A Fuzzy Criteria Weighting for Adaptive FMS Scheduling

Kikwang Lee, Wan Chul Yoon and Dong Hyun Baek

Department of Industrial Engineering,
Korea Advanced Institute of Science and Technology

Abstract

Application of machine learning to scheduling problems has focused on improving system performance based on opportunistic selection among multitudes of simple rules. This study proposes a new method of learning scheduling rules, which first establishes qualitatively meaningful criteria and quantitatively optimizes the use of them, a similar way as human scheduler accumulate their expertise. The weighting of these criteria is trained in response to the system states through simulation. To mimic human quantitative feelings, distributed fuzzy sets are used for assessing the system state. The proposed method was applied to job dispatching in a simulated FMS environment. The job-dispatching criteria used were the length of the processing time of a job and the situation of the next workstation. The results show that the proposed method can develop efficient and robust scheduling strategies, which can also provide understandable and usable know-hows to the human scheduler.

1. Introduction

Optimizing production scheduling is one of the most important means to enhance productivity of modern manufacturing industries. Heuristic rules have often been adopted for their simplicity and understandability.

Blackstone et al. [2] showed that, although no rule can always be the best in all the states of floor, some rules tend to perform consistently better than others in certain situations. Based on this idea, opportunistic strategies which dynamically select the most suitable rule considering the current state of the system were devised [3], [5]-[7]. To use this approach, one should wisely select or devise several attributes which properly represent the system states. This approach also requires some evaluation methods to determine the best rule at a given state. Simulation has most often been adopted for learning the situational behavior for its ability to use past field data in that a great number of trials and observations are possible in a short time.

Another important source of knowledge about the effectiveness of rules at different system states is the human expertise. Extracting such human expertise is not easy since much of its quantitative part is only implicit or 'subconscious'. Also, in very complex modern manufacturing environments, the human expertise may be hampered by low repetition rate of same states and noisy feedback of decisions previously made. For this, some researchers doubt the quality of human expertise in scheduling problems [1]. Thus the quantitative knowledge may better be collected and analyzed by a computer provided that a sound simulation model is available.

This study proposes a new method of learning that is similar to the way in which human schedulers presumably accumulate their expertise. Decision criteria that are qualitatively meaningful are first devised and the use of the criteria is then quantitatively optimized through learning with simulation. The quantitative use of the criteria can be implemented by adjusting the weights of those criteria according to the assessed situation. Also the applicability of a weight set can be quantitatively assessed by comparing the current value set of the state variables with that of the learned states.

This strategy was evaluated by applying to the optimization of job dispatching in a simulated FMS. Four situational variables and two decision criteria were used in this example. We used fuzzy set theory in integrating the state variables to assess the situations. The results showed that the proposed approach is consistently more effective than commonly used heuristics.

In chapter 2 the proposed learning scheme is described. Chapter 3 presents the application of this method to an FMS and discusses the results.

2. The learning and decision-making scheme

Our learning scheme mimics the way in which human schedulers accumulate their expertise, i.e., establishing some semantically meaningful criteria and quantitatively optimizing the use of them. In dealing with the criteria we use fuzzy set theory aiming at the potential benefit of robustness and smoothness. Using fuzzy number instead of crisp number used in most production rules is a widely accepted alternative to cope with the vulnerability to changes in system situation. To assure both the smoothness and sensitivity, we assign multiple fuzzy numbers to instances of system state variables [4].

The learning is first prepared by an initialization step in which the basic characteristics of the manufacturing system is obtained through simulation. This information is later used to scale and fuzzify the numerical values of the system state variables for evaluating the current state of system. Fig. 1 shows the overall architecture of our learning and decision-making scheme.

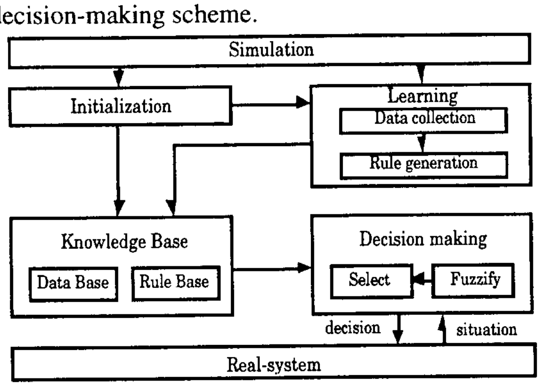

Fig. 1 The intelligent scheduling system

2.1. Learning by simulation

2.1.1. Distributed fuzzy representation of state variables

To enhance classification power in fuzzy subspaces for effective learning, using multiple fuzzy numbers simultaneously from coarse numbers to fine ones is proposed by Ishibuchi et al. [4].

Suppose that a set of fuzzy numbers { A_1^K, A_2^K, $\cdots$, A_K^K } is used for both of the two axes X_1 and X_2 in the pattern space where K is the largest fuzzy number in the set.

In this case, an ordinary fuzzy rule can be represented as follows:

If x_1 is A_i^K and x_2 is A_j^K then $\cdots$, $i, j = 1, 2, \cdots, K$. (1)

A distributed fuzzy rule is defined with regard to different values of K simultaneously. If it is 3 and 5, a distributed fuzzy rule is presented as follows:

If x_1 is A_i^K and x_2 is A_j^K then $\cdots$, $i, j = 1, 2, \cdots, K$; $K = 3,5$. (2)

2.1.2 The Construction of rule base

Suppose that the training data consist of the state vectors and the performance of all possible decisions. If the state vector is characterized by m attributes, it is denoted by $\mathbf{x}_p=(x_{1p}, x_{2p}, \ldots, x_{mp})$, $p=1,2, \ldots, n$, where n is the data size. When there exist possible decisions, a performance value of decision D_t , $t=1,2,\ldots,T$ at the state $\mathbf{x}_p$ can be denoted by $PV^{\mathbf{x}_p}_{D_t}$. If x_{1p} is A_i^K, x_{2p} is A_j^K, ..., x_{mp} is A_v^K then $\mathbf{x}_p$ is said to have a value of $\mathbf{A}^K_{ij\cdots v}$.

We construct rule base which contains the following fuzzy if-then rules with grade of certainty (GC) : If $\mathbf{x}_p$ is $\mathbf{A}^K_{ij\cdots v}$ then use D_t with $GC = GC(D_t \mid \mathbf{A}^K_{ij\cdots v})$.

If a given state $\mathbf{x}_p$ has a value of $\mathbf{A}^K_{ij\cdots v}$, the grade of certainty (GC) of each decision D_t is calculated by the following procedure.

<u>Notations</u>

$PV^{\mathbf{x}_p}_{\max}$: maximum performance value by the best decision at state $\mathbf{x}_p$

$PV^{\mathbf{x}_p}_{\min}$: minimum performance value by the worst decision at state $\mathbf{x}_p$

$\mu_i^K(x)$: membership function which defines fuzzy set A_i^K assuming the state space is indexed by K fuzzy numbers { A_1^K, A_2^K, $\cdots$, A_K^K }

(i) Let us assume that a smaller performance value represents a better performance. The degree of effectiveness($DE^{\mathbf{x}_p}_{D_t}$) of each decision D_t at state $\mathbf{x}_p$ is calculated as follows :

$$DE^{\mathbf{x}_p}_{D_t} = \frac{PV^{\mathbf{x}_p}_{\max} - PV^{\mathbf{x}_p}_{D_t}}{PV^{\mathbf{x}_p}_{\max} - PV^{\mathbf{x}_p}_{\min}}, \text{ where } t=1, 2, \cdots, T \quad (3)$$

(ii) The cumulative score of decision D_t in each state $\mathbf{A}^K_{ij\cdots v}$ is calculated as follows :

$$\beta(D_t \mid \mathbf{A}^K_{ij\cdots v}) = \sum_{p=1}^{n} \mu_i^K(x_{1p}) \mu_j^K(x_{2p}) \cdots \mu_v^K(x_{mp}) DE^{\mathbf{x}_p}_{D_t},$$
$$t = 1,2,\cdots,T;\ K = 2,3,\cdots,L. \quad (4)$$

(iii) $GC(D_t \mid \mathbf{A}^K_{ij\cdots v})$ can now be calculated as follows :

$$GC(D_t \mid \mathbf{A}^K_{ij\cdots v}) = \frac{\beta(D_t \mid \mathbf{A}^K_{ij\cdots v}) - \bar{\beta}}{\max_h\{\beta(D_h \mid \mathbf{A}^K_{ij\cdots v})\}} \quad (5)$$

where $\bar{\beta} = \sum_{t=1}^{T} \beta(D_t \mid \mathbf{A}^K_{ij\cdots v}) / T$, $1 < GC(D_t \mid \mathbf{A}^K_{ij\cdots v}) < 1$.

2.2. Making decisions based on the rule base

The procedure for finding the most favorable decision D_t for a pattern $\mathbf{x}_p$ is as follows :

(i) Calculate for $t=1,2, \ldots , T$

$$Support^K_{D_t} = \max_t\{\sum_i \sum_j \cdots \sum_z \mu_i^K(x_{1p}) \mu_j^K(x_{2p}) \cdots \mu_v^K(x_{mp}) GC(D_t \mid \mathbf{A}^K_{ij\cdots v})\}$$
$$i, j, \cdots, z = 1,2,\cdots,K;\ K = 2,3,\cdots,L. \quad (6)$$

(ii) Find the decision D^* which satisfies

$$Support_{D^*} = \max\{Support^1_{D^*_l}, Support^2_{D^*_l}, \cdots, Support^L_{D^*_l}\}. \quad (7)$$

(iii) If the support of selected decision D^* is less than a user-defined threshold, the control vector which showed the best performance in initialization phase is used.

3. Application of the method to FMS

3.1. System description

The proposed learning and decision making was implemented in a hypothetical FMS as shown in Fig. 2. We construct a rule base by learning only with job mix A, and then apply the rule base to job mix B and C to verify the robustness of the acquired rule base. Job mix A, B and C differ from each others in the proportion of job types.

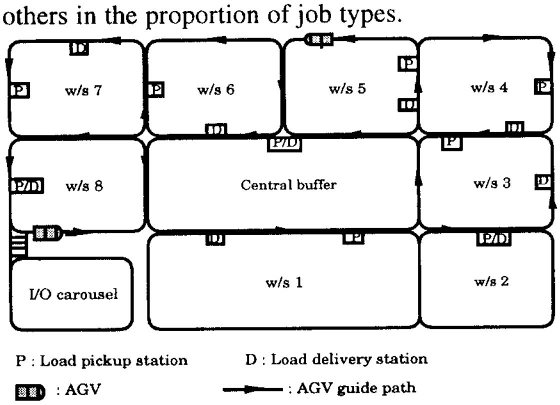

Fig. 2 A hypothetical FMS

3.2. Initializing by simulation

The following data are obtained in the initialization phase used in later learning and decision making.

(i) maximum/minimum/average number of jobs in the incoming buffer queue of each workstation

(ii) maximum/average sum of operation processing times of all jobs in the incoming buffer queue of each workstation

(iii) maximum/average idle time of each workstation

3.3. Learning by simulation

3.3.1. Distributed fuzzy set

Two sets of fuzzy numbers as shown in Fig. 3 are used to adopt the concept of distributed fuzzy set mentioned in 3.1.1. Accordingly, the value of state variables are converted into two sets of fuzzy numbers.

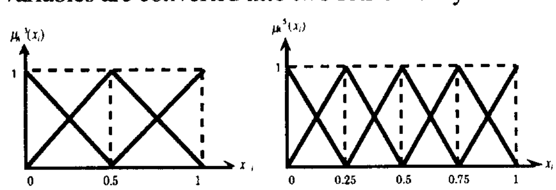

Fig. 3 Fuzzy set A^3 and A^5

3.3.2. The structure of training data

(1) Control vector and criteria

The system calculates the priority of job k in the incoming buffer as follows :

$$\text{Priority}(k) = \text{Weight}_1 \times \text{Criterion}_1(k) + \text{Weight}_2 \times \text{Criterion}_2(k) \quad (8)$$

The decision here is to determine the pair of values of the weights, which will be referred to as *control vector.* A larger value means higher priority.

The degree of urgency of subsequent workstation (DUNQ) and the degree of length of processing time (DLPT) are selected for the criteria. DUNQ reflects the potential starvation (the idle time of workstation) or excessive incoming buffer queue length in the subsequent workstation. A large value implies excessive queue and a small one starvation. DLPT is calculated based on the processing time of each job in the incoming buffer and a large value indicates the shorter processing time. The values of DUNQ and DLPT are normalized to [0, 1]. More detail calculations are omitted for space.

The elements of a control vector are the two weights corresponding to DUNQ and DLPT and are shown in Table 1. A negative value of w_2 for DLPT implies that a job with the largest processing time gets a highest priority. A positive value of w_2 implies the opposite.

Table 1. The weight combination

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
w_1	1	1	1	1	1	1	1	0	0.25	0.5	0.75	0	0.25	0.5	0.75
w_2	-0.75	-0.5	-0.25	0	0.25	0.5	0.75	1	1	1	1	-1	-1	-1	-1

(2) State vector $\mathbf{x}=(x_1,x_2,\ldots,x_m)$

x_1 = the maximum degree of LPT (Longest Processing Time) among all jobs in the input buffer

x_2 = the maximum degree of urgency of subsequent w/s calculated for jobs in the incoming buffer

x_3 = the maximum degree of SPT (Shortest Processing Time) among all jobs in the incoming buffer

x_4 = the degree of incoming buffer queue length

To represent above state by multiple fuzzy numbers, a distributed fuzzy set A_i^K for x_1, x_2, x_3 and x_4 are defined as shown in Fig. 2.

(3) Resulting performance $PV_{D_l}^{x_p}$

The $PV_{D_l}^{x_p}$ is the resulting mean flow time when D_t is used as a control vector in state x_p.

3.3.3. Data collection

In order to get the data for learning during the simulation in the phase of data collection, whenever a dispatching decision is required, the sub-simulation function evaluates the performance of all the control vectors. During the sub-simulation, if the same state as the starting one occurs the control vector which is under consideration is used, otherwise the one which showed the best performance in initialization phase is used as a default control vector.

If there already exists one or more rules learned for the current system state, the *Support* of the most favorable control vector is examined. If it is greater than a user-defined threshold then the best control vector is used for the current decision. Otherwise, data collection procedure is executed for additional learning.

3.4. Simulation result

The mean flow time till 400 throughput was used as the evaluation measure. When the simulation starts, all the machines were idle and their buffers were all empty. In order to remove the initial bias and to collect data in the steady state, we discarded first 1000 observations.

The performance of our learning method was compared with those of 12 widely used dispatching rules [8] and 15 fixed control vectors shown in Table 1.

Table 2 shows the results of the simulation experiments with the rules which have consistently better performance than others in [8]. The experiment was conducted with eight replications per every rule.

Learn1 used a rule base which was learned from the scratch. Learn2 cumulated more learning starting with the rule base built in Learn1. Learn2 improved performance over Learn1.

Note that even though the rule bases used in Learn1 and Learn2 are acquired only with job mix A, they showed good performances in the case of job mix B and C, too. This implies that our learning method resulted in robust rules that are effective in different system states.

Table 2. Simulation result

Rule	Jobmix A			JobmixB			JobmixC		
	MFT	s.d	Imp(%)	MFT	s.d	Imp(%)	MFT	s.d	Imp(%)
FCFS	280.7	4.6	0.0	249.0	23.1	0.0	265.2	33.5	0.0
SPT	276.5	9.0	1.5	247.7	24.7	0.5	260.8	31.8	1.6
LPT	293.1	12.5	-4.4	260.6	23.0	-4.7	281.5	24.8	-6.2
MWKR	280.8	5.4	0.0	250.9	23.2	-0.7	269.1	36.2	-1.5
LWNQ	276.0	5.9	1.7	248.5	21.9	0.2	265.9	34.9	-0.3
Learn1	274.7	6.5	2.2	244.4	24.7	1.8	259.5	31.6	2.2
Learn2	267.9	6.1	4.6	244.2	23.8	1.9	254.8	30.0	3.9

* MFT : Mean flow time(minute)
s.d : standard deviation
Imp(%) : Percentage of improvement to FCFS

4. Conclusions

This study developed a new method of learning job-dispatching rules that is presumably similar to the way in which human schedulers accumulate their expertise. The proposed learning method selects qualitatively meaningful criteria and quantitatively optimizes the use of them. The space of system states was represented with distributed fuzzy numbers to emulate the robustness and smoothness of human expertise. The quantitative optimization of the use of the selected criteria is done by learning the appropriate weight set for the criteria through simulation at different system states.

Application results in the hypothetical FMS showed that the proposed method improved system performance compared to conventional heuristics. Furthermore, the learned rule set was found to retain its effectiveness in different environment created by different job mix patterns.

References

[1] Aytug, H., Bhattacharyya, S., Koehler, G. J., and Snowdon, J. L., "A review of machine learning in schduling", IEEE Transactions on Engineering Management, Vol.41, No.2, pp.165-171, 1994.

[2] Blackstone, J. H., Phillips, D. T., and Hogg, G. L., "A state-of-the-art survey of dispatching rules for manufacturing job shop operations", INT. J. PROD. RES., Vol.20, No.1, pp.27-45, 1982.

[3] Chiu, C., and Yih, Y., "A learning-based methodology for dynamic scheduling in distributed manufacturing systems", INT. J. PROD. RES., Vol.33, No.11, pp.3217-3232, 1995.

[4] Ishibuchi, H., Nozaki, K., and Tanaka, H., "Distributed representation of fuzzy rules and its application to pattern classification", Fuzzy Sets and Systems, 52, pp.21-32, 1992.

[5] Nakasuka, S., and Yoshida, T., "Dynamic scheduling system utilizing machine learning as a knowledge acquisition tool", INT. J. PRD. RES., Vol.32, No.2, pp.411-431, 1992.

[6] Pierreval, H., and Ralambondrainy, H., "A simulation and learning technique for generating knowledge about manufacturing systems behavior", J. Opl Res. Soc., Vol.41, No.6, pp.461-474, 1990.

[7] Piramuthu, S., Raman, N., and Shaw, M. J., "Integration of simulation modeling and inductive learning in an adaptive decision support system", Decision Support Systems, 9, pp.127-142, 1993.

[8] Sabuncuoglu, I., and Hommertzheim, D. L., "Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion", INT. J. PROD. RES, Vol.30, No.7, pp.1617-1635, 1992.