Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II
IMECS 2008, 19-21 March, 2008, Hong Kong

# Makespan Computation for Cyber Manufacturing Centre Using Bottleneck Analysis: A Re-entrant Flow Shop Problem

Salleh Ahmad Bareduan and Sulaiman Hj. Hasan

***Abstract*— This paper presents the development of an alternative method for makespan computation algorithms of a re-entrant flow shop scheduling problem using bottleneck analysis. The computation is specifically intended for the cyber manufacturing centre (CMC) which is an Internet based collaborative design and manufacturing services at Universiti Tun Hussein Onn Malaysia. The CMC processes scheduling resembles a four machine permutation re-entrant flow shop with the process routing of M1,M2,M3,M4,M3,M4. It was shown that under the sequence dependent bottleneck characteristics, the makespan can be accurately determined by the algorithm developed using bottleneck analysis. In cases where the bottleneck limitation is violated, the makespan can still be accurately determined by the introduction of bottleneck correction factor.**

***Index Terms*— bottleneck, cyber manufacturing, scheduling, re-entrant flow shop,**

## I. Introduction

Flow shop manufacturing is a very common production system found in many manufacturing facilities, assembly lines and industrial processes. It is known that finding an optimal solution for a flow shop scheduling problem is a difficult task [1] and even a basic problem of $F3 \parallel C_{max}$ is already strongly NP-hard [2]. Therefore, many researchers have concentrated their efforts on finding near optimal solution within acceptable computation time using heuristics.

One of the important subclass of flow shop which is quite prominent in industries is re-entrant flow shop. The special feature of a re-entrant flow shop compared to ordinary flow shop is that the job routing may return one or more times to any facility. Among the researchers on re-entrant flow shop, [3] has developed a cyclic scheduling method that takes advantage of the flow character of the re-entrant process. This work illustrated a re-entrant flow shop model of a semiconductor wafer manufacturing process and developed a heuristic algorithm to minimize average throughput time using cyclic scheduling method at specified production rate. The decomposition technique in solving maximum lateness problem for re-entrant flow shop with sequence dependent setup times was suggested by Dermirkol and Uzsoy [4]. Mixed integer heuristic algorithms was later on elaborated by Pan and Chen [5] in minimizing makespan of a permutation flow shop scheduling problem. Significant works on re-entrant hybrid flow shop can be found as in [6],[7],[8] while hybrid techniques which combine lower bound-based algorithm and idle time-based algorithm was reported in [9].

In scheduling literature, heuristic that utilize the bottleneck approach is known to be among the most successful methods in solving shop scheduling problem. This includes shifting bottleneck heuristic [10],[11] and bottleneck minimal idleness heuristic [12],[13]. However, not much progress is reported on bottleneck approach in solving re-entrant flow shop problem. Among the few researches are Dermirkol and Uzsoy [4] who developed a specific version of shifting bottleneck heuristic to solve the re-entrant flow shop sequence problem.

In this paper we explore and investigated an Internet based collaborative design and manufacturing process scheduling which resembles a four machine permutation re-entrant flow shop. The study is searching for the potential of developing an effective makespan minimization heuristic by firstly developing makespan computation algorithm using bottleneck analysis. This computation is specifically intended for the cyber manufacturing centre at Universiti Tun Hussein Onn Malaysia (UTHM).

## II. Cyber Manufacturing Centre

UTHM has recently developed a web-based system that allows the university to share the sophisticated and advanced machinery and software available at the university with the SMEs using Internet technology [14]. The heart of the system is the cyber manufacturing centre (CMC) which consists of an advanced computer numerical control (CNC) machining centre fully equipped with cyber manufacturing system software that includes computer aided design and computer aided manufacturing (CAD/CAM) system, scheduling system, tool management system and machine monitoring system.

The Petri net (PN) model that describes a typical design and manufacturing activities at the CMC is shown in Figure 2. The places denoted by P22, P23, P24 and P25 in Figure 2 are the resources utilized at the CMC. These resources are the CAD system, CAM system, CNC postprocessor and CNC machine centre respectively. At the CMC, all jobs must go through all processes following the sequence represented in the PN model. This flow pattern is very much similar with flow shop manufacturing [2],[15]. However, it can be noticed from the PN model that there are a few processes that share

Manuscript received August 3, 2007.

S. A. Bareduan is with Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Johor, Malaysia. ( phone: 607-453-7772; fax: 607-453-6080; e-mail: saleh@uthm.edu.my).

S. Hj. Hasan is with Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Johor, Malaysia. (e-mail: sulaiman@uthm.edu.my).

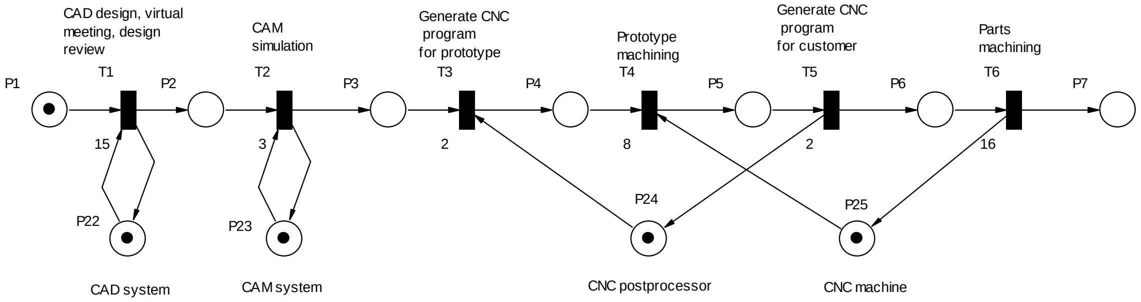


Figure 1 : Petri Net Model of CMC activities

common resources. The process of generating CNC program for prototyping (T3) and the process of generating CNC program for customer (T5) are executed on the same CNC postprocessor (P24). Similarly, the processes of prototype machining (T4) and parts machining (T6) are executed on the same CNC machine centre. Thus, this process flow is considered as a re-entrant flow shop as described in [3]. It can also be noticed that both shared resources (P24 and P25) must completely finish the processing of a particular job at T5 and T6 before starting to process any new job at T3 and T4. In other words, this problem can be also identified as four machine permutation re-entrant flow shop with the processing route of M1,M2,M3,M4,M3,M4 as similarly described in [16].

## III. CMC Makespan Computation Under Bottleneck Limitations

Let say, the CMC is currently having four jobs that need to be processed. Typical processing time ranges for all processes are shown in Table 1. From Table 1, it is obvious that most probably T1 is the bottleneck for the overall process because it is having the longest processing time range. By using the time ranges in Table 1, sets of random data was generated for four jobs that need to be processed. These data is shown in Table 2. Assuming that the data in Table 2 is arranged in the order of First-come-first-served (FCFS), then a Gantt chart representing a FCFS schedule is built as illustrated in Figure 2. The Gantt chart is built by strictly referring to the PN model in Figure 1 together with strict permutation rule.

Table 1 : Processing Time Range (hr)

| | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| Minimum time | 70 | 2 | 2 | 8 | 2 | 8 |
| Maximum time | 100 | 8 | 8 | 40 | 8 | 40 |

Table 2 : Processing Time Data (hr)

| | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| Job A | 73 | 8 | 3 | 8 | 5 | 30 |
| Job B | 90 | 2 | 5 | 32 | 5 | 32 |
| Job C | 98 | 2 | 3 | 8 | 8 | 17 |
| Job D | 75 | 6 | 3 | 36 | 4 | 35 |

By referring to Table 2, Figure 1 and Figure 2, the scheduling algorithm for the CMC can be written as the followings and is identified as Algorithm 1:

| P | Job | 0 | 24 | 48 | 72 | 96 | 120 | 144 | 168 | 192 | 216 | 240 | 264 | 288 | 312 | 336 | 360 | 384 | 408 | 432 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | JA | 24 | 24 | 24 | 1 | | | | | | | | | | | | | | | |
| | JB | | | | 23 | 24 | 24 | 19 | | | | | | | | | | | | |
| | JC | | | | | | | 5 | 24 | 24 | 24 | 21 | | | | | | | | |
| | JD | | | | | | | | | | | 3 | 24 | 24 | 24 | | | | | |
| T2 | JA | | | | 8 | | | | | | | | | | | | | | | |
| | JB | | | | | | | 2 | | | | | | | | | | | | |
| | JC | | | | | | | | | | | 2 | | | | | | | | |
| | JD | | | | | | | | | | | | | | | 6 | | | | |
| T3 | JA | | | | 3 | | | | | | | | | | | | | | | |
| | JB | | | | | | | 3 | 2 | | | | | | | | | | | |
| | JC | | | | | | | | | | | 1 | 2 | | | | | | | |
| | JD | | | | | | | | | | | | | | | 3 | | | | |
| T4 | JA | | | | 8 | | | | | | | | | | | | | | | |
| | JB | | | | | | | | 22 | 10 | | | | | | | | | | |
| | JC | | | | | | | | | | | | 8 | | | | | | | |
| | JD | | | | | | | | | | | | | | | 15 | 21 | | | |
| T5 | JA | | | | 4 | 1 | | | | | | | | | | | | | | |
| | JB | | | | | | | | | 5 | | | | | | | | | | |
| | JC | | | | | | | | | | | | 8 | | | | | | | |
| | JD | | | | | | | | | | | | | | | | 3 | 1 | | |
| T6 | JA | | | | | 23 | 7 | | | | | | | | | | | | | |
| | JB | | | | | | | | | 9 | 23 | | | | | | | | | |
| | JC | | | | | | | | | | | | 6 | 11 | | | | | | |
| | JD | | | | | | | | | | | | | | | | | 23 | 12 | |

Figure 2: Gantt Chart for ABCD Job Sequence

**Algorithm 1**
Let $i$ = Transition number, process number or work centre number ($i$=1,2,3,….)
$j$ = Job number ($j$=1,2,3,…)

Start ($i,j$) = start time of the $j^{th}$ job at $i^{th}$ work centre.
Stop ($i,j$) = stop time of the $j^{th}$ job at $i^{th}$ work centre.
P($i,j$) = processing time of the $j^{th}$ job at $i^{th}$ work centre.

For $i$=1,2,5,6 and $j$=1,2,3,…$n$
Start ($i,j$) = Max [Stop ($i,j$-1), Stop ($i$-1,$j$)] except Start (1,1) = initial starting time
Stop ($i,j$) = Start ($i,j$) + P ($i,j$)

For $i$ =3,4 and $j$=1,2,3,…$n$
Start ($i,j$) = Max [Stop ($i,j$-1), Stop ($i$-1,$j$), Stop ($i$+2,$j$-1)]
Stop ($i,j$) = Start ($i,j$) + P($i,j$)

Thorough study on the schedule Gantt chart as in Figure 2, a general makespan computation algorithm for the example case can be described as below:

Let $i$ = process sequence of the job at CMC ($i$=1,2,3,4,5,6)
$j$ = job number according to the scheduling sequence ($j$=1,2,3…$n$)
$P(i,j)$ = processing time of the $j^{th}$ job at $i^{th}$ process sequence

The makespan calculation is:

$$\sum_{j=1}^{n} P(1,j) \quad + \quad \sum_{i=2}^{6} P(i,n) \qquad \textbf{(Equation 1)}$$

Equation 1 is very much similar with completion time algorithm described in [13] for the problem $Fm|ddm|\gamma$ and $Fm|no\text{-}wait,ddm|\gamma$. They illustrated the scheduling sequence of decreasing dominant machine ($ddm$) in which Min{$j$=1,2…$n$}[$P(k,j)$] ≥ Max{$j$=1,2…$n$}[$P(r,j)$]. While reports in [13] concentrated on some special cases of general, no-wait and no-idle permutation flow shop scheduling problems, this paper focuses on the problem of a re-entrant flow shop that exhibits dominant or bottleneck machine characteristics at the first task.

After careful studies on the Gantt charts of other possible jobs arrangements, it is observed that Equation 1 is valid for makespan computation if some localized sequence dependent conditions are met. These localized sequence dependent conditions for the 4-job example case are:

**Condition 1**
$P(1,2) + P(1,3) + P(1,4) \geq VP(2,1) + VP(2,2) + VP(2,3)$
Where, $VP$ = Virtual Processing Time.

Virtual processing time is an imaginary processing time that assumes the starting time of any process at a work centre must begin immediately after the completion of the previous imaginary process. For example, consider a job X starting on task 2 and at the same time a job Y starts at task 1. If the completion time of job X on task 2 is earlier than the completion time of job Y at task 1, under the imaginary concept, the $VP$ of job X at task 2 is extended from its actual processing time to match the completion time of job Y at task 1. This means the $VP$ of job X at task 2 is equivalent to the processing time of job Y at task 1 since task 2 of job Y can only be started immediately after its completion at task 1 regardless of the earlier completion time of job X at task 2. The concept of $VP(i,j)$ is introduced in this condition to simplify the algorithm so that very limited numbers or not even a single element of P($i,j$) is shown on the right side of the conditions statement.

Condition 1 is meant to make sure that for the last job sequence, task 2 can immediately be started as soon as task 1 completed its process. For example, if Condition 1 is violated, $P(2,n\text{-}1)$ completion time is later than the completion time of $P(1,n)$, this means that $P(2,n)$ cannot start immediately after the completion of $P(1,n)$. It can only begin after the completion of $P(2,n\text{-}1)$ which is also indicated by the completion time of $VP(2,n\text{-}1)$. This introduces a delay between $P(1,n)$ and $P(2,n)$ thus affecting the accuracy of Equation 1.

The virtual processing time for task 2 are assigned as the followings:
For j=1, $VP(2,1) = \text{Max}\,[P(2,1), P(1,2)]$
For j=2,3…$n$-1,
$VP(2,j)$ =

$$Max\left[\left[\sum_{k=1}^{j-1} VP(2,k)\right] + P(2,j), \left[\sum_{k=2}^{j+1} P(1,k)\right]\right] - \sum_{k=1}^{j-1} VP(2,k)$$

**Condition 2**

$P(1,2) + P(1,3) + P(1,4) + P(2,4) \geq$
$P(2,1) + VP(3,1) + VP(3,2) + VP(3,3)$

Condition 2 functions to ensure that for the last job sequence, task 3 can immediately be started as soon as task 2 completed its process. For example, if Condition 2 is violated, this means that the right side value of the above condition is larger than its left side value. Since $P3$ and $P5$ are sharing the same postprocessor P24 (refer Figure 1), the violation of Condition 2 will result to a later completion time of $P(5,n\text{-}1)$ compares to the completion time of $P(2,n)$. Consequently, $P(3,n)$ cannot start immediately after the completion of $P(2,n)$. It can only begin after the completion of $P(5,n\text{-}1)$ which is indicated by the completion time of $VP(3,n\text{-}1)$. This introduces a delay between $P(2,n)$ and $P(3,n)$ thus affecting the accuracy of Equation 1.

The virtual processing time for task 3 are assigned as the followings:

For j=1, $VP(3,1) = \text{Max}\,[\{VP(2,1) + P(2,2)\},\{P(2,1) + P(3,1) + P(4,1) + P(5,1)\}] - P(2,1)$

For j=2,3…$n$-1, $VP(3,j)$ =

$$Max\left[\begin{array}{l}\left[\sum_{k=1}^{j} VP(2,k)\right] + P(2,j+1), \left[P(2,1) + \sum_{k=1}^{j-1} VP(3,k) + \sum_{i=3}^{5} P(i,j)\right], \\ P(2,1) + P(3,1) + \sum_{k=1}^{j-1} VP(4,k) + \sum_{i=4}^{5} P(i,j)\end{array}\right]$$

$$- \left[P(2,1) + \sum_{k=1}^{j-1} VP(3,k)\right]$$

**Condition 3**

$$P(1,2) + P(1,3) + P(1,4) + P(2,4) + P(3,4) \geq P(2,1) + P(3,1) + VP(4,1) + VP(4,2) + VP(4,3)$$

Condition 3 functions to guarantee that for the last job sequence, task 4 can immediately be started as soon as task 3 completed its process. Since $P4$ and $P6$ are sharing the same CNC machine P25 (refer Figure 1), the violation of Condition 3 will result to a later completion time of $P(6,n\text{-}1)$ compares to the completion time of $P(3,n)$. Consequently, $P(4,n)$ cannot start immediately after the completion of $P(3,n)$. It can only begin after the completion of $P(6,n\text{-}1)$ which is indicated by the completion time of $VP(4,n\text{-}1)$. This introduces a delay between $P(3,n)$ and $P(4,n)$ thus affecting the accuracy of Equation 1.

The virtual processing time for task 4 are assigned as the followings:

For j=1, $VP(4,1) = \text{Max}\ [\{VP(3,1) + P(3,2)\},\{P(3,1) + P(4,1) + P(5,1) + P(6,1)\}] - P(3,1)$

For j = 2,3…$n$-1, $VP(4,j)$ =

$$Max\left[\left[\sum_{k=1}^{j} VP(3,k)\right] + P(3,j+1), \left[P(3,1) + \sum_{k=1}^{j-1} VP(4,k) + \sum_{i=4}^{6} P(i,j)\right]\right] - \left[P(3,1) + \sum_{k=1}^{j-1} VP(4,k)\right]$$

## IV. GENERALIZED CMC MAKESPAN COMPUTATION

By meeting all Conditions 1, 2 and 3, a job sequence arrangement is said to have fulfilled the P1 (process 1) bottleneck characteristics of the CMC and this enables Equation 1 to be used for the makespan computation. If any of the Conditions 1, 2 and 3 is violated, Equation 1 is no longer valid for the makespan computation. This equation has to be modified and improved by introducing a dedicated correction factor in order to absorb the violated conditions if it is still to be used for makespan computation beyond the above stipulated conditions.

Detail observations of Conditions 1, 2 and 3 reveals that the inaccuracy of Equation 1 due to the violation of Condition 1 is inclusive in its computation of VP(2,$j$) which will be used in Condition 2. Similarly, the error of Equation 1 resulted from the violation of Condition 2 is also inclusive in its computation of VP(3,$j$) which will later be used in Condition 3. As such, by evaluating and monitoring specifically on Condition 3, all the errors of Equation 1 resulted from the violations of either Conditions 1, 2 and 3 or their combinations can be computed.

Table 3 is specifically developed using majority data from Table 2 in order to show the process of determining the values of $VP(2,j)$, $VP(3,j)$ and $VP(4,j)$. These values will be used to detect the occurrences of bottleneck at processes other than $P(1,j)$. In other words, this table will be used to suggest the correction factor need to be added to Equation 1 if the previously described Condition 3 is violated. This correction factor can be computed as the followings:

From Condition 3:

$$P(1,2) + P(1,3) + P(1,4) + P(2,4) + P(3,4) \geq P(2,1) + P(3,1) + VP(4,1) + VP(4,2) + VP(4,3)$$

If Condition 3 is violated, it means:

$$P(1,2) + P(1,3) + P(1,4) + P(2,4) + P(3,4) < P(2,1) + P(3,1) + VP(4,1) + VP(4,2) + VP(4,3)$$

Therefore, the correction factor can be computed as:

$$\text{P1BCF} = \{P(2,1) + P(3,1) + VP(4,1) + VP(4,2) + VP(4,3)\} - \{P(1,2) + P(1,3) + P(1,4) + P(2,4) + P(3,4)\}$$

If $\{P(2,1) + P(3,1) + VP(4,1) + VP(4,2) + VP(4,3)\} - \{P(1,2) + P(1,3) + P(1,4) + P(2,4) + P(3,4)\} < 0$ then, P1BCF = 0

The general formulation of the correction factor can be written as the following:

P1BCF =

$$\text{Max}\left[0, \sum_{i=2}^{3} P(i,1) + \sum_{j=1}^{n-1} VP(4,j) - \sum_{j=2}^{n-1} P(1,j) - \sum_{i=1}^{3} P(i,n)\right]$$

where,

P1BCF = Process 1 Bottleneck Correction Factor

Therefore the generalized makespan computation algorithm for the CMC is:

$$\text{Makespan} = \sum_{j=1}^{n} P(1,j) + \sum_{i=2}^{6} P(i,n) + \text{P1BCF} \qquad (\textbf{Equation 2})$$

Table 3 : Table For Makespan Computation

| Job | $j$ | $P(1,j)$ | $P(2,j)$ | $P(3,j)$ | $P(4,j)$ | $P(5,j)$ | $P(6,j)$ |
|---|---|---|---|---|---|---|---|
| Job A | 1 | 73 | 8 | 3 | 8 | 5 | 30 |
| Job B | 2 | 90 | 2 | 5 | 32 | 5 | 32 |
| Job C | 3 | 98 | 2 | 3 | 35 | 8 | 39 |
| Job D | 4 | 75 | 6 | 3 | 36 | 4 | 35 |

| | A | B | C | D | E | F | G | H | K |
|---|---|---|---|---|---|---|---|---|---|
| j | Sum $P(1,k)$ $k$=2,$j$+1 | $VP(2,j)$ | Sum $VP(2,k)$ $k$=1,$j$-1 For $j$=2,3…$n$ | Sum $VP(2,k)$ $k$=1,$j$ | $VP(3,j)$ | Sum $VP(3,k)$ $k$=1,$j$-1 For $j$=2,3…$n$ | Sum $VP(3,k)$ $k$=1,$j$ | $VP(4,j)$ | Sum $VP(4,k)$ $k$=1,$j$-1 For $j$=2,3…$n$ |
| | | | | | | | | | |
| 1 | 90 | 90 | | 90 | 84 | | 84 | 86 | |
| 2 | 188 | 98 | 90 | 188 | 98 | 84 | 182 | 96 | 86 |
| 3 | 263 | 75 | 188 | 263 | 79 | 182 | 261 | 82 | 182 |

To verify the accuracy and reliability of Equation 2 in performing the makespan computations, a total of 10,000 tests were conducted using random data of between 1 to 80 hours for each of $P(1,j)$, $P(2,j)$, $P(3,j)$, $P(4,j)$, $P(5,j)$ and $P(6,j)$ with four job sequence for each test. Each set of random data obtained was also tested with a total of 24 different sequences that resembles the sequence arrangement of ABCD, ABDC, ACBD etc. This means that with 10000 sets of random data, a total of 240,000 job sequence arrangements were testedThe makespan results from using Equation 2 were compared with the makespan value obtained from Algorithm 1. The results from the comparisons showed that all makespan value from both Equation 2 and Algorithm 1 are the same. This indicates that Equation 2 produces accurate makespan computation for 4-job CMC scheduling problem. Equation 2 was also tested for estimating the makespan for 6-job, 10-job and 20-job CMC scheduling. Each test was conducted with 10,000 sets of random data between 1 to 80 hours for each of $P(1,j)$, $P(2,j)$, $P(3,j)$, $P(4,j)$, $P(5,j)$ and $P(6,j)$. Each set of random data obtained was also tested with a total of 720 different sequences that resembles the sequence arrangement of ABCDEF, ABCDFE, ABCEDF etc. All the results indicate that Equation 2 produces accurate makespan result exactly the same with Algorithm 1. This shows the reliability of Equation 2 in predicting the makespan of the CMC scheduling arrangements.

## V. Conclusion

In this paper, we explore and investigated the CMC processes scheduling which resembles a four machine permutation re-entrant flow shop with the process routing of M1,M2,M3,M4,M3,M4. It was shown that under the P1 bottleneck characteristics, the makespan of the job sequence can be accurately determined by the makespan algorithm developed using bottleneck analysis. In cases where the sequence dependent P1 bottleneck limitation is violated, the makespan can still be accurately determined by the introduction of bottleneck correction factor. With the successful makespan computation using bottleneck analysis, the next phase of this research is to further utilize the bottleneck approach in developing heuristic for optimizing the CMC scheduling sequences.

## References

[1] Z. Lian, X. Gu and B. Jiao, "A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan," *Chaos, Solitons and Fractals* 2006, doi:10.1016/j.chaos.2006.05.082.

[2] M. Pinedo, *Scheduling: Theory, algorithms, and systems*. 2nd ed. Upper Saddle River, N.J.: Prentice-Hall; 2002

[3] S. C. Graves, H. C. Meal, D. Stefek and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *Journal of Operations Management*, 1983, 3(4), pp. 197-207

[4] E. Demirkol and R. Uzsoy, "Decomposition methods for reentrant flow shops with sequence dependent setup times," *Journal of Scheduling*, 2000, 3, pp. 115-177

[5] J. C. Pan and J. S. Chen, "Minimizing makespan in re-entrant permutation flow-shops," *Journal of Operation Research Society*, 2003, 54(6), pp. 642-653

[6] K. Yura, "Cyclic scheduling for re-entrant manufacturing systems," *International Journal of Production Economics*, 1999, 60, pp. 523-528

[7] W. L. Pearn, S. H. Chung, A. Y. Chen and M. H. Yang, "A case study on the multistage IC final testing scheduling problem with reentry," *International Journal of Production Economics*, 2004, 88(3), pp. 257-267

[8] S. W. Choi, Y. D. Kim and G. C. Lee, "Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop," *International Journal of Production Research*, 2005, 43(11), pp. 2149-2167

[9] S. W. Choi and Y. D. Kim, "Minimizing makespan on an *m*-machine re-entrant flowshop," *Computers & Operations Research*, 2006, doi:10.1016/j.cor.2006.09.028

[10] J. Adams, E. Balas and D. Zawack, "The shifting bottleneck procedure for job shop scheduling," *Management Science*, 1988, 34, pp. 391-401

[11] S. Mukherjee and A. K. Chatterjee, "Applying machine based decomposition in 2-machine flow shops," *European Journal of Operational Research*, 2006, 169, pp. 723-741

[12] A. A. Kalir and S. C. Sarin, "A near optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots," *Omega*, 2001, 29, pp. 577-584

[13] J. B. Wang, F. Shan, B. Jiang and L. Y. Wang, "Permutation flow shop scheduling with dominant machines to minimize discounted total weighted completion time," *Applied Mathematics and Computation*, 2006, 182(1), pp. 947-954

[14] S. A. Bareduan, S. H. Hasan, N. H. Rafai and M. F. Shaari, "Cyber manufacturing system for small and medium enterprises: a conceptual framework," *Transactions of North American Manufacturing Research Institution for Society of Manufacturing Engineers*, 2006, 34, pp. 365-372

[15] G. C. Onwubolu, "A flow-shop manufacturing scheduling system with interactive computer graphics," *International Journal of Operations & Production Management*, 1996, 16(9), pp. 74-84

[16] D. L. Yang, W. H. Kuo and M. S. Chern, "Multi-family scheduling in a two-machine re-entrant flow shop with setups," *European Journal of Operational Research*, 2006, doi:10.1016/j.ejor.2006.06.065