346 (2020) (In Japanese)
18. Kopp, S., Dauzère-Pérès, S., Yugma, C.: Flexible job-shop scheduling with extended route flexibility for semiconductor manufacturing. In Proceedings of the 2014 Winter Simulation Conference, 2478-2489 (2014)
19. Alvarez-Valdes, R., Fuertes, A., Tamarit, J.M., Giménez, G., Ramos, R.: A heuristic to schedule flexible job-shop in a glass factory. European Journal of Operational Research, 165, 525-534 (2005)