

Makespan Algorithms and Heuristic for Internet-Based Collaborative Manufacturing Process Using Bottleneck Approach

Salleh Ahmad BAREDUAN, Sulaiman HASAN

Faculty of Mechanical and Manufacturing Engineering, University Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia.
Email: saleh@uthm.edu.my

Received August 20th, 2009; revised September 15th, 2009; accepted November 12th, 2009.

ABSTRACT

This paper presents makespan algorithms and scheduling heuristic for an Internet-based collaborative design and manufacturing process using bottleneck approach. The collaborative manufacturing process resembles a permutation re-entrant flow shop environment with four machines executing the process routing of M1,M2,M3,M4,M3,M4 in which the combination of the last three processes of M4,M3,M4 has high tendency of exhibiting dominant machine characteristic. It was shown that using bottleneck-based analysis, effective makespan algorithms and constructive heuristic can be developed to solve for near-optimal scheduling sequence. At strong machine dominance level and medium to large job numbers, this heuristic shows better makespan performance compared to the NEH.

Keywords: *Heuristic, Re-Entrant Flow Shop, Bottleneck, Scheduling, Dominant Machine*

1. Introduction

Flow shop manufacturing is a very common production system in many manufacturing facilities, assembly lines and industrial processes. In this environment, the operations of all jobs must follow the same order following the same route along the machines assumed to be set up in series [1]. It is known that finding an optimal solution for a flow shop scheduling problem is a difficult task [2] and even a basic problem involving three machines is already NP-hard [1]. Therefore, many researchers have concentrated their efforts on finding near optimal solutions within acceptable computation time using heuristics. Most heuristics are developed by the researchers after gaining familiarity and in-depth understanding of the system's characteristic or behaviour.

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. A group of researchers developed a cyclic scheduling method that takes advantage of the flow character of the re-entrant process [3]. 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 branch and bound technique was utilized in [4,5] while the decomposition technique in solving maximum lateness problem for re-entrant flow shop with sequence dependent setup times was suggested in [6]. Mixed integer heuristic algorithms was later on elaborated in [7] for minimizing the makespan of a permutation flow shop scheduling problem. Significant works on re-entrant hybrid flow shop can be found in [8] while hybrid algorithms which combine a few well known techniques was reported in [9–11].

In scheduling literature, there are a number of studies conducted using the bottleneck approach in solving shop scheduling problem. This includes shifting bottleneck heuristic [12] and bottleneck minimal idleness heuristic [13,14]. Other related studies are the dispatching rule heuristic for proportionate flow shop [15] and flow shops with deteriorating jobs on no-idle dominant machine [16]. However, not much progress is reported on bottleneck approach in solving re-entrant flow shop problem. Among the few researches are [6] who developed a specific version of shifting bottleneck heuristic to solve the re-entrant flow shop sequence problem.

In this paper we explored and investigated an Internet-based collaborative design and manufacturing process scheduling which resembles a four machine permuta-

tion re-entrant flow shop. The study develops a makespan minimization heuristic using bottleneck approach known as Bottleneck Adjacent Matching 2 (BAM2) heuristic. This procedure is specifically intended for the cyber manufacturing centre (CMC) at Universiti Tun Hussein Onn Malaysia (UTHM) that allows the university to share the sophisticated and advanced machinery and software available at the university with the small and medium enterprises (SMEs) using Internet technology [17]. The remainder of the paper is organised as follows: In the next section, the CMC operations are described and followed by discussions on alternative makespan computation using bottleneck approach. The later sections explain the proposed heuristic. The two final sections evaluate the heuristic performance, summarize the findings and present some future heuristic development.

2. 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 [17]. 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 1. The places denoted by P22, P23, P24 and P25 in Figure 1 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 [1]. However, it can be noticed from the PN model that there are a few processes that share 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. One important characteristic observed at the CMC is that the processing time at the CNC machine centre or the M4 is always the longest. This means the M4 always shows dominant machine characteristic. Due to the re-entrant nature of the CMC process, the M4 dominant characteristic is identified as $M4 + M3 + M4$ or also recognized as $P4 + P5 + P6$ especially when the processing time is used in the discussion. It was also found out from the CMC operations data that the number of jobs at the CMC is ranged from minimum of 6 to maximum of 20.

3. Alternative Makespan Computation Using Bottleneck Approach

Referring to Figure 1, the permutation scheduling algorithm for the CMC can be written as the followings and is identified as Algorithm 1 [18]:

- Algorithm 1
 Let i = Transition number, process number or work centre number ($i=1,2,3,\dots,6$)
 j = Job number ($j=1,2,3,\dots,n$)
 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 $j=1,2,3,\dots,n$
 Start (i,j) = Max [Stop ($i,j-1$), Stop ($i-1,j$)] except Start (1,1) = initial starting time

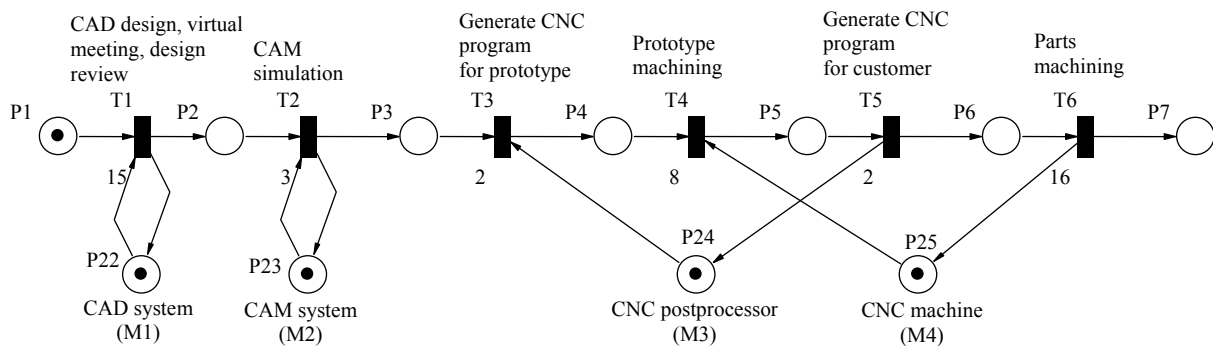


Figure 1. Petri net model of CMC activities

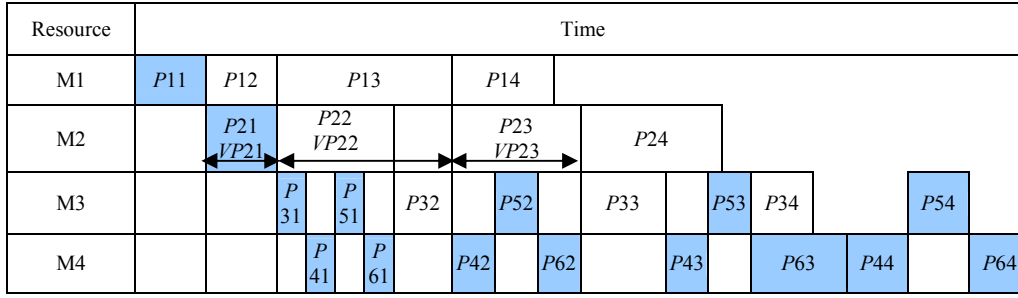


Figure 2. Example schedule focusing on M4

$$\text{Stop}(i,j) = \text{Start}(i,j) + P(i,j)$$

For $i=3,4$ and $j=1,2,3,\dots,n$

$$\text{Start}(i,j) = \text{Max}[\text{Stop}(i,j-1), \text{Stop}(i-1,j), \text{Stop}(i+2,j-1)]$$

$$\text{Stop}(i,j) = \text{Start}(i,j) + P(i,j)$$

The makespan for the CMC is computed using Algorithm 1 by determining the completion time of the last task belongs to the last job or $\text{Stop}(6,n)$. The example schedule for the CMC can also be observed by focusing on the M4 as the dominant machine and this is shown in Figure 2.

The makespan for the example in Figure 2 is computed as the following:

$$C_{max} = \sum_{i=1}^3 P(i,1) + \sum_{j=1}^n \sum_{i=4}^6 P(i,j) + \sum_{j=2}^n P4BCF(j) \quad (1)$$

where

$P4BCF = P4$ Bottleneck Correction Factor

$$\sum_{j=2}^n P4BCF(j)$$

$$= (\text{Gap between } P61 \text{ and } P42) + (\text{Gap between } P62 \text{ and } P43) + (\text{Gap between } P63 \text{ and } P44)$$

$$= \text{Max}[0, P32-P61, (VP21+P22+P32) - (P21+P31+P41+P51+P61)]$$

$$+ \text{Max}[0, P33-P62, (VP21+VP22+P23+P33) - (P21+P31+P41+P51+P61) - (P42+P52+P62)]$$

$$+ \text{Max}[0, P34-P63, (VP21+VP22+VP23+P24+P34) - (P21+P31+P41+P51+P61) - (P42+P52+P62+P43+P53+P63)]$$

The generalised equation for $P4BCF$ is described as follows:

For $j=2, P4BCF(j) = \text{Max}$

$$\left[0, \{P(3,j) - P(6,j-1)\}, \left\{ \sum_{i=2}^3 P(i,j) + VP(2,j-1) - \sum_{i=2}^6 P(i,1) \right\} \right] \quad (2)$$

For $j=3,4,\dots,n, P4BCF(j) = \text{Max}$

$$\left[0, \{P(3,j) - P(6,j-1)\}, \left\{ \sum_{i=2}^3 P(i,j) + \sum_{k=1}^{j-1} VP(2,k) - \sum_{i=2}^6 P(i,1) - \sum_{i=4}^6 \sum_{k=2}^{j-1} P(i,k) \right\} \right] \quad (3)$$

where,

$VP =$ Virtual processing

For $j = 1, VP(2,1) = \text{Max} [P(2,1), P(1,2)]$

For $j = 2,3,\dots,n-1,$

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

Virtual processing (VP) time is an imaginary processing time that assumes the starting time of any work process (WP) must begin immediately after the completion of the previous imaginary WP at the same work centre (WC). For the example in Figure 2, consider a job X starting on WC 2 (P22) and at the same time a job Y starts at WC 1 (P13). If the completion time of job X on WC 2 is earlier than the completion time of job Y at WC 1, under the imaginary concept, the VP of job X at WC 2 is extended from its actual processing time to match the completion time of job Y at WC 1. This means the VP of job X at WC 2 (or VP22) is equivalent to the processing time of job Y at WC 1 since the process at WC 2 for job Y can only be started immediately after the completion of Job Y at WC 1 regardless of the earlier completion time of job X at WC 2.

The accuracy of Equation (1) was tested with a total of 10,000 simulations 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 six job sequence for each simulation. The simulations were coded in VBA for Microsoft Excel. 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. The makespan from (1) were compared with makespan from Algorithm 1. The result of the simulation shows that 100% of the makespan values for both methods are the same. This indicates the accuracy of (1) in computing the makespan of the 6 job CMC operations scheduling. Equation (1) was also tested for computing the makespan for 10-job and 20-job CMC scheduling. All results indicate that (1) produces accurate makespan result.

4. Bottleneck Adjacent Matching 2 (BAM2) Heuristic

The Bottleneck Adjacent Matching 2 (BAM2) heuristic, which is thoroughly illustrated in this section, exploits the bottleneck limiting characteristics of the CMC process scheduling. The BAM2 heuristic will generate a

schedule which selects a job based on the best matching index to the previous job bottleneck processing time, which is the $P_4 + P_5 + P_6$ (or P_{456}) of the previous job. Ultimately, this minimizes the discontinuity time between the bottleneck machines and thus produces near-optimal schedule arrangement. The procedures to implement the BAM2 heuristic to the CMC scheduling are as the followings:

Step 1:

Select the job with the smallest value of $P(1,j) + P(2,j) + P(3,j)$ as the first job. If more than one job are having the same smallest value of $P(1,j) + P(2,j) + P(3,j)$, select the first job found to have the smallest $P(1,j) + P(2,j) + P(3,j)$ value.

This step is in accordance with (1), which indicated that minimum makespan can be achieved by assigning smallest $P(1,j) + P(2,j) + P(3,j)$ as first job.

Step 2:

Compute the BAM2 index for the potential second job selection by testing each of the remaining jobs as the second job. The BAM2 indexes are derived from the $P4BCF$ algorithm as in (2) and (3) and are computed as the followings:

For $j=2$, BAM2 index=

$$\text{Max} \left[\{P(3,j) - P(6,j-1)\}, \left\{ \sum_{i=2}^3 P(i,j) \right\} + VP(2,j-1) - \sum_{i=2}^6 P(i,1) \right] \quad (5)$$

For $j = 3, 4, \dots, n$, BAM2 index= Max

$$\left[\{P(3,j) - P(6,j-1)\}, \left\{ \sum_{i=2}^3 P(i,j) + \sum_{k=1}^{j-1} VP(2,k) - \sum_{i=2}^6 P(i,1) - \sum_{i=4}^6 \sum_{k=2}^{j-1} P(i,k) \right\} \right] \quad (6)$$

where j = remaining jobs to be selected one after the other

$j-1$ = the immediate preceding job that has been assigned

The value of VP is computed using (4).

Step 3:

Select the job that has zero BAM2 index. If no zero BAM2 index is available, select the job that has the largest negative BAM2 index (negative BAM1 index closest to zero). If no negative BAM2 index is available, select the job with the smallest positive BAM2 index. Assign this job for the current job scheduling. If more than one job have the same best index value, select the first job found to have the best index value from the jobs list.

Step 4:

Compute the BAM2 index for job scheduling assignment number 3, 4, ..., $n-1$ one after the other using algorithm at Step 2 and select the best job allocation using Step 3. Assign the last remaining job as the last job.

Step 5:

Compute the makespan of the completed job scheduling sequence using (1).

Step 6:

For the first completed schedule only, use the bottle-

neck scheduling performance 2 (BSP2) index to evaluate the schedule performance. This index is computed as the followings:

$$\text{BSP2 index} = \sum_{i=1}^3 P(i,1) + \sum_{j=2}^n P4BCF(j) \quad (7)$$

Excluding the first job in the completed schedule, identify other jobs which have the value of $\sum_{i=1}^3 P(i,j)$

less than the BSP2 index. Assign these jobs one after the other as the first job and repeat Step 2 to Step 5.

Step 7:

From the completed schedule arrangement list, select the schedule that produces the minimum makespan as the BAM2 heuristic solution.

5. BAM2 Heuristic Performance Evaluation

This section discusses the BAM2 heuristic performance evaluation under a few selected operating conditions. Since the P_{456} dominance level is the major characteristic being considered in developing the BAM2 heuristic, it is appropriate to test the performance of this heuristic under various groups of dominance level values. The dominance level is measured by observing how many times the value of $P_2+P_3+P_4+P_5+P_6$ of any job greater than $P_1+P_2+P_3$ of other jobs. Similar to [13], the dominance level groups are divided into levels of weak P_{456} dominance, medium P_{456} dominance and strong P_{456} dominance. The determination of the group dominance level ranges is solely depended on the value of the maximum possible P_{456} dominance level divided by 3. For the experimentation that uses 6 job analysis, the maximum possible P_{456} dominance level equals to $(n-1)n = (6-1)6 = 30$. The P_{456} dominance level range values are summarised in Table 1.

The performance evaluation was simulated using groups of 6 jobs waiting to be scheduled at the CMC. The selection of 6 jobs enables fast enumeration of all possible job sequences that can be used to compare with the BAM2 heuristic result. The processing time for each process was randomly generated using uniform distribution pattern on the realistic data ranges as in Table 2. During each simulation, data on P_1 dominance level, minimum makespan from BAM2 heuristic and optimum makespan from complete enumeration were recorded. The ratio between BAM2 heuristic makespan and the optimum makespan from enumeration was then computed for performance measurement. A total of 3000 simulations were conducted using the randomly generated data and the results are tabulated in Table 3.

The average makespan ratio in Table 3 represents the average ratio of the makespan from BAM2 heuristic to the optimum makespan from complete enumeration. The optimum result column registers the percentage of oc-

currences in which the makespan from BAM2 heuristic equals the optimum makespan from complete enumeration. The general results indicate that the BAM2 heuristic produces overall makespan solutions that are 1.7% above the optimum. This is shown by the overall average makespan ratio of 1.017. However, the result also suggested that the BAM2 heuristic is very effective in solving the scheduling problems within the strong P456 dominance level range. This is indicated by the average makespan ratio of 0.1% above the optimum at the strong P456 dominance level range. Moreover, it was also noted that at this dominance range, 89.47% of the solution generated by the heuristic are the optimum solutions. The percentage of optimum results decreases at the medium P456 dominance (42.26%) and the weak P456 dominance (47.37%).

For comparison purposes, a similar test was also conducted using the NEH heuristic, which is the best known heuristic for flow-shop scheduling [13,19] in predicting the job sequence that produces optimum makespan for the CMC. The result of this test is illustrated in Table 4.

Table 1. P456 dominance level groups

P456 Dominance Descriptions	Ranges of P456 Dominance Level (P456DL)
Weak	$0 \leq P456DL \leq (1/3)n(n-1)$
Medium	$(1/3)n(n-1) < P456DL \leq (2/3)n(n-1)$
Strong	$(2/3)n(n-1) < P456DL \leq n(n-1)$

Table 2. Process time data range (hours)

	$P(1,j)$	$P(2,j)$	$P(3,j)$	$P(4,j)$	$P(5,j)$	$P(6,j)$
Minimum	8	4	4	8	4	8
Maximum	150	16	16	60	16	60

Table 3. BAM2 heuristic performance for 6 job problems

P456 Dominance Level	Average Makespan Ratio	Optimum result (%)
Strong	1.001	89.47
Medium	1.020	42.26
Weak	1.017	47.37
Overall	1.017	51.17

Table 4. NEH heuristic performance for 6 job problems

P456 Dominance Level	Average Makespan Ratio	Optimum result (%)
Strong	1.0004	93.32
Medium	1.0001	98.04
Weak	1.00001	99.70
Overall	1.0002	97.63

Comparing Tables 4 and 3, it can be clearly seen that NEH heuristic produces good results and is superior to BAM2 heuristic in solving the CMC 6 job re-entrant flow shop problem. This indicates that for larger problems, where complete enumeration is not practical, NEH heuristic is an appropriate tool that can be used to measure the BAM2 performance. In analysing the six job problems, the makespan results from the BAM2 heuristic were also compared with the NEH heuristic. The result of this comparison is illustrated in Table 5.

From the makespan performance comparison between BAM2 and NEH in solving the CMC scheduling for 6 job problems (Table 5), it can be seen that BAM2 produces best result at strong P456 dominance level. Here, 84.82% of BAM2 results are the same with NEH, 5.47% of BAM2 results are better than NEH while 9.72% of BAM2 results are worse than NEH. Since this study considers NEH as the best and appropriate tool for BAM2 performance verification, it can be highlighted that at strong P456 dominance level, BAM2 produces 84.82% + 5.47% or 90.29% accurate result. This dominance level also produces average BAM2 makespan performance of 0.1% above the NEH makespan. Observations at Table 5 also suggest that BAM2 is less accurate in solving the CMC scheduling problem at both medium and weak P456 dominance level.

The BAM2 performance evaluation was also simulated using groups of 10 jobs waiting to be scheduled at the CMC. Similar with the 6 job test, the processing time for each process for the 10 job problems was randomly generated using uniform distribution pattern on the realistic data ranges as in Table 2. A total of 3000 simulations of 10 job problems using the randomly generated data were conducted. The simulation result analysis is presented in Table 6.

From Table 6, it can be seen that for 10 job problems, BAM2 also produces best result at strong P456 dominance level. Here 90.83% of BAM2 results are the same with NEH, 6.59% of BAM2 results are better than NEH while 2.58% of BAM2 results are worse than NEH. Overall, at the strong P456 dominance level BAM2 produces 90.83% + 6.59% or 97.42% accurate results that equal to or better than the NEH makespan results. This dominance level also produces average BAM2 makespan performance of 0.02% below the NEH makespan. Observations at Table 6 also suggest that BAM2 is less effective in solving the CMC 10 job scheduling problems at both medium and weak P456 dominance level.

A new simulation was also conducted to evaluate the capability of the BAM2 heuristic in estimating near optimal job sequences for CMC 20 job problems. A total of 1500 simulations of 20 job problems using the randomly generated data that fulfilled the typical processing time ranges at Table 2 were conducted. The simulation result analysis is presented in Table 7.

Table 5. BAM2 vs NEH makespan performance for 6 job problems

P456 Domi- nance Level	Average BAM2/NEH Ratio	BAM2 < NEH (%)	BAM2 = NEH (%)	BAM2 > NEH (%)
Strong	1.001	5.47	84.82	9.72
Medium	1.020	0.81	41.88	57.31
Weak	1.017	0	47.52	52.48
Overall	1.016	1.4	50.2	48.4

Table 6. BAM2 vs NEH makespan performance for 10 job problems

P456 Domi- nance Level	Average BAM2/NEH Ratio	BAM2 < NEH (%)	BAM2 = NEH (%)	BAM2 > NEH (%)
Strong	0.9998	6.59	90.83	2.58
Medium	1.011	7.50	40.44	52.06
Weak	1.015	4.36	21.51	74.13
Overall	1.010	7.03	44.13	48.83

Table 7. BAM2 vs NEH makespan performance for 20 job problems

P456 Domi- nance Level	Average BAM2/NEH Ratio	BAM2 < NEH (%)	BAM2 = NEH (%)	BAM2 > NEH (%)
Strong	0.9999	1.02	98.98	0
Medium	1.004	6.96	54.50	38.54
Weak	1.010	0.77	6.15	93.08
Overall	1.005	3.27	49.33	47.4

From Table 7, it can be seen that at strong P456 dominance level, BAM2 heuristic produces 98.98% makespan results equal to NEH, 1.02% results better than NEH while none of BAM2 results is worse than NEH. Overall, at the strong P456 dominance level BAM2 produces 100% results that are equal or better than NEH makespan results. This dominance level also produces average BAM2 makespan performance of 0.01% less than the NEH makespan.

6. Conclusions

In this paper, we explore and investigate the potential development of a bottleneck-based makespan algorithms and heuristic to minimize the makespan of an Internet-based collaborative design and manufacturing process that resembles a four machine permutation re-entrant flow shop with the process routing of M1,M2,M3,M4,M3,M4. It was shown that using bottleneck-based analysis, effective makespan algorithms and a constructive heuristic known as the BAM2 heuristic can be developed to solve for near-optimal scheduling sequence. The simulation results indicated that especially at strong P456 domi-

nance level, the BAM2 heuristic is capable to produce near optimal results for all the problem sizes studied. At strong P456 dominance level, this heuristic generates results which are very much compatible to the NEH. To some extent, in the specific 10 and 20 job problems simulation conducted during the study, the BAM2 shows better makespan performance compared to the NEH within the strong P456 dominance level. The bottleneck approach presented in this paper is not only valid for the CMC alone, but can also be utilised to develop specific heuristics for other re-entrant flow shop operation systems that shows significant bottleneck characteristics. With the successful development of the BAM2 heuristic, the next phase of this research is to further utilize the bottleneck approach in developing heuristic for optimizing the CMC scheduling for the medium and weak P456 dominance level.

7. Acknowledgments

This work was partially supported by the Fundamental Research Grant Scheme, Ministry of Higher Education, Malaysia (Cycle 1 2007 Vot 0368).

REFERENCES

- [1] M. Pinedo, "Scheduling: theory, algorithms, and systems," 2nd Edition, Upper Saddle River, Prentice-Hall, N.J., 2002.
- [2] 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*, Vol. 35, No. 5, pp. 851–861, 2008.
- [3] S. C. Graves, H. C. Meal, D. Stefek, and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *Journal of Operations Management*, Vol. 3, No. 4, pp. 197–207, 1983.
- [4] J. S. Chen, "A branch and bound procedure for the reentrant permutation flow-shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, Vol. 29, pp. 1186–1193, 2006.
- [5] S. W. Choi, and Y. D. Kim, "Minimizing makespan on a two-machine reentrant flowshop," *Journal of The Operational Research Society*, Vol. 58, pp. 972–981, 2007.
- [6] E. Demirkol, R. Uzsoy, "Decomposition methods for reentrant flow shops with sequence dependent setup times," *Journal of Scheduling*, Vol. 3, pp. 115–177, 2000.
- [7] J. C. Pan, and J. S. Chen, "Minimizing makespan in re-entrant permutation flow-shops," *Journal of Operation Research Society*, Vol. 54, pp. 642–653, 2003.
- [8] S. W. Choi, Y. D. Kim, and G. C. Lee, "Minimizing total tardiness of orders with reentrant lots in a hybrid flow-shop," *International Journal of Production Research*, Vol. 43, pp. 2049–2067, 2005.
- [9] S. W. Choi, and Y. D. Kim, "Minimizing makespan on an m-machine re-entrant flowshop," *Computers & Operations Research*, Vol. 35, No. 5, pp. 1684–1696, 2008.

- [10] J. S. Chen, J. C. H. Pan, and C. K. Wu, "Hybrid tabu search for re-entrant permutation flow-shop scheduling problem. *Expert Systems with Applications*," Vol. 34, No. 3, pp. 1924–1930, 2008.
- [11] J. S. Chen, J. C. H. Pan, and C. M. Lin, "Hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications*," Vol. 34, pp. 570–577, 2008.
- [12] S. Mukherjee, and A. K. Chatterjee, "Applying machine based decomposition in 2-machine flow shops," *European Journal of Operational Research*, Vol. 169, pp. 723–741, 2006.
- [13] 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*, Vol. 29, pp. 577–584, 2001.
- [14] 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*, Vol. 182, No. 1, pp. 947–957, 2006.
- [15] B. C. Choi, S. H. Yoon, and S. J. Chung, "Minimizing maximum completion time in a proportionate flow shop with one machine of different speed," *European Journal of Operational Research*, Vol. 176, No. 2, pp. 964–976, 2007.
- [16] M. B. Cheng, S. J. Sun, and L. M. He, "Flow shop scheduling problems with deteriorating jobs on no-idle dominant machines," *European Journal of Operational Research*, Vol. 183, pp. 115–124, 2007.
- [17] 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*, Vol. 34, pp. 365–372, 2006.
- [18] S. A. Bareduan, S. H. Hasan, and S. Ariffin, "Finite scheduling of collaborative design and manufacturing activity: a Petri net approach," *Journal of Manufacturing Technology Management*, Vol. 19, No. 2, pp. 274–288, 2008.
- [19] P. J. Kalczynski, and J. Kamburowski, "On the NEH heuristic for minimizing the makespan in permutation flow shops," *Omega*, Vol. 35, pp. 53–60, 2007.