Chapter 15
A Modified Simulated Annealing Algorithm for Optimal Capacity Allocation in Make-to-Order Job-Shops

Liang Huang

Abstract This paper presents a new capacity allocation method to support decisions in the design or redesign of a make-to-order job-shop with stochastic orders and processing times. The solutions for capacity allocation can be adding/removing machines or work shifts at every work stations. A bi-criteria objective function comprising fixed costs and tardiness penalty is used to evaluate each solution. A simulation model is applied to compute the objective function iteratively in a modified simulated annealing procedure until a feasible and profitable solution is generated. Bottleneck analysis is used as guidance for the neighborhood-generation in the modified simulated annealing procedure in order to accelerate convergence. Consequently, the run time of the procedure is short enough for practical use. Different problems were tested. Solutions from the proposed method were compared to those from the classical simulated annealing and the comparison showed relatively positive results.

Keywords Job-shop • Make-to-order • Capacity allocation • Bottleneck analysis • Simulated annealing

Introduction

Many studies focused on the production scheduling to minimize the tardiness of jobs in a make-to-order job-shop. In these studies, it is generally assumed that the capacity at each work station is determined. However, in practice, it is often needs to be changed dynamically by making use of the numerical or empirical outcomes from production scheduling. For example, when too much tardiness of jobs

L. Huang (✉)
School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, People's Republic of China
e-mail: huangliang797@yahoo.com.cn

R. Dou (ed.), *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation(IEMI2012)*,
DOI 10.1007/978-3-642-33012-4_15,

repeatedly occur after proper production scheduling, it is necessary to allocate or reallocate capacities at relevant work stations in order to reduce the tardiness in future production (Yeh 1997; Fry and Russell 1993). This paper will address optimal planning for capacity allocation to support medium to long term (several months to years) decisions under a given production scheduling method in a make-to-order job-shop with stochastic orders and processing times.

For capacity allocation, most problems need to allocate multiple work stations' capacity simultaneously. These are complex combinational optimization problems. Arakawa et al. (2000, 2003) presented a simulation model for job-shop scheduling incorporating capacity adjustment. In their study, a backward/forward hybrid simulation method is used for production scheduling at the first step; and based on the result of scheduling, a pattern search method is used to adjust capacity at the second step. Yang et al. (2005) used the particle swarm optimization (PSO) algorithm for integration of process planning and production scheduling in a job-shop. Some studies use simulation models as well as meta-heuristics algorithms in the design of the manufacturing systems similar to job-shops. Seshadri and Pinedo (1999) presented a framework consist of an optimization model and a simulation model to adjust the capacity for assembly and applied an iterative algorithm using CPLEX 10.2 to deal with the optimization. Shahabudee and Krishnaiah (1999) set the parameters of a multi-product Kanban system using genetic algorithm (GA); the parameters include the number of machines at each work station. In another study of Shahabudeen et al. (2003), they set similar parameters of a multi-product Kanban system using simulated annealing (SA). In all these studies, meta-heuristics algorithms usually use neighborhood search to reach the optimum solution from an initial solution. Coupled with simulation models, many alternatives were examined by simulation in the search procedure. For this reason, they often consume too much time in solving large-scale problems.

In this paper, bottleneck analysis is used as approximate discrete gradients of the objective function of the weighted tardiness. A modified simulated annealing is also presented, in which the neighborhood-generation is guided by the gradients in order to accelerate convergence and reduce the run time of the neighborhood search procedure. Our aim is to make the run time short enough for practical use, even if simulation is performed many times in the search procedure.

Optimization Model

In this study, the alternatives for capacity allocation can be adding/removing machines or work shifts. The available operation hours in regular time, such as working 8 h at daytime, is defined as the capacity of a machine. For example, at a work station, five machines can be allocated at most under the plant space availability. In this way, various numbers of machines can provide five discrete alternatives for capacity allocation from 8 to 40 h per day at the work station.

Array these alternatives according to their capacity from low to high. The alternatives can be denoted by the integral values from 1 to 5.

Therefore, it is assumed that in a general job-shop that consists of m work stations, a linear array $s = [c_1\ c_2\ \ldots\ c_m]$ is the solution vector of the optimization model, where c_j is the alternative number of the capacity level at work station j, for $j = 1, 2, \ldots, m$. Then, the feasible region of s is a set of discrete vectors, denoted as S.

For make-to-order production, weighted tardiness is a general performance measure of job-shops. In this study, one of the purposes of capacity allocation is to fulfill the due dates of all jobs as much as possible. Suppose n jobs belong to p product classes will be manufactured in an m work stations job-shop within a q-months period, we can formulate the first object function to measure the performance of the job-shop in the q-months planning period as follow

$$z^{\mathrm{T}}(s) = \sum_{l=1}^{p} w_l^{\mathrm{TP}} \sum_{i \in I_l} n_i^{\mathrm{LS}} \max(x_i^{\mathrm{C}}(s) - x_i^{\mathrm{D}}, 0), \tag{15.1}$$

where wTP l is the weight on tardiness penalty per unit product and per unit time of class l, I_l is sets of i when job i belongs to class l, nLS i is the lot size of job i, xC $i(s)$ is the completion time of job i in solution s, and xD i is the due date of job i. In the capacity allocation tool, each wTP l is assumed to be a fixed value in the q-months planning period, estimated by the production manager using historical data or practical experience. nLS i, xC $i(s)$ and xD i are generated by the simulation model.

Another purpose of capacity allocation is to reduce the fixed cost, which mainly consists of the depreciation of machines and the fixed salary of operators in this study. The mean monetary values of the depreciation per month and per machine wM j at each work station j were provided by the production manager according to the cost accounting of the workshop. Supposing these values in the q-months planning period will be similar to their historical values, we estimated the fixed cost per month of the job-shop for all solutions s according to the number of machines nM $j(s)$. Then, the second objective function is

$$z^{\mathrm{C}}(s) = q \sum_{j=1}^{m} w_j^{\mathrm{M}} n_j^{\mathrm{M}}(s). \tag{15.2}$$

The two objective functions are both considered in this study to get a feasible and profitable solution for a practical use. Hence, the optimization model with a bi-criteria objective function is

$$\min z^{\mathrm{T}}(s) + z^{\mathrm{C}}(s) \tag{15.3}$$

$$\text{subject to} : s \in S. \tag{15.4}$$

Gradient-Based Simulated Annealing

Kirkpatrick et al. (1983) firstly presented SA in 1983. In its neighborhood search, SA accepts inferior solutions according to a probability in order to bypass local optimums. Thus, in this study, we couple the gradient-based method with SA and present a hybrid method named GBSA to optimizing capacity allocation. The GBSA has not only the capability of avoiding local minima, but also a higher speed of convergence to approach stationary compared to the traditional SA.

Step 1: Input the control parameters of the GBSA: Initial Temperature T_i, Termination Temperature T_f, Cooling Rate α, Freeze Limit Φ, and Accept Limit β. Take T_i as current temperature T. Generate initial solution s_0. A simulation is performed to compute the object function value z_0 in solution s_0. In this study, the initial solution s_0 was set to be 1.2 times (an empirical value from the practical case) of the mean capacity requirement per day in the tested cases.

Step 2: Detect the bottlenecks in the job-shop. To detect and measure the shifting bottlenecks in a job-shop, a statistical method called the active period method has been presented by Roser et al. (2002). They proposed that at any given time the momentary bottleneck is the machine with the longest uninterrupted active period at this time and in any given period of time the average bottlenecks can be measured by the percentage of the time that a work station. Although this method is not an exact one, it is very robust, easy to apply and has the ability to detect the bottlenecks in steady state systems or non-steady state systems.

Step 3: Suppose there are n_s solutions neighbor to the current solution s_0 in the feasible region N^+. They are denoted as h_k ($k = 1, 2, \ldots, n_s$). In this step, "neighbor to" means only one element is +1 or −1. If the neighborhood h_k is a solution to add machines to work station j, let $p_k = b_j$; otherwise, let $p_k = -b_j$. Denote the minimum in p_k as $p_{\min}$. We select a new solution s_1 from the neighborhoods of s_0 according to a probability shown as follows:

$$P(s_1 = h_k) = \frac{(p_k - p_{\min})^{\gamma}}{\sum_{k=1}^{n_s} (p_k - p_{\min})^{\gamma}}. \tag{15.5}$$

Therefore, the neighbor of a better estimated objective-function value has a higher probability to be chosen in order to accelerate convergence. Parameter γ in (15.5) is used to adjust the influence of the bottleneck analysis in the search procedure. Based on pilot experiments, we observe that when the objective-function value has a large improvement in the previous iteration indicating that the guidance of the gradient works well at this stage of the search procedure, γ should be set to a larger value to make full use of the guidance of the gradient, or else γ should be set to a smaller value to have a better chance to move from one local minimum area to another one. For this consideration, in this study γ is set to 1 at the beginning of the search procedure and will be adjusted at each iteration as stated in Step 4.

Step 4: Calculate the objective function value z_1 in the new solution s_1 through a simulation. Let $\Delta z = z_1 - z_0$. If $\Delta z < 0$, the current solution s_0 will be replaced by the new solution s_1; otherwise, apply a probability $P(A) = e^{-\Delta z/T}$ to determine whether the replication should be performed. Set $\gamma = |\Delta z|/(|\Delta z|)_{\max}$, where $(|\Delta z|)_{\max}$ is the maximum among all the $|\Delta z|$ values in the past iterations.

Step 5: The current temperature T is adjusted after every Φ iterations according to α. If it's below T_f or the solution has not been improved for too many consecutive iterations to overstep β, stop the search produce; otherwise, go to Step 2.

Step 6: Report s_0 and z_0 as the final solution and its objective function value, respectively.

In the proposed GBSA, the neighborhood-generation is not a random produce like that in the traditional SA, but controlled by the results of the bottleneck analysis. And γ will be changed at each iteration according to the improvement of the objective-function value. These modifications speed up the search for a better solution in the area with the most potential while still allows the search to move away from a local area to another. Thus, the neighborhood search may stop earlier as controlled by β and the computing time is reduced.

Computational Experiments

In this paper, three case studies are tested using our proposed GBSA. Case 1 consists of 3 types of orders and 5 work stations, Case 2 consists of 5 types of orders and 10 work stations, and Case 3 consists of 15 types of orders and 30 work stations, respectively. In this paper, only the data of Case 1 to be given in detail for the space constraints.

In Case 1, there are 3–10 machines at each of the five work stations. The scheduling method used in this workshop is a dispatching rule, earliest due date with the tie broken by first come first service (EDD/FCFS), for it is very easy to be applied in a dynamic job-shop with stochastic demand and processing times. Within a work station, the scheduling is complex in this workshop. For we have not enough detailed records about it, according to the production manager's suggestion, we make an assumption that a task can always make full use of the capacity within a work station and the processing time of the tasks processed at the work station will decrease/increase linearly with adding/removing capacity to the work station.

In the simulation model, inter arrival times of the orders and processing times of the tasks are generated in exponential distributions; constraints of lead times, tardiness penalties per hour and depreciation of machines are set to be fixed values. These data is shown in Tables 15.1 and 15.2.

The simulation software was developed in Microsoft SQL2000. In all the cases, the simulation for any given solution was performed in the duration of 25,000 h. The simulations were all performed in a personal Pentium IV computer with 2.4G

Table 15.1 Demand requirements and tardiness penalties in Case 1

Product type	Mean inter arrival time of orders (hour)	Constraints of lead time (hour)	Tardiness penalty (RMB/hour)
1	40	50	20
2	60	60	15
3	80	70	10

Table 15.2 Processing times and depreciation of machines in Case 1

Product type	Work station	Mean processing time (hour)	Depreciation of machines (1,000 RMB)
1	1	2.25	20
	2	2.00	20
	3	2.50	10
2	1	1.25	7.5
	2	1.25	7.5
	3	2.00	15
3	1	1.75	10
	2	1.25	7.5
	3	2.25	10

Table 15.3 Control parameters

		Control parameter values				
	Type	T_i	T_f	α	Φ	β
A1	GBSA	1	0.1	0.9	10	20
A2	GBSA	1	0.3	0.7	5	10
A3	Traditional SA	1	0.1	0.9	10	20
A4	Traditional SA	1	0.3	0.7	5	10

Table 15.4 Results of the computational experiments

	Objective function value (1,000 RMB)			Run time (minute)		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
A1	1,053	1,794	4,250	9.64	14.59	42.37
A2	1,053	1,794	4,287	6.13	12.95	37.08
A3	1,053	1,826	4,420	38.31	76.45	225.54
A4	1,053	1,815	4,587	31.66	52.21	89.40

CPU and 1G memory. The mean simulation time of each simulation (including the time for bottleneck analysis) is 35 s in Case 1.

According to the pilot runs, two groups of control parameters are used to both traditional SA and GBSA. Therefore, there are four kinds of algorithm with different control parameter values or different neighborhood-generation methods applied to Case 1, 2, and 3, which is denoted as A1, A2, A3, and A4. Their control parameter values are shown in Table 15.3. The results of the three cases are shown in Table 15.4.

Conclusions

In this paper, a modified SA, named GBSA, is used as an optimization tool to optimize capacity allocation in make-to-order job-shops. Although the optimums of all the algorithms equip to each other in Case 1, the proposed GBSA used noticeably smaller computing time than the traditional SA. Moreover, with less computing time, GBSA found better solutions in Case 2 and 3 compared to the traditional SA. These results show that the proposed method can often finds better solutions with a shorter computation time compared to the traditional method. These optimal solutions for capacity allocation can be very useful to support decisions in performing tradeoffs between the tardiness penalty and the cost of capacity allocation.

References

Arakawa M, Fuyuki M, Nakanishi H, Inoue I (2000) A simulation-based capacity adjustment method for job shop production scheduling. In: Proceedings of the third Asia-Pacific Conference on Industrial Engineering and Management Systems (APIEMS'2000), Hong Kong, China, pp 84–90

Arakawa M, Fuyuki M, Nakanishi H (2003) An optimization-oriented method for simulation-based job shop scheduling incorporating capacity adjustment function. Int J Prod Econ 85 (3):359–369

Fry TD, Russell GR (1993) Capacity allocation strategies in a hypothetical job-shop. Int J Prod Res 31(5):1097–1115

Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 13:671–680

Roser C, Nakano M, Tanaka M (2002) Shifting bottleneck detection. In: Proceedings of the 2002 winter simulation conference, New York, USA, pp 1079–1086

Seshadri S, Pinedo M (1999) Optimal allocation of resources in a job shop environment. IIE Trans 31(3):195–206

Shahabudee P, Krishnaiah K (1999) Design of a bi-criteria kanban system using genetic algorithm. Int J Manag Syst 15(3):257–274

Shahabudeen P, Krishnaiah K, Thulasi Narayanan M (2003) Design of a two-card dynamic kanban system using a simulated annealing algorithm. Int J Manag Syst 21(10–11):754–759

Yang YH, Zhao FQ, Hong Y, Yu DM (2005) Integration of process planning and production scheduling with particle swarm optimization (PSO) algorithm and fuzzy inference systems. In: Proceedings of the ICMIT 2005: control systems and robotics, Chongqing, China, pp 2292–2297

Yeh CH (1997) Schedule based production. Int J Prod Econ 51(3):235–242