대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

Quasi Assignment Algorithms in Job Shops

Eui-Seok Byeon, Jang-Yong Lee

Department of Industrial Engineering, SunMoon University
#100, Kalsanri, Tangjeongmyeon, Asansi, Chungnam 336-840, Korea

Abstract

Production scheduling has been one of the most critical issues in a manufacturing environment. Job-shop scheduling problems(JSP) are well known from the standpoint of production planning and operations control. In this research scheduling against due date is a measure of performance and the objective is minimizing total weighted tardiness. This paper presents an idea of decomposition of the problem and shows robustness of the schedule under various disturbances along with exact and approximation methods. The proposed method can indeed handle shop disturbances more effectively when compared with traditional and dynamic scheduling methods.

1. Introduction

Job shop scheduling often has to face various disturbances in real world applications. Disturbances are unplanned events such as processing time deviations, machine breakdowns, etc. These events undoubtedly cause scheduling performance to degrade or rendered useless. Clearly the less a schedule is sensitive to disturbances, the more robust it is. In this paper, improving schedule robustness is done by increasing its flexibility and reducing its sensitivity to disturbances. The ability of a schedule to handle disturbances depends on flexibility that exists in the schedule and the information available about the disturbances. Leon et al. [5] defined a robust schedule as one that is insensitive to unforeseen shop floor disturbances. Moreover, it maintains good performance in the presence of these disruptions.

Other research in job-shop scheduling problem has focused on the development of dispatching rules which are not sensitive to varying shop conditions or performance measures [4,6,7]. Dispatching in a job-shop is a decision of selecting from a machine queue a job to be processed next. As a result, a scheduling heuristic determines how the jobs at each machine should be sequenced. In earlier researches [3,8], the method which decomposes scheduling decisions into two different stages has been proposed: a first stage makes “partial” scheduling decision by partitioning the operations into subsets. The second stage completes the scheduling decision over time in a dynamic fashion. The method provides an additional flexibility in scheduling and thereby increases its robustness. In this paper, we show a detailed simulation study of the above decomposition scheme and demonstrate the robustness of the method based on its performance under a wide range of processing time variations.

2. Problem Statement

We have developed a decomposition scheme where the operation of a Jobshop Scheduling Problem (JSP) is assigned to a number of subsets(p) each with a size of α_k. Each subset does not necessarily have the same size. Associated with a JSP is a disjunctive graph G (N,A,E) where A is a set of conjunctive arcs representing the precedence constrains among operations, and is a set of disjunctive arcs representing the possible processing orders on machine m, $E= \cup_m E_m$.

If we try to assign each of |N| operations into a predetermined number of subsets p, the corresponding formulation is for variation of assignment problem(VAP) as follows:

$$\textbf{(VAP): Minimize } \sum_i \sum_k C_{ik} X_{ik} \qquad (1)$$

$$\text{s.t. } \sum k X_{ik} = 1 \qquad i \in N \qquad (2)$$

$$\sum i X_{ik} = \alpha_k \qquad k=1,\ldots p \qquad (3)$$

$$X_{jl} \leq \sum_k X_{ik} \qquad (i,j) \in A \text{ and } l=1,\ldots p \qquad (4)$$

$$X_{ik} \in \{0,1\} \qquad i \in N \text{ and } k=1,\ldots p \qquad (5)$$

In this formulation, C_{iK} is called a cost incurred for assigning operation i to subset k. XiK is the decision variable: $X_{iK}=1$ if operation i is assigned to subset k, $X_{iK}=0$ otherwise. The above problem is a variation of the classical assignment problem (VAP). Solving VAP resolves a part of the disjunctive arcs in the set E. Specially, for a disjunctive arc (i,j) with i,j not assigned to a same subset will have its direction resolved. Suppose E_{Km} represents the remaining, unresolved set of disjunctive arcs in subset k, on machine m, then $E_{Km}=\{(i,j) \in E_m \mid X_{ik}=X_{jk}=1\}$. After solving VAP on the original graph, G(N,A,E), we have a new disjunctive graph, $\overline{G}(N, \overline{A}, \overline{E})$, where $\overline{A}$ consists of the newly resolved disjunctive arcs as well as the original conjunctive arcs. $\overline{E}$ is the union of E_{km} for all m and k. Therefore, the new graph $\overline{G}$ represents the original JSP with an additional set of precedence constraints. This new JSP makes global scheduling decision while the remaining sequencing decisions are allowed great flexibility through the use of dynamic dispatching. In the rest of the paper, the robustness of "VAP schedules" via intensive simulation is demonstrated.

3. Experiments

The objective of this study is to examine how the proposed decomposition scheme performs when operation processing times vary. The proposed method is simulated and compared with a static and a dynamic scheduling method. We measure the mean weighted tardiness (wT) of the schedules after 100 simulation runs, and use it as the robustness measure. Disturbances are generated by altering the processing time by an exponential distribution. We have tested a wide range of disturbances by varying the mean (τ) of the exponential from 5 to 60, incremented by 5. The actual processing times (Pi') for each operation are generated as follows: $P_i'=P_i \pm Exp(1/\tau)$, where $\tau=5,10,15,\ldots 60$. Note that since Pi' could be negative, assume an unit processing time when $P_i' \leq 0$.

3.1 VAP Heuristic

As a solution procedure for VAP, use a heuristic which assigns operations to subsets based on a priority index. Specifically, we use an "ATC-index" due to Vepsalainen and Morton [7]. Given a job duedate(dd_J) and weight (w_J), the index at the current time t is computed as follows:

$$\frac{Wj}{Pi}\exp\left[-\left(\frac{dd_j - t - P_i - \sum g \in \psi_i (w_g + p_g)}{Kp}\right)\right]$$

Where, operation i belongs to job J while having a set of successors Ψi. The p is the average processing time, k is a look-ahead parameter, and set k=3. W is a leadtime estimation parameter as in Vepsalainen and Morton [7], and set to 2. Based on the above index, assign the first highest α_1 operations to subset 1, and the second highest α_2 operations to subset 2, and so on. Each heuristic assignment is evaluated based on the lower bound of its corresponding scheduling problem. The procedure is as follows:

Iterative Searching Procedure

Step 0. Set Itrn=1, and denote NTI be the number of total Iterations. Compute ATC index using original duedate and weight. Goto Step 2.

Step 1. Compute ATC index using updated duedate and weight. Goto Step 3.

Step 2. Assing the first highest α_1 operations to subset 1, and the second highest $\alpha 2$ operations to subset 2, and so on. Compute Lower Bound of above graph.

Step 3. If Itrn <NTI, then goto Step 4. Otherwise, stop.

Step 4. Update duedates as follows :

$$dd^{t+1} = dd^t(j) + \text{stepsize} \times dd^t(j) \times \left(\frac{SLACKj}{SS}\right)$$

Goto Step 1. End.

The stepsize ranges from 0.5% to 3.0% through extensive empirical testing. The SS is total job slacks, where SLACKj=ddj-completion time(Cj).

The above heuristic generates half completed schedules while the detailed schedule is generated dynamically. We complete this dynamic schedule using the ATC dispatching rule. The ATC heuristic by Vepsalainen and Morton [7] is a dynamic dispatching rule with priority updated each time a job is scheduled. Their priority index for each unscheduled job is a function of weight, duedate and processing time. In this paper, the robustness of the VAP heuristic in the presence of processing time variations is tested. Simulation results are compared with pre-generated static schedule and Morton's dynamic schedule as follows.

3.2 Static Schedule

We generate a static schedule using the ATC heuristic at the beginning of the planning horizon, before disturbances occur.

3.3 Dynamic Schedule

Here, a schedule using ATC-rule is generated, but all the scheduling decisions are made dynamically. That is, the dynamic schedule updates its parameters over time based on varied processing times. We use three sets of test problems for the simulation study. All test problems are from Applegate and Cook[1]. The first set contains 10×10 problems. Knowing the optimal makespan schedule for the above problems, subtract a constant(i.e. 100) from each job's completion time, and set them as job duedate. The second and third sets are larger sized problems, i.e. 20×15 and 30×10. In these cases, the optimal makespan schedule is not available. Thus set duedate as follows: first generate a non-delay schedule[2] and obtain the job completion times. Subtract 100 from each job completion time, and use them as job duedates.

4. Results

The VAP heuristic simulated is characterized by two primary factors; the number of subsets, and the size of each subset α_k. We tested a various number of subsets including 2,3,4,5 and 10. Then, given a specific number of subsets, we varied run. The number after "W" stands for the number of subset used. For example, in A the W5 column represents the simulation result of VAP with 5 subsets of identical sizes(i.e., 20 operations per subset). For subsets with non-identical sizes, we used additional numbers after "W". For example, W225 represents 2 subsets where 25 operations are in the first subset and 75 operations in the second. Similarly, W212 in B and C represents 100 operations in the first subset and 200 in the second.

5. Conclusions

In summary, more robust schedule could be indeed generated when compared to the traditional

	Problem	Static	VAP Heuristic(τ=15)						Dynamic
	(10×10)		W2	W225	W275	W4	W5	W10	
A	abz5	11770	10595	11924	**9814**	12387	13255	11967	10376
A	la 16	4480	**4414**	4716	4554	4871	4510	4736	4895
A	la 17	5598	4904	5050	4915	5086	5901	5423	**4102**
A	la 20	8368	6019	5428	**4493**	6188	6613	6089	5802
A	orb3	12175	8592	9376	10175	**8319**	8741	9589	9590
A	orb6	7331	8554	5038	6551	5871	**4940**	7064	7280
	Problem	Static	VAP Heuristic(τ=30)						Dynamic
	(30×10)		W2	W212	W221	W3	W5	W10	
B	la 31	91459	**34504**	44443	41574	50521	75511	99701	35593
B	la 32	75855	**33685**	50139	47002	51934	87166	88005	40936
B	la 35	79567	**38930**	55463	49302	61499	74854	117048	39554
	Problem	Static	VAP Heuristic(τ=10)						Dynamic
	(30×10)		W2	W212	W221	W3	W5	W10	
C	abz7	20501	14288	17218	13474	16727	21449	20545	**12649**
C	abz8	23165	21041	22533	17641	21736	23740	26629	**16627**
C	abz9	23012	14822	18845	15115	20553	23968	25104	**12963**

<Table 1>Mean Weighted Tardiness after 100 simulation Runs

the subset sizes. For example, for 10×10 problems with two subsets, we consider the following splits: [50,50], [25,75] and [75,25]. Table 1 shows the mean weighted tardiness(WT) from 100 simulations static and the dynamic methods. We have demonstrated through simulation that the VAP scheduling heuristic can manage a wide range of disturbances due to its embedded flexibility.

6. References

[1] Applegate, D and W. Cook, " A computational Study of the Jobshop Scheduling Problem," ORSA Journal on Computing, Vol. 3, 1991, pp. 149-156.

[2] Baker, K.R., Introduction to Sequencing and Scheduling, John Wiley & Sons, 1974.

[3] Byeon, E.S., S.D. Wu and R.H. Storer, "Decomposition Heuristics for Robust Jobshop Scheduling," IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, 1998, pp. 303-313.

[4] Dooley, K.J. and F. Mahmoodi, "Identification of Robust Scheduling Heuristics : Application of Taguchi Method in Simulation Studies," Computers and Industrial Engineering, Vol. 22, No. 4, 1992, pp. 359-368.

[5] Leon, V.J., S.D. Wu and R.H. Strorer, "Robustness Measures and Robust Scheduling for Job Shops," IIE Transactions, Vol. 26, No. 5, 1994, pp. 32-43.

[6] Philipoom, P.R. and T.D. Fry, "The Robustness of selected Jobshop Dispatching Rules with respect to Load Balance and Work Flow Structure," Journal of Operational Research Society, Vol. 41, No. 10, 1990, pp. 897-906.

[7] Vepsalainen, A.P. and T.E. Morton, "Priority Rules for Job Shops with Weighted Tardy Costs," Management Science, Vol. 33, No. 8, 1987, pp. 95-103.

[8] Wu, S.D., E.S. Byeon and R.H. Storer, "A Graph Theoretic Decomposition of Jobshop Scheduling Problems to Achieve Scheduling Robustness." Operations Research, Vol. 47, No. 1, 1999, pp. 113-124.