SHEET METAL FABRICATION SCHEDULING USING SELECTIVE PERFORMANCE MEASURE AND PRIORITY DISPATCHING RULES

M.R.Osman, N.Ismail, M. R. M. Zaihirain, R.M.Yusuff, S. M. Sapuan, and M.M.H.M. Ahmad
Department of Mechanical and Manufacturing Engineering,
Universiti Putra Malaysia, Serdang, Malaysia
E-mail: rasid@eng.upm.edu.my

ABSTRACT

In today's highly competitive market, on-time delivery performance is very crucial for job shop manufacturing companies to maintain and continue to increase their market share. Quick and effective scheduling contributes to better manufacturing productivity by reducing lead time, reducing inventory, improving on-time delivery and prolong good relationship with customers. Scheduling process, which is to organise, maintain, update and reschedule jobs, is a very tedious work and time consuming. To plot the charts manually is not a practical solution. Due to the combinatorial nature of scheduling problems, it is often difficult to evaluate all possible schedules. Identifying the performance measures to be used in selecting the schedule is important. A logical strategy is thus to pursue methods that can consistently generate good schedules with quantifiable quality in a computationally efficient manner. A window-based scheduling program, developed in this paper is capable to assist planner to quickly generate a better schedule. The program is designed using multiple performance measures with various dispatching rules.

Key words: *Sequencing and Scheduling, Selective Performance Measure, Priority Dispatching Rules*

INTRODUCTION

In today's highly competitive markets, companies have come to view customer satisfaction as a key to maintaining and increasing their market share. One of the most important measures of the quality of service a company provides is on-time delivery performance. Orders delivered after a promised due date will result in lost customer good-will and, ultimately, in lost market share. This is particularly true for small make-to-order companies whose business is based on producing a special product to customer specification rather than on producing standardised parts to stock [1].

Scheduling is a key factor for manufacturing productivity. Effective scheduling will improve on-time delivery, reduce inventory, cut lead times, and improve the utilisation of bottleneck resources. Because of the combinatorial nature of scheduling problems, it is often difficult to obtain optimal schedules, especially within a limited amount of computation time [2].

The whole scheduling process is a very tedious work and time consuming especially to organise, maintain, update and reschedule the jobs manually. After having all the required information on process flow and estimated process time, the planner has to plan the job so that it meets the customers' requirements or at least to know in advance the expected completion date. A planner has to decide on which job to run first and plot the schedule. The process of deciding the sequence and plotting the operation hours on Gantt chart alone is very time consuming. On the other hand, when a few jobs come in at the same time, it is difficult to schedule which one to run first as it is difficult to see the actual progress of jobs which are still running. In deciding which priority dispatching rule (PDR) to use – first come first serve (FCFS), shortest processing time (SPT), earliest due date (EDD) or critical ratio (CR) will provide the best performance measure. To run all the PDRs by manually plotting the Gantt chart is not only expensive but also very time consuming.

For n x m job shop scheduling problems, where n is the number of jobs and m is the number of tasks per job, there are $(n!)^m$ possible schedules for n jobs, each requiring m machines. Plotting the time charts manually is not a very practical solution and laborious. Thus, a logical strategy is thus to pursue methods that can consistently generate good schedules with quantifiable quality in a computationally efficient manner.

Scheduling is a very dynamic process where one has to update the progress and reschedule if necessary. Failing to carry out this process periodically will make it difficult to oversee the whole movement of jobs, machine

ISSN 1823-1039 © 2004 FEIIC

allocation as well as to monitor the job progress and the delivery schedule. Therefore, an automatic process is very much required to assist the planner to generate a better schedule and thus generate efficient scheduling.

This paper discusses the development of a scheduling and sequencing program using selective performance measure by utilising various priority dispatching rules. The scheduling program is to schedule n jobs concurrently by developing a schedule for each part traveling among the machines with the objective specified by the performance measures. The program is to select the machines and beginning times for individual operations to achieve certain performance measure(s).

Job Shop Scheduling

Adams et al. [3], Carlier and Pinson [4] and Mcmahon and Florian [5] describe job shop scheduling problem as finding a schedule for processing n jobs by m machines. Each job consists of a sequence of operations, which may have different processing times. Operations in a job must be processed in a specified order, and each machine can process only one operation at a time. The objective is to find the minimum time to complete all the jobs.

Chu and Ngai [6] point out that flow job scheduling is a still more complicated form of production scheduling problem. Flow job scheduling is when each job consists of a certain number of operations is processed by a different machine with certain processing time where the order of processing is the same for all jobs.

Banga et. al. [7] and Riggs [8] pointed out that sequencing problem quickly become more tedious as the number of jobs and machines increases. As an example, with n jobs passing only from machine 1 to machine 2, there are n! alternatives. It would take almost 40 million time charts to show all the sequence patterns possible for just 11 jobs. Thus, charting is not a very practical solution tool for larger exercises. More complex problems are treated by simulation techniques, but the cost and time required to produce a satisfactory solution are still very large.

According to Krajewski and Ritzman [9], identifying the performance measures to be used in selecting the schedule is important. The schedules should reflect managerially acceptable performance measures. Some of the most common performance measures used in operation scheduling are job flow time, makespan, past due, work-in-process-inventory, total inventory and utilization.

There has been a large amount of works on scheduling problems. For over fifty years, a variety of scheduling problems and solution methods have been proposed and evaluated. According to Jain and Meeran [10], Johnson who develops an efficient algorithm for a simple two machine flow shop which minimises the maximum flow time back in 1954 is probably the earliest work in scheduling theory using efficient method. French [11] predicts that no efficient algorithms will ever be developed for the majority of scheduling problems and also expresses the view that an integer programming formulation of scheduling problems is computationally infeasible.

Adams et. al. [3] introduce shifting bottleneck procedure (SBP) which utilises the advent of more powerful computers. The main contribution of this approach is the way the one machine relaxation is used to decide the order in which machines should be scheduled. A general weakness of the SBP approach is the level of programmer sophistication required and the whole procedure has to be completed before a solution is obtained.

Morton and Pentico [12] describe PDR as a very popular technique applied to solve job shop scheduling problems. This is due to their ease of implementation and their substantially reduced computational requirement.

Ashby and Uzsoy [1] develop a set of scheduling heuristics that integrate order release, group scheduling, and order sequencing for a make-to-order manufacturing facility organised into group technology cells. The results show that at least in certain production environments order release can significantly improve due date performance and that a poor choice of dispatching rule can entirely negate the benefits of a good order release policy. However, extra work is required to show how well the order release rules developed by Ashby and Uzsoy [1] would perform in more complex production setting, that is processes that involve more than a single stage.

Chang, et. al. [13] carry out a comparative study on PDR by evaluating the performance of 42 PDRs using a linear programming model. Their analysis indicates that the shortest processing time (SPT) related rules consistently perform well while the longest processing time (LPT) based rules consistently perform badly.

Sun and Nobel [14] develop a method for scheduling job shop with sequence-dependent set up times based on concepts from the shifting bottleneck procedure. It offers solution to job shop scheduling problems with sequence-dependent setups and due date related performance measures. This method requires increase in the computational time and only considers due date as its performance measure.

ElMaraghy et. al. [15] conduct a study using dispatching rules with various performance measures. The developed scheduling approach and formulation proved to be useful for optimising production performance, and can be used to define a priori the best dispatching rules and schedules for a given set of production requirements and objectives.

Veral [16] proposes a methodology to establish tight and reliable estimates for job flow times. By recognising that the difference priority rules and shop characteristics result in different job flow patterns, Veral [16] attempted to improve due-date performance without loosening due-dates. The study only limits its scope for performance comparisons with various due date setting rules.

There is agreement in the literature that due to the combinatorial nature of scheduling problem, it is often difficult to evaluate all possible schedules. Scheduling process is a very tedious work and time consuming. Plotting the time charts manually is not a very practical solution and to evaluate the whole possible time charts is not computationally efficient.

The computational costs rise rapidly as a function of problem complexity even when optimal solutions are not being sought. As the number of jobs to be scheduled and the number of machines increase, the time required to prepare a schedule increases much more rapidly. Thus the size of many actual manufacturing shops means that many of the techniques used for research are not computationally economical for solving day-to-day scheduling problems. The computational costs of simulating all job arrivals and all possible schedules for each machine and worker can be prohibitively high for many actual applications. Therefore, a schedule that is the best possible near optimal solution is often acceptable.

Glover and Greenberg [17] indicate that PDR approach is able to attain near optimal solutions, within moderate computing times and is more suitable for larger problem. Therefore, priority dispatching rule technique with selected performance measure is used to develop the sequencing and scheduling program. PDR approach is preferred because it offers a substantially reduced computational requirement and ease of implementation as mentioned by Morton and Pentico [12].

METHODOLOGY

This section outlines in details the methodology that had been adopted and carried out in the study covering scheduling process, variable definition, constraints, approach and the algorithms used.

Problem Formulation

Generally, in sheet metal fabrication activities, there are five main processes involve in completing the job namely shearing, turret punching, laser cutting, bending and welding. Table 1 shows the five machines use in the developing the program. Process flow of a product is developed during cost estimation stage. The manufacturing cost for the part is directly based on the estimated processing time required at each machine. The estimated processing time for each particular product was derived based on its cutting length, shape to be cut, number of punches, number of bending, type of welding, welding length and finishing quality.

Once the required information on process flow and estimated process time has been collected, the planner has to plan the job so that it meets the customer's requirement or at least to know in advance the expected completion date. The planner has to decide which job to run first and plot the schedule. This process has to be repeated several times based on certain priority rule. Most of the times, the planner will use first come first serve rule to sequence the jobs. The process of deciding the sequence and plotting the operation hours to Gantt chart alone is very time consuming. To simulate all the PDRs by manually plotting the Gantt chart is not only expensive but also very time consuming. The planner does not have the time to run all the possible options.

Table 1: Type of process and quantity of machines

Process	*No. of machines*
Shearing	*1*
Laser cutting	*1*
Turret punching	*1*
Bending	*1*
Welding	*1*

Scheduling is a very dynamic process where one has to update the progress and reschedule if necessary. Failing to carry out this process periodically will make it difficult to oversee the whole movement of jobs, machine allocation as well as to monitor the job progress and the delivery schedule. Efficient scheduling and job sequencing at each process or machine is very crucial. Therefore, a logical approach is thus to pursue methods that can consistently generate good schedules with quantifiable quality in a computationally efficient manner. Hence, an automatic process is very much required to assist the planner to generate a better schedule and thus generate efficient scheduling.

Table 2 shows a sample of information required for each job such as job number, job descriptions, available date, due date, process hours at each machine, schedule start date and preferred performance measure. The variables defined in the study and the formulations of these methods are shown in Table 3.

Table 2: Sample of input data

Job no.	*Descriptions*	*Avail. Date*	*Due date*	*Process hour [t(i,j)]*				
				Shear	*Laser*	*Turret*	*Bend*	*Weld*
J(i)		*AD(i)*	*DD(i)*	*t(i,1)*	*t(i,2)*	*t(i,3)*	*t(i,4)*	*t(i,5)*
790	*Toll fare indicator*	*28/09/2000*	*23/10/2000*	*5*	*12*	*24*	*5*	*6*
795	*Air diffuser*	*02/10/2000*	*16/10/2000*	*5*	*0*	*0*	*2*	*20*
801	*Dolly muffler*	*03/10/2000*	*06/10/2000*	*2*	*8*	*0*	*0*	*0*
816	*Heater clamp*	*16/10/2000*	*16/10/2000*	*0*	*4*	*0*	*2*	*0*
818	*Chair stand*	*16/10/2000*	*24/10/2000*	*0*	*8*	*0*	*0*	*0*

Schedule start date : 2/10/2000
Performance measure : Minimise average job flow time

The constraints considered in this paper are as follows:

1. Each job consists of operation with a known process sequence and processing time.
2. Each job is processed at any particular machine only once.
3. One machine can only process one job at one time. No two activities requiring the same machine may execute at the same time.
4. Machine availability constraint. Allocate job at the earliest machine available time.
5. Average machine setup time is imposed to a new job at each machine once started. Average setup time at each machine is based on user input.
6. Average setup factor is imposed upon each job. This is to make sure that the operation starts immediately after the set up is done. Average setup factor at each machine is based on user input.
7. Handling time is included in the machine process time. Transportation is negligible because distance from machine to machine is very near.
8. For big quantity lot size, batching is applied. User has to split lot into smaller lot quantity.

Table 3: Variable definition

Let,		
J(i)	=	*Job number i*
AD(i)	=	*Available date of job i*
DD(i)	=	*Due date for job i*
M(j)	=	*Machine number j*
t(i,j)	=	*Process hour for job i at machine j*
SSD	=	*Schedule start date*
S(j)	=	*Standard setup hours at machine j*
T(i,j)	=	*Operation hours of job i at machine j*
N(i)	=	*Number of processes for job i*
T(i)	=	*Total process hour for job i at all machines*
CR(i)	=	*Critical ratio for job i*
SRO(i)	=	*Slack per remaining operation for job i*
TC(i)	=	*Total process hour for job i which has been completed*
NC(i)	=	*Number of processes for job i which has been completed*
ESD(k,j)	=	*Early start date of job k at machine j*
MAD(j)	=	*Earliest available time at machine j*
SD(k,j)	=	*Start date of job k at machine j*
FD(k)	=	*Finish date of job k*
JFT(k)	=	*Job flow time of job k*
AJFT	=	*Average job flow time*
Max{k}	=	*Total number of jobs*
DE(k)	=	*Number of days early for job k*
DPD(k)	=	*Number of days past due for job k*
NJPD(k)	=	*Number of jobs past due*

Where, i = 1, .., n ; k = 1, .., n and j = 1, .., m
Task:

1. *Given information such as J(i), AD(i), DD(i), t(i,j), S(j) and SSD find the best scheduling arrangement based on selected performance measure by using available priority dispatching rules.*

System Design

In scheduling multiple workstation, each operation is treated independently. When a workstation becomes idle, priority rule is applied to the job waiting for that operation. The one with the highest priority is selected. When that operation is completed, the job is moved to the next operation.

PDR technique is adopted in the paper. According to Morton and Pentico [12], PDR offers a substantially reduced computational requirements and ease of implementation. Hence, in this paper a sequencing and scheduling program is developed to assist the planner to quickly generate a better schedule.

Figure 1 shows the block diagram of the scheduling program. The user is only required to provide a minimum input consisting of job number, customer name, job description, process time, schedule start date and their preferred performance measure. Based on these inputs, the system automatically calculates the PDR index and sequences the jobs. The system then generates the schedule by allocating machine-hours to each job. Once the schedule has been generated, the system calculates the performance measure of each generated schedule with difference PDR and analyses them. The system then recommends and generates the most optimum schedule solution based on selected performance measure. In short, the system is able to generate all the possible schedules with different PDR, analyses the performance measure of each schedule generated and recommends the best option. Hence an optimum schedule is generated.

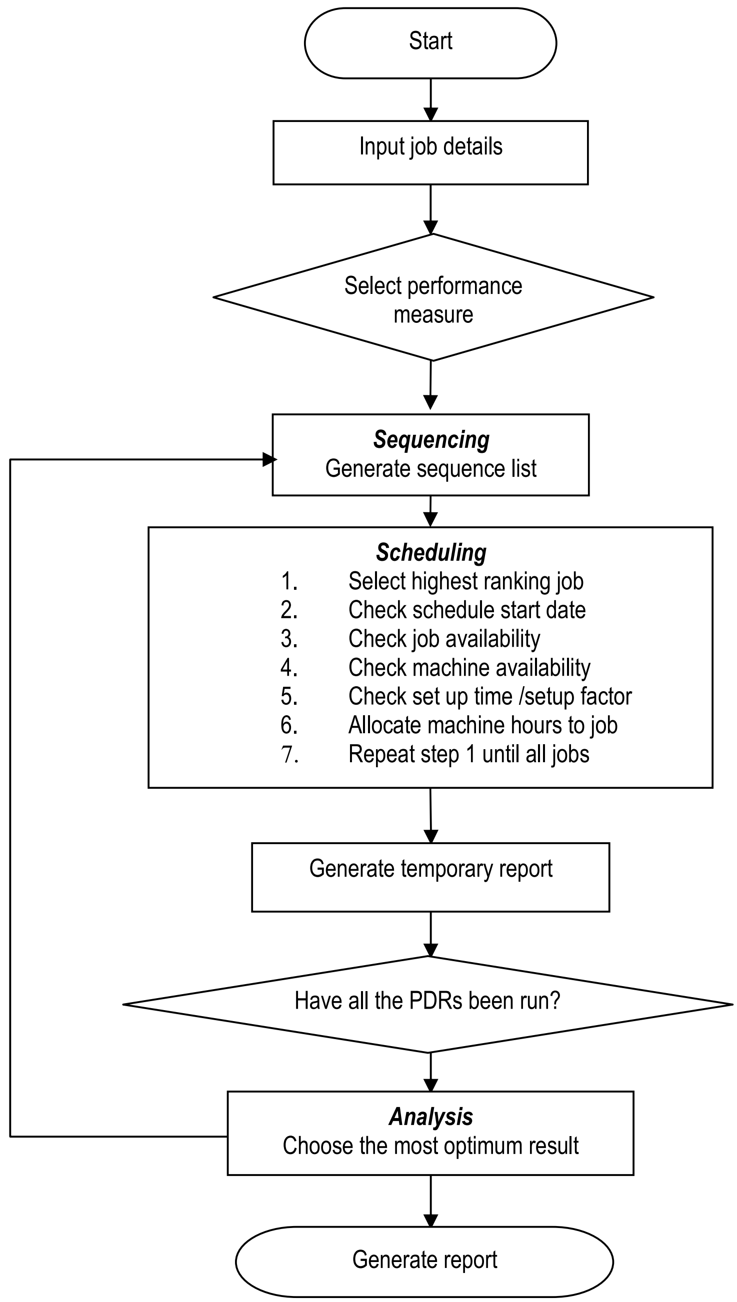

Figure 1: Block diagram of the scheduling program

The task of the program is to schedule n jobs concurrently by developing a schedule for each part traveling among the m machines with the objective specified by the performance measure. The five-most widely used performance measures namely minimise average job flow time, minimise make span, minimise percentage past due, minimise average days past due and maximise average early delivery are included in the program.

Scheduling involves with the assignment of start dates and completion date of a job at specific machine. The scheduling process is complicated by the fact that there may be hundreds jobs competing for time on limited number of machines. Once the job sequence is established, required machine hours are allocated to the job at the earliest available time of the job and the machine. The allocation is subjected to the availability of the job, availability of the machine, average set-up time and average set-up factor at each machine. Table 4 shows the algorithm used in scheduling jobs.

Table 4: Algorithm for scheduling jobs

(1).	Set k = 0 and j = 0	[Assign job-k at machine-j]
(2).	Set k = k + 1	[Select the highest ranking job based on PDR index]
(3).	Set j = j + 1	[Start at machine-1 first]
(4).	Set ESD(k,j) = max { SSD, AD(k)}	[Check job availability]
(5).	Set SD(k,j) = max { ESD(k,j), MAD(j) }	[Check machine availability]
(6).	Set t = SD(k,j)	[Allocate machine-j hour slot at time SD(r,j)]
(7).	If machine-j hour slot is empty at time t, then allocate 1 machine-j hour slot to job-r, otherwise assign t = t + 1	[search for the next earliest available machine-j hour slot]
(8).	Repeat step 7 until all T(k,j) complete	
(9).	Repeat step 3 to 8 until j = 5	[Select next process / machine]
(10).	Repeat step 2 to 9 until all jobs are selected, where k = max {k}	[Select the next ranking job based on PDR index]

RESULTS AND DISCUSSION

This section presents and discusses the results generated by the scheduling program after the users have selected their preferred performance measure. The program selects each PDR, generates the schedule and records its performance measures. It then analyses each performance measure and selects the best option.

Input Data

Table 5 shows a sample of a group of jobs used to test the system. Seven jobs were tested in the system, with job numbers from 801 to 807. The earliest job available is on 28/9/2000 and the latest is on 17/10/2000. The earliest due date is on 14/10/2000 and the latest on 31/10/2000. Two jobs need to go through all the processes. Job 806 requires only two processes to go through. The average machine set-up time of one hour and set-up factor of one is applied to each machine. Set-up factor ensures that the job will be carried out and the machine is in operation immediately after set up is done. Schedule start date is set to 2/10/2000 and "Minimise Makespan" is selected as the performance measure. Figure 2 shows the interface for job listing of the seven jobs entered to the system.

Table 5: Input job details

Job No	Descriptions	Avail. Date	Due date	Shear	Laser	Turret	Bend	Weld
801	Toll fare indicator	28/09/2000	31/10/2000	5	12	24	5	6
802	Air diffuser	02/10/2000	16/10/2000	5	12	0	60	20
803	KLIA box	05/10/2000	26/10/2000	4	15	25	0	12
804	Dolly muffler	07/10/2000	17/10/2000	2	8	0	12	0
805	Heater clamp	13/10/2000	14/10/2000	0	4	14	2	0
806	Chair stand	16/10/2000	24/10/2000	0	8	0	22	0
807	19" rack	17/10/2000	27/10/2000	4	5	5	14	12

		Set-up time	Set-up factor
Shear	:	1 hour	1
Laser	:	1 hour	1
Turret	:	1 hour	1
Bend	:	1 hour	1
Weld	:	1 hour	1

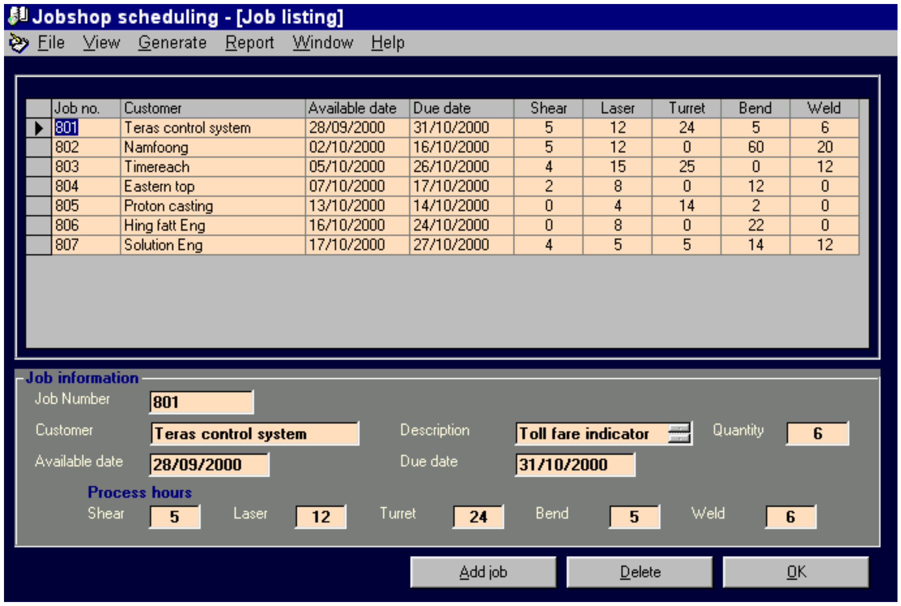

Figure 2: Job listing

Scheduler Action

Figure 3 shows the operation summary of the schedule. With “Minimise makespan” as the performance measure, the system recommends CR as the best PDR. CR offers only 23 days, which is the least number of days required to complete all the jobs in the system. The next best PDR is SRO with 24 days. This is followed by EDD with 27 days, fourth is FCFS with 31 days the last is SPT with 31 days.

From the operation summary, scheduler knows that all the jobs can be completed within 23 days. The first job, job 802, starts on 2/10/2000 at 9:00am and the last job, job 801, finishes on 24/10/2000 at 6:00pm. 28.57% of the jobs are delayed with average days past due per job equal to 0.86 days. Four jobs complete earlier than their due dates and one job completes on the due date, resulting with average days early of 3.43 days. CR is the best PDR to use if the objective is to minimise makespan.

Based on the information, scheduler first takes the following actions:

1. Inform management that the whole job will complete within 23 days. The average job flow time is 12.29 days. Two jobs delayed. Average days early per job is 3.43 days.
2. Notify customer of Job 802 that the job will only be completed on 17/10/2000 instead of 16/10/2000
3. Notify customer of Job 805 that the job will only be completed on 19/10/2000 instead of 14/10/2000

Assuming that the two customers accept the revised expected completion date, then the scheduler can proceed with this plan. The scheduler then issues instruction notifying the order sequence to start with jobs 802, 803, 801, 805, 807, 804 and 806.

If the customers still insist on meeting the initial due date then the scheduler will have to generate a new schedule. This can be done by using the first schedule as the initial solution and then arrange overtime works to meet the customers’ strict requirements.

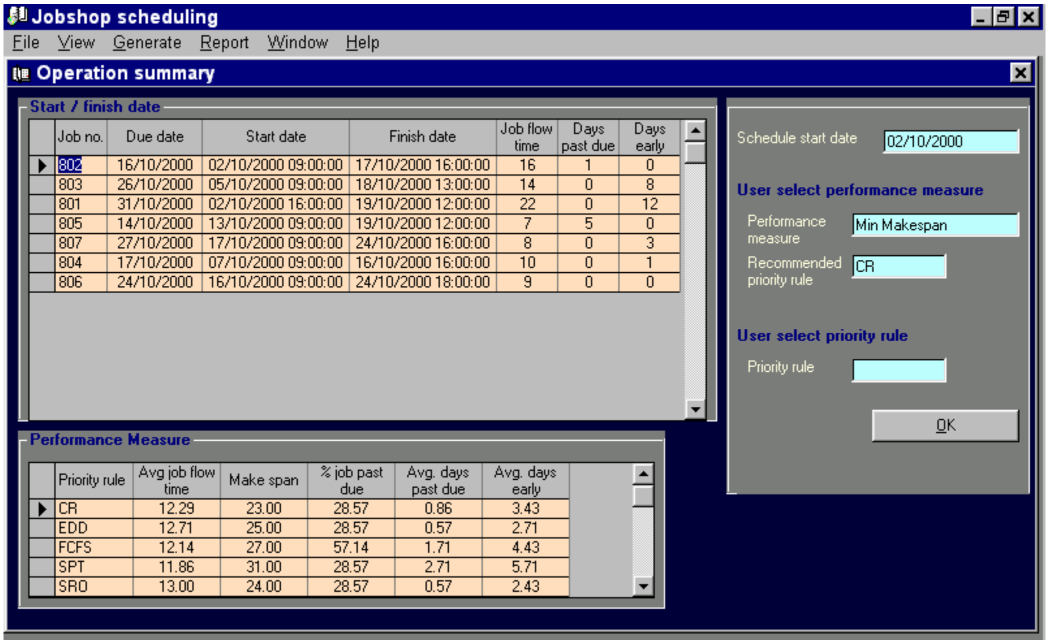

Figure 3: Operation summary with Makespan as performance measure

CONCLUSION

Due to the combinatorial nature of scheduling problems, it is often difficult to evaluate all possible schedules. The whole scheduling process to organise, maintain, update and reschedule the job is very tedious work and time consuming. As highlighted by Krajewski and Ritzman [9], Banga et. al. [7] and many other researchers, for n number of jobs and m number of machines, there are $(n!)^m$ possible schedules. To plot the schedule charts manually is not a very practical solution. Within a limited amount of computation time, to evaluate the whole time charts is not computationally efficient.

Priority dispatching rule with multiple choice of performance measure is used in developing the sequencing and scheduling program. PDR approach is preferred because it offers a substantially reduced computational requirement and ease of implementation as mentioned by Morton and Pentico [12].

The window-based scheduling program developed in this paper is a practical job sequencing and scheduling solution. It was designed using multiple performance measures and various priority dispatching rules to ease the process of sequencing and scheduling number of jobs to certain number of machines.

The program is very easy to use and user friendly. The user is only required to input job details, schedule start date and their preferred performance measure for the schedule. Based on these inputs, the system can then automatically generate the most optimum schedule solution. The system is able to automatically compute the PDR index, sequence the jobs, allocate machine hours to job and compute the performance measure. Within a short period, the user will have machines schedule, details operation for each job and the overall performance of the schedule generated. This is very helpful as to schedule the jobs manually will take a longer period of time, typically a few hours, starting from sequencing the jobs, generating Gantt chart to calculating the performance measure. The user can also use this first state of the schedule to further improve on the performance of the schedule.

ACKNOWLEDGEMENTS

The authors would like to extend its appreciation to Advanced Machining Centre, Technology Park Malaysia for all their help in carrying out this work.

REFERENCES

1. Ashby, J. R. and Uzsoy, R. (1995). Scheduling an Order Release in a Single-Stage Production System. Journal of Manufacturing Systems. 14(4), 290-306.
2. Zhang, Y. (1997). An Optimization-based Algorithm. http ://www.eng2.uconn.edu / msl / paper / jihua-india / node1.html
3. Adams, J., Balas, E. and Zawack, D. (1988). The Shifting Bottleneck Procedure for Job-Shop Scheduling. Management Science, 34(3), 391-401.
4. Carlier, J. and Pinson, E. (1989). An Algorithmn for Solving the Job Shop Problem. Management Science. 35(2) 164-176.
5. Mcmahon, G. and Florian, M. (1975). On Scheduling with ready Times and Due Dates to Minimize Lateness. Operation Research 23, 475-482.
6. Chu, W. W. and Ngai, P. H. (1993). Embedding Temporal Constraint Propagation in Machine Sequencing for Job Shop Scheduling. AI EDAM. 7(1), 37-52.
7. Banga, T. R., Sharma, S. C. and Agarwal, N. K. (1995). Industrial Engineering and Management Science. Delhi: Khanna Publisher.
8. Riggs, J. L. (1981). Production Systems: Planning, Analysis and Control (Third Edition). U.S.A: John Wiley & Sons.
9. Krajewski, L. J. and Ritzman, L. P. (1998). Operations Management: Strategy and Analysis (Fifth Edition). U.S.A: Addison-Wesley Publishing Company, Inc
10. Jain, A. S. and Meeran, S. (1998). The State-of-the-Art Review of Job Shop Scheduling Techniques. Department of Applied Physics, Electronic and Mechanical Engineering University of Dundee, Dundee, Scotland, UK, DD1 4HN. http ://citeseer.nj.nec.com/jain98stateart.html
11. French, S. (1982). Sequencing and scheduling – An Introduction to the Mathematics of Job Shop. New York: Ellis Horwood, John Willey & Sons.
12. Morton, T. E. and Pentico, D. W. (1993). Heuristic Scheduling Systems. New York: Wiley Series in Engineering and Technology management.
13. Chang, Y. L., Sueyoshi, T. and Sullivan, R. S. (1996). Ranking Dispatching Rules by Data Envelopment Analysis in a Job-Shop Environment. IIE Transaction, 28(8), 631-642.
14. Sun, X., and Noble, J. S. (1999). An Approach to Job Shop Scheduling With Sequence-Dependent Setups. Journal of Manufacturing Systems, 18(6), 416-430.
15. ElMaraghy, H., Patel, V., and Abdallah, I. B. (2000). Scheduling of Manufacturing Systems Under Dual-Resource Constraints Using Genetic Algorithms. Journal of Manufacturing Systems. 19(3), 186-201.
16. Veral E. A. (2001). Computers Simulation of Due-date Setting in Multi-machine Job Shops. Computers and Industrial Engineering, 41, 77-94
17. Glover, F. and Greenberg, H. J. (1989). New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence. European Journal of Operations Research, 39(2), 119-130.