Journal of Industrial Technology 8 (2), 1999, 61-67

DEVELOPMENT OF A SOFTWARE PACKAGE FOR NEAR OPTIMAL JOB SCHEDULING IN BATCH MANUFACTURING SYSTEM USING BORLAND C++ VERSION 5

Jaharah A. Ghani and Hew Yoon Ming

Department of Mechanical and Material Engineering
Faculty of Engineering
Universiti Kebangsaan Malaysia
43600 Bangi
Selangor Darul Ehsan, Malaysia

RINGKASAN: *Perisian komputer ini dibangunkan untuk membantu perancangan dan penjadualan aliran kerja dalam sistem pembuatan berkelompok. Dalam projek ini kajian dilakukan ke atas algoritma heuristik yang sedia ada dan digunakan untuk menyelesaikan masalah-masalah penjadualan dalam sistem tersebut. Algoritma heuristik yang sesuai dipilih dan diprogramkan. Jujukan kerja ini merangkumi sistem pusat mesin yang diatur secara siri dan selari. Algoritma yang digunakan adalah kaedah masa pemprosesan terpendek (SPT), algoritma Johnson, Campbell et al., dan Kaspi - Montreuil. Perisian ini telah dibangunkan dengan menggunakan Borland C++ versi 5 dan kaedah berorientasikan objek. ObjectWindow Libraries (OWL5) yang terkandung di dalam Borland C++ versi 5 dapat memendekkan masa dan usaha yang diperlukan dalam penulisan aturcara. Sistem operasian sebagai dasar perisian yang dipilih adalah Windows 95 yang merupakan sistem operasian 32 bit.*

ABSTRACT: Software was developed to assist planning and scheduling of the flow of jobs in a batch manufacturing system. In this project, studies were made on those existing heuristic algorithms that could be used to solve scheduling problems in order to obtain optimal or near optimal solutions. Appropriate heuristic algorithms were chosen and programmed. The job sequencings both in serial machine centre and parallel machine centre systems were covered. The algorithms used were shortest processing time method (SPT), Johnson algorithm, Campbell *et al.,* and Kaspi - Montreuil algorithms. The software was developed using Borland C++ version 5 and constructed using object oriented programming. *ObjectWindow Libraries* (OWL5) included in the Borland C++ version 5 helped to shorten the time and effort required to develop the program. The platform for developing the software was the 32-bit operating system, *Windows 95.* This software is typical window programme having common characteristics of a window application.

KEYWORDS: Batch manufacturing, scheduling, computer simulations, Borland C++.

INTRODUCTION

Manufacturing system in industry can be classified as a conversion system, which convert the input resources such as raw material to become the demanded products. This conversion system depends on the conversion process characteristics as well as the product produced (Adam and Ebert,1986). One of the typical methods that are widely used in industry is batch processing. One of the main problems facing this type of processing is how to schedule and sequence jobs properly in order to obtain an optimal output. In general, planning and operation scheduling entails the following: to fix the time and the amount of production, the use of operation capacity, and balancing the production with capacity at desired level to compete for the efficiency (Schonberger,1985). It appears that the use of genetic algorithms is an effective and popular method of addressing production scheduling. One such approach looks at a job-shop problem as an ordering problem (Guoyong,1997). Another approach studies the problem of simultaneous scheduling and optimizing processing times in batch operations. The best processing time for each job using a fixed job sequence is determined. Once the optimal processing time is determined, these times are then used to determine the sequence of processing (Karabati *et al.,*1997).

The objective of this study is to develop software using particular available heuristics to solve the scheduling problems in batch processing. When this is done, the efficiency of the shop is maximised. The shop efficiency is defined in terms of a set of quantified criteria. The quantified criteria used to judge the performance include production lead time, mean lateness, flow time and tardiness. Scheduling problems in batch processing are complicated and time consuming to solve manually. The development of the software as such would be beneficial for the manufacturers to simulate their real working layout, which would avoid losses due to the unplanned schedule. Besides, the capability of Borland C++ 5.0 is evaluated before deciding to develop the software.

Borland C++ 5.0 as an Object Oriented Programming Language

Borland C++ is one of the well-known object oriented programming languages available today. The basic idea of this language is to integrate both data and function that operate above the data. The integral unit is called an object. The characteristics of the object oriented programming language are: i) object ii) class, iii) inheritance, and iv) polymorphism (Lafore,1991). Borland C++ 5.0 is a C++ compiler produced by Borland Company to support Microsoft Windows 95 operating system. ObjectWindows version 5 or OWL5 is included in the Borland C++ 5.0 and acts as a library which helps shorten the time taken to develop a windows programme. Windows 95 was chosen as a programming platform due to certain advantages as follows (Lafore,1991):

(i) Multitasking: Its allows several programs to run simultaneously.
(ii) Windows unique characteristics such as GUI (graphical user interface) and the intensive use of mouse help develop a user-friendly programme.
(iii) The operation system in Windows 95 is widely used in the world, and the future of the system is very encouraging.

SOFTWARE DEVELOPMENT

Heuristic technique algorithm was used to develop the software in order to solve the scheduling problems. Certain rules should be followed in order to get an optimal, or near optimal solution. However, one weakness of this technique is that it was unable to check or measure the optimum of the solution. Optimum solution could only be obtained for the problems with small number of machines. Typical performance criteria that should be optimised were (Elsayed and Boucher,1994):

i) Mean flow time
ii) Machines idle time
iii) Mean lateness of jobs
iv) Mean earliness
v) Mean tardiness of jobs
vi) Mean queue time
vii) Average jobs in the system
viii) Percentage of lateness of jobs.

The following factors describe and classify particular scheduling problem (Elsayed and Boucher,1994):

i) Number of jobs that should be scheduled
ii) Number of machines in the workshop
iii) Type of manufacturing system (flow system or job system)
iv) Methods of how the jobs arrived to the shopfloor (statics or dynamics)
v) The considered criteria for scheduling.

There are several techniques used to analyse the sequencing arrangement problems, but none of them give the optimum solution, especially for bigger number of n jobs and m machines. Most of the researchers used simulation technique and algorithm heuristics as their research tool. Algorithm heuristics method could give an optimum or near optimum solutions. The disadvantage about this method is, although the solution is obtained, it cannot be checked as to how far it has deviated from the optimum.

In the process of software development, the problems involving the following cases were considered and the heuristics were programmed (Elsayed and Boucher,1994):

i) n jobs one machine.
There are n jobs to be processed by one machine, all jobs must pass through this machine and all jobs will experience two types of time: waiting and processing time. To schedule the sequence of the jobs, the mean flow time (MFT) should be minimised. MFT is computed as:

$$\text{MFT} = \sum_{i=1}^{n} \frac{Ci}{n} \tag{1}$$

Where MFT = mean flow time in the shopfloor
C_i = finish time for job i
n = number of jobs to be processed.

The rule used to optimise the sequence i.e. minimise the mean flow time, is the shortest processing time (SPT), whereby job with shortest processing time will be first scheduled, followed by the second shortest processing time etc. Another method is using common due date. The problem is as before, except that there is one common due date for all the jobs. This problem could be formulated as:

$$\text{minimised } Z = \sum_{i=1}^{n} \left| Ci - d \right| \tag{2}$$

$$\text{and } S[1] = 0$$

Where $\bar{C_i}$ = finish time for work i ($C_i = W_i + t_i$) (3)
W_i = waiting time for job i
S[1] = initial time for job i

Clearly d ≤ ms (ms is a makespan for all the jobs, i.e. equal to $\sum_{i=1}^{n} ti$)

ii) n jobs, two machines

This is the situation where n jobs must be processed by two machining centres; i.e. M1 followed by M2. The processing time for all jobs at M1 and M2 are known. It is required to find out the optimum sequence that could minimise the makespan for n jobs (i.e. the sequence to minimise the time taken to complete all jobs). Johnson had developed one algorithm that could be used to achieve the optimal sequence. In general, the following steps will be taken for Johnson's algorithm:

1. List out the processing time for all the jobs that should be processed in machine M1 and machine M2.
2. Check out all processing time for all jobs. Find out the minimum processing time.
3. If the minimum processing time was at machine 1, put it first (as early as possible) in that sequence. If it was at M2, place the job at the end (as late as possible) in the sequence.

4. Eliminate the assigned jobs (already placed in the sequence as a result of step 3 and repeat step 2 and step 3 until all the jobs are sequenced).
5. A tie between two processing times is broken arbitrarily because it cannot affect the minimum elapsed time to complete all the jobs.

iii) n jobs, three machines

This is continuity from n jobs, two machines. Optimum solution could only be obtained with problems where all jobs must be processed in the same sequence at each machine. Consider three machines with M1, M2, and M3. All jobs must be processed in the same sequence at each machine. We define tij as a processing time for job i at machine j. Johnson algorithm for n jobs, two machines could be used for n jobs, three machines, and the optimum solution could be obtained if it fulfills the following:

min tij $\geq$ max ti2 (i = 1, 2, ...,n)

or min ti3 $\geq$ max ti2 (i = 1, 2,.......,n)

The problems were formulated by developing two dummy machines, i.e. M1', M2' to replace the 3 existing machines. Processing time for job i at M1' is ti1 + ti2, and the processing time at M2' is ti2 + ti3. Therefore, Johnson algorithm was used for two machines M1' and M2' to find out the optimum sequence.

iv) n jobs, m machines

This is a typical sequence scheduling problem, where n jobs must be processed by a set of machines m. All jobs must be processed at the initiation of scheduling period, and no new jobs should arrive during that period. Also, jobs are not allowed to pass each other (that is, jobs maintain the same position in the sequence). The problem is to schedule n jobs on m machines such that the jobs are completed in a minimum span of time. Unfortunately, there is no general solution for any problem where m > 3. However, some heuristic techniques may obtain good sequences or may obtain optimum sequences. This type of problem is solved using Campbell *et al* algorithm. They developed an algorithm that generates a series of sums for each job similar to the two sets of sums generated in the n jobs, three machines problem (M1 and M2). With m machines, M-1 two - column sets of job times can be developed and can then be solved using Johnson's algorithm for n jobs, two machines.

v) Sequencing jobs on parallel machines

In typical job - shop - based plant layouts, machines are grouped according to a common function and are placed in the same department. When multiple jobs (parts) arrive at a department to be processed, a job sequence is made and the jobs are assigned to the machines accordingly. Two criteria are widely used for sequencing jobs on machines: job makespan and mean tardiness.

DISCUSSION

The software is developed especially for scheduling jobs in batch production manufacturing system. In general, the input data comes from the user and using a particular heuristic the data is processed. Then, the solution will be displayed on the computer screen. The solutions obtained are optimum or near optimum for scheduling problems in batch production system, which would eliminate the tardiness and time consuming if done manually. In addition, the result obtained from computer computational is accurate and could help the manufacturer to optimise the available facility i.e. machines and man power. Furthermore the profit is gained from delivery of goods on time, as scheduled, to the customer.

For the best illustration, the example of how the n jobs, three machines work is as below:

Table 1: 6 jobs with different processing time on 3 machines

	Processing time on :		
Job	**M1**	**M2**	**M3**
1	5	3	9
2	7	2	5
3	4	3	7
4	8	4	3
5	6	2	2
6	7	0	8

The result obtained is shown on the screen as follows:

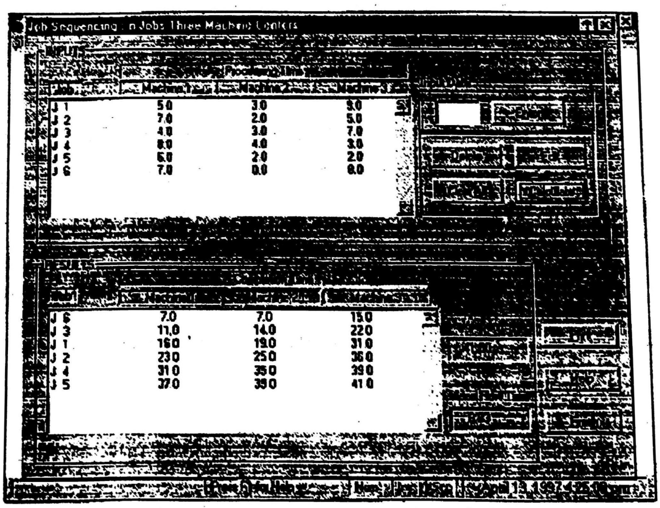

Figure 1. *Dialog box for n jobs, three machines.*

The result shows that the best sequence of the jobs is J6 → J3 → J1 → J2 → J4 → J5, which will give the minimum time to finish all the jobs assigned.

CONCLUSION

In this project, one user-friendly software was successfully developed. It could be used to help the manufacturers plan and control their scheduling production in a batch manufacturing system. The software was tested and found to be satisfactory and free from error.

ACKNOWLEDGEMENT

The work presented in this paper was carried out at the Department of Mechanical and Materials Engineering, UKM. Support of the university through its research grant FK 53/93 is gratefully acknowledged.

REFERENCES

Adam, E.E. Jr. and Ebert, R.J. (1986). Production And Operation Management - Concepts, Models and Behavior.Englewood Cliffs, Prentice- Hall International.

Arnush, C. (1996). Teach Yourself Borland C++ 5.0 In 21 Days. Indianapolis, Sam Publishing.

Elsayed, E.A. and Boucher, T.O. (1994). Analysis and Control of Production Systems. Englewood Cliffs, Prentice Hall International Inc.

Guoyong, S. (1997). Genetic algorithm applied to a classic job-shop scheduling problem. *Intl. J. System Science,* **28(1)**: pp 25-32.

Karabati, Selcuk, Kouvelis, Panagiotis. (1997). Flow-line scheduling problem with controllable processing times, *IIE Transactions (Institute of Industrial Engineers),* **29(1)**, pp 1-14.

Lafore, R. (1991). Object Oriented Programming in Turbo C++. Emeryville, White Group Press.

Schonberger, R.J. (1985). Operations Management, Productivity and Quality. Plano, Business Publication Inc.