A NEW SCHEDULING METHOD BASED ON DISPATCHING RULES AND APPLICABLE BY SIMULATION MODELS

Ali Razzazi[1]

[1]Science and Research Branch of Iran Azad University,
Faculty of Industrial Management, Tehran, Iran

Razzazi@Gmail.com

Abstract

Scheduling and sequencing problems are an important part of production planning problem and usually many assumptions are considered in order to simplify these problems. Simulation models are used because simplifier assumptions can be put aside and so the models reflect reality more precisely. In this paper a method which performs scheduling by combining different dispatching rules has been presented. It means that when deciding on a specific machine some different dispatching rules are used to determine the best candidate on that machine. The model is run for various combination of dispatching rules and the best observed result is reported. Since dispatching rules are easily executed by simulation engines this method can be generally used as a simulation based scheduler module for production systems. Finally, some suggestions for improving the model are presented.

Keywords: Sequencing and Scheduling, Simulation, Dispatching Rules.

Presenting Author's biography

ALI RAZZAZI is a master simulation consultant. He received his MSc in Industrial Management from Science and Research Branch of Islamic Azad University and his BSc in Industrial Engineering from Isfahan University of Technology of Iran. He has been manager of number of simulation projects and has taught simulation software in number of companies and universities.

1 Introduction

Scheduling and sequencing is an important part of production planning process and has a wide variety of application in production and non production environments. In most cases applying appropriate sequence of tasks (in other word the way of using resources) has a great effect on the efficiency of companies. Now, there are many methods and algorithms which undertake scheduling and sequencing until optimal or near optimal solutions are gained but the point is these methods and algorithms are not much applicable in real world because:

1. Most of those methods have a static approach to the systems but in reality systems are dynamic and their parameters are rarely constant in the scheduling intervals.
2. Running times of methods increase exponentially when number of resources increase and this makes the methods practically infeasible.
3. Many simplifier assumptions are considered that distinguish the problems from those of the real world, for example in single machine scheduling, simplifier assumptions are:
    (a) All the works are available in the shop at the beginning.
    (b) Setup times are independent of task sequencing.
    (c) Processing times of every task are determined and constant.
    (d) Machines are always available and no idleness is allowed.
    (e) Preemption is not allowed [1].

These assumptions are not hold in most cases in a real world.

Today, regarding to competitive supply market and make to order production systems plus customers ever changing and unpredictable interests no one can expect that all works are available in the shop at the beginning of scheduling process and so the first assumption does not hold. In the cases where jigs and fixtures are used, setup times are strongly dependent on the sequencing of tasks. This situation also holds for processes like painting in which the sequence of paints is a great concern in setup times, so the second assumption is also questionable. Most of the processes do not have a constant processing time and their timing behaviors follow statistical distributions, using statistical mean instead of processing time simplifies the problem but also makes the solution distant from its real quantity (third assumption). Production resources are not always available due to machine breakdowns, maintenance activities and power supply problems and also up times are not usually predicable and in the best case it is just possible to model the failure pattern by statistical distributions (forth assumption). In some cases it is possible to postpone a task on a resource and process another task on that resource (fifth assumption).

Simulation, as a problem solving technique has a great ability to model details and complexity of the real world systems. Generally, we can make a suitable model with sufficient details by dedicating experts and spending money and of course enough time, so it means that orders can enter the model randomly and this makes the model independent of the first assumption, other assumptions are relaxed as follows. Setup times can be considered by static or dynamic patterns or by dependency upon previous operations (second assumption). It is possible to consider stochastic processing times for different operations (third assumption). Down times can be modeled by different patterns and functions (forth assumption). Finally it is possible to stop any processing and start processing another order (fifth assumption).

Another advantage of application of simulation in scheduling is the capability of generating charts and graphical figures.

Application of simulation models in scheduling and sequencing in the literature can be categorized in two categories:

In the first category, simulation models perform as a function which produces outputs depending on inputs. There is often another supporting module which is responsible for evaluation and distinguishing inputs in order to improve the performance of the model. For example meta heuristics are of the famous of such modules.

In the second category, simulation model can dynamically assign jobs according to the conditions of decision point and predefined rules. In this category dispatching rules are of special interest, while any resource is supposed to start a new job the one which has an extreme value among the other in the queue is selected and assigned to that resource.

R. Baker is one of the pioneers of collecting and summarizing the literature of this topic where in his book principles of application of simulation in scheduling and combination of dispatching rules is explained [1]. The idea of using multiple dispatching rules in a problem such that each of rules invoked in a constant time interval is also presented and proved to be an efficient way if the intervals are selected carefully [2]. Development of this approach by using non equal intervals led to enhancement of the basic idea. However it was concluded that if the intervals are determined inappropriately, using a single rule is preferable to multiple rules [3]. Further development of this approach was to use different rules in each interval for different resources [4]. Application of neural network and genetic algorithm for selecting dispatching rules is presented in 1993 [5], [6]. A new simulation based scheduling scheme based on dynamic use of multiple dispatching rules is also presented. The word dynamic implies that during run time executive situation determines the time to change the dispatching rule in use [7].

In another survey the effect of dynamic selection of dispatching rules and dynamic time interval changing

was studied and compared, in this paper it is suggested that some events such as a maximum for queue length can trigger the change point of dispatching rule [8], [9]. Selection of dispatching rule with regard to the best ones in literature is also considered [10]. Iterative simulation has also been considered for scheduling. For example where objective function is to minimize delay from due date, jobs priorities in each iteration are assigned based on the delays in previous iteration [11], [12].
In this paper combination of ten dispatching rules is used for scheduling.

2 Model Mechanism

The model is developed in Enterprise Dynamics which is discrete event simulation software; the software is able to model breakdowns, alternative routes for products, sequence dependent setup times and so is used for simulating productions systems. However in this paper other software could have been used because experimental models are standard.

Beside every resource a queue is located in which jobs can accumulate when waiting to be processed. Each time a resource is ready an order is selected from the queue by a mechanism which will be explained in the following section.

Information regarding product routings and processing times are stored in some tables in the model and each job follow its routing by retrieving data according to its current position

2.1 Selection mechanism

In this paper, 10 dispatching rules out of many dispatching rules available in literature are chosen which are as follows:

- SOPT: shortest operation processing time
- EDD: value-earlier due date
- FCFS: first come first served
- SJPT: shortest job processing time
- SRW: shortest remaining work
- PDJT: Processing time divided by job processing time
- PDRW: Processing time divided by remaining work
- PMJT: Processing time multiplied by job processing time
- PMRW: Processing time multiplied by remaining work
- SLACK

Each rule is given a weight which is a measure of the importance of that rule.
Each time, a machine is ready to process an order amounts of above rules are computed for each job in the queue stored in a matrix (decision matrix) such that each row corresponds to an job and each column corresponds to a dispatching rule amount, for example Fig. 1 shows 10 jobs waiting to be processed and the decision matrix (in the picture eight rules are shown).

jobs name	SOPT	EDD	FCFS	SJPT	SRW	PDJT	PDRW	PMJT
a= factor	0.6	0	0	0	0	0	0.4	0
1	38	1170	1	1170	1170	0.032478632	0.032478632	44460
2	81	1096	2	1096	1096	0.073905109	0.073905109	88776
3	70	986	3	986	986	0.070993914	0.070993914	69020
4	4	1207	4	1207	1207	0.003314001	0.003314001	4828
5	60	1047	5	1047	1047	0.057306590	0.057306590	62820
6	35	1015	6	1015	1015	0.034482758	0.034482758	35525
7	56	822	7	822	822	0.068126520	0.068126520	46032
8	82	1095	8	1095	1095	0.074885844	0.074885844	89790
9	58	1099	9	1099	1099	0.052775250	0.052775250	63742
10	74	1114	10	1114	1114	0.066427289	0.066427289	82436

Fig. 1 Sample of a decision matrix

Contrary to Many articles which calculate the amounts of dispatching rules and then sort them and select the one with an extreme value, in this paper each column in the decision matrix is normalized individually and then each cell is multiplied by its column weight, Since the values of some rules are desired to be low and some other are desired to be high, normalization should be done in a way that take this point into account. Then the sum of each row represents the priority of that order in that selection step. The following equation summarizes the process:

Priority of jobs = $\alpha1$*SOPT+ $\alpha2$*EDD+ $\alpha3$*FCFS+ $\alpha4$*SJPT+ $\alpha5$*SRW+ …+ $\alpha10$*Slack

Subject to:
$\alpha1+ \alpha2+ \ldots +\alpha10 = 1$

Finally, the calculated priority of jobs is sorted in a descending order.

2.2 Model Validation

To validate the model, some test problems are selected, in which their optimal solutions are attainable by using just one of the ten-selective dispatching rules. For each of problems the model has been run while the corresponding weight which leads to optimality is set to one, and the other rules are set to zero. Reaching optimal solutions for the problems approved validity of the model.

3 Analysis of result set

The proposed method has been evaluated in two sections. First, by running the model for 34 test problems and then comparing the results with optimal solutions (or best known solutions). Second, by comparing the result set with those of Weckman et al models (Weckman's model is selected because it is up to date and published in a credible journal) [13]. In both cases, the objective is minimizing flow time or make span.

3.1 Test Problem

34 problem instances of job shop scheduling devised by Taillard have been selected as benchmark problems, from 15*15 to 100*20 (the first digit shows

the number of jobs and the second shows number of machines, supposing that all the jobs require all the machines). Objective of these instances are to minimize maximum flow time.

140 combinations of selective dispatching rules have been generated, 10 out of these 140 combinations were those in which only one of the rules had their weights set to one and the other zero, in the other 130 combinations weight were assigned their values randomly such that weights should sum to one.

Each of the instances is run once for each of these 140 combinations and the best results are recorded in Table 1. As shown in the table maximum mean deviation of each instance size from optimal solutions is 18% and by increasing the size of instances the performance of the proposed model improves, for 100*20 instance mean deviation is less than 3% (Fig. 2).

Tab. 1 Result set for benchmarking instances

Taillard ...	Job	Machine	Best known solution	Our model	deviation	Mean Deviation
1	15	15	1231	1371	0.114	0.135
2	15	15	1244	1361	0.094	
3	15	15	1218	1431	0.175	
4	15	15	1175	1368	0.164	
5	15	15	1224	1381	0.128	
11	20	15	1359	1581	0.163	0.171
12	20	15	1367	1583	0.158	
13	20	15	1342	1616	0.204	
14	20	15	1345	1520	0.130	
15	20	15	1339	1607	0.200	
21	20	20	1644	2021	0.229	0.169
22	20	20	1600	1837	0.148	
23	20	20	1557	1804	0.159	
24	20	20	1646	1909	0.160	
25	20	20	1595	1829	0.147	
31	30	15	1764	1986	0.126	0.142
32	30	15	1795	2128	0.186	
33	30	15	1791	2105	0.175	
34	30	15	1829	2088	0.142	
35	30	15	2007	2169	0.081	
41	30	20	2018	2343	0.161	0.173
42	30	20	1949	2334	0.198	
43	30	20	1858	2220	0.195	
44	30	20	1983	2346	0.183	
45	30	20	2000	2257	0.129	
51	50	15	2760	3070	0.112	0.086
52	50	15	2756	3052	0.107	
53	50	15	2717	2906	0.070	
54	50	15	2839	2862	0.008	
55	50	15	2679	3033	0.132	
61	50	20	2868	3185	0.111	0.114
62	50	20	2869	3206	0.117	
71	100	20	5461	5608	0.027	0.030
72	100	20	5181	5352	0.033	

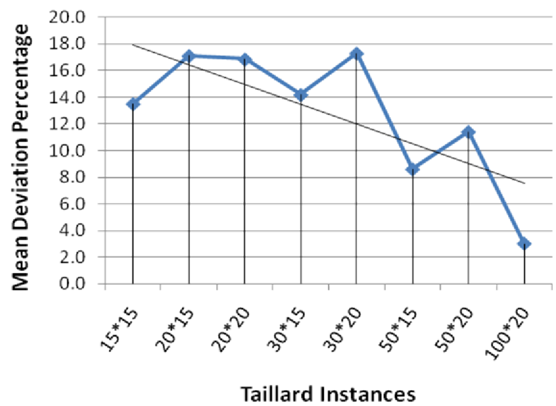

Fig. 2 Deviation trend from best known solution

3.2 Benchmarking by Weckman model

Weckman et al proposed two models based on GA and ANN and tested them by some problem instances illustrating in Fig. 3.

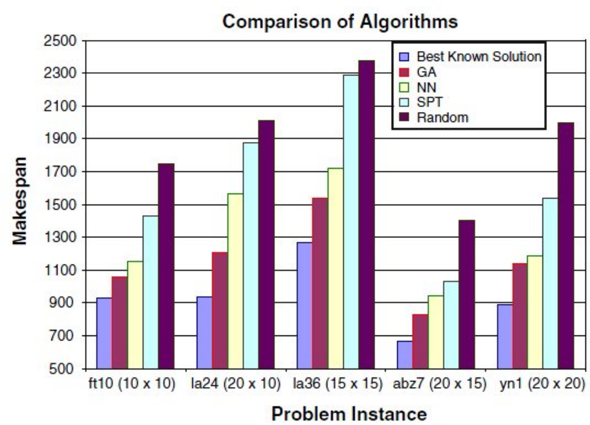

Fig. 3 Illustration of performance of different scheduler

The proposed model of this paper is run for Weckman instances and two result sets are compared. In all instances the proposed model yield better performance than the Weckman, Fig. 4.

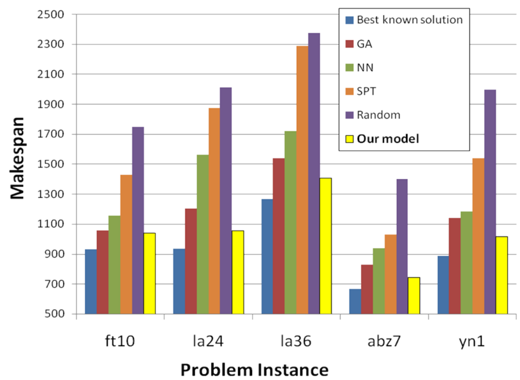

Fig. 4 Comparison of the proposed method with Weckman et al method

4 Conclusion and Suggestion

In this paper application of combination of dispatching rules instead of using one of them is proposed. In order to evaluate the method simulation model is constructed and run for some problem instances and also other reputable models. The result set shows that deviation from optimality is acceptable and also reduces when problem size increases; it means that this method can be used in world. It is suggested to use nonlinear combination of dispatching rules that can generated more widely range of combinations as well as using more of dispatching rules.

Using a meta heuristics based method like GA for generating combinations will improve the performance of the model.

Since the proposed method uses dispatching rules to select the best job to process in each step, similar to what simulation models do, so the method can be a good candidate as a scheduler for simulation models.

The last point is that the method of this paper does not directly relate to any specific objective function so it is possible to apply the method for problems with various objective functions.

5 References

[1] R. K. Baker. Introduction to sequencing and scheduling, 214-230. John Wiley & Sons, 1994.

[2] S. D. Wu, and R. A. Wysk. Multi-pass expert control system – a control/scheduling structure for flexible manufacturing cells, *Journal of Manufacturing Systems, Vol. 7, pp. 107-120, 1998.*

[3] N. Ishii, and J. J. Talavage. A transient-based real-time scheduling algorithm in FMS, *Int'l Journal of Production Research, Vol.29, pp. 2501-2520, 1991.*

[4] N. Ishii, and J. J. Talavage. A mixed dispatching rule approach in FMS scheduling. *Int'l Journal of Flexible Mfg. Systems, Vol. 2, pp. 69-87, 1994*.

[5] H. Cho, and R. A. Wysk. A robust adaptive scheduler for an intelligent workstation controller, *Int'l Journal of Production Research, Vol. 31, pp. 771-789, 1993.*

[6] H. Aytug, G. J. Koehler, and J. L. Snowdon. Genetic learning of dynamic scheduling within a simulation environment. *Computers and Operations Research, Vol. 21, No. 8, pp. 909-925, 1994.*

[7] M. H. Kim, and Y. Kim, Simulation-based real-time scheduling in a flexible manufacturing system, *Journal of Manufacturing Systems, Vol. 13, pp. 85-93, 1994*.

[8] K. C. Jeong, and Y. D. Kim, A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules, *Int Journal of production Research, Vol. 36, No. 9, pp.2609-2626, 1998*.

[9] G. Metan and I. Sabuncuoglu, A simulation based learning mechanism for scheduling systems with continuous control and update structure. *In proceeding of the 2005 Winter simulation conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds, 2087-2093*

[10] E. Kutanoglu, and I, Sabuncuoglu, Experimental Investigation of Iterative Simulation-Based Scheduling in a Dynamic and Stochastic Job Shop. *Journal of Manufacturing Systems, Vol. 20, No. 4, pp. 87-95, 2001*.

[11] L. S. Rong, D. Hang, X. Youlun, and D. Runsheng, Iterative learning scheduling: a combination of optimization and dispatching rules, *Journal of Manufacturing Technology Management, Vol. 15, No. 3, pp. 298-305, 2004*.

[12] A. Klemmt, Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems, *Robot and Computer Integrate Manufacturing, doi:10.1016 /j.rcim. 2009.04.012, 2009*.

[13] R. G. Weckman, V. C. Ganduri and D. A. Koonce. A neural network jobshop scheduler, *Journal of Intelligent Manufacturing, Vol. 19, pp. 191-201, 2008*.