Proceedings of the 38th International Conference on Computers and Industrial Engineering, Oct.31-Nov.2, 2008, Beijing, China, edited by Guoping Xia, Xiuquan Deng

BOTTLENECK ADJACENT MATCHING 1 (BAM1) HEURISTIC FOR RE-ENTRANT FLOW SHOP WITH DOMINANT MACHINE

Salleh Ahmad Bareduan
Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia
Malaysia
saleh@uthm.edu.my

Sulaiman Hj Hasan
Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia
Malaysia
sulaiman@uthm.edu.my

ABSTRACT

This paper presents a scheduling heuristic to minimize the makespan of a re-entrant flow shop using bottleneck analysis. The heuristic is specifically intended for the cyber manufacturing centre (CMC) which is an Internet-based collaborative design and manufacturing between the Universiti Tun Hussein Onn Malaysia and the small and medium enterprises. The CMC processes scheduling resembles a four machine permutation re-entrant flow shop with 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, an effective constructive heuristic can be developed to solve for near-optimal scheduling sequence. At strong machine dominance level and large job numbers, this heuristic shows slightly better makespan performance compared to the NEH. However, for smaller job numbers, NEH is superior.

KEYWORDS

Re-entrant flow shop, heuristic, dominant machine, bottleneck approach, scheduling

1 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 (Lian et al., 2008) and even a basic problem of $F3 \| C_{max}$ is already strongly NP-hard (Pinedo, 2002). 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, Graves et al. (1983) 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 (2000). Mixed integer heuristic algorithms was later on elaborated by Pan and Chen (2003) in minimizing makespan of a permutation flow shop scheduling problem. Significant works on re-entrant hybrid flow shop can be found in Yura (1999), Pearn et al. (2004) and Choi et al. (2005) while hybrid techniques which combine lower bound-based algorithm and idle time-based algorithm was reported by Choi and Kim (2008).

In scheduling literature, heuristics that utilize the bottleneck approach are known to be among the most successful methods in solving shop scheduling problems. These include shifting bottleneck heuristic (Adams et al., 1988), (Mukherjee and Chatterjee, 2006) and bottleneck minimal idleness heuristic

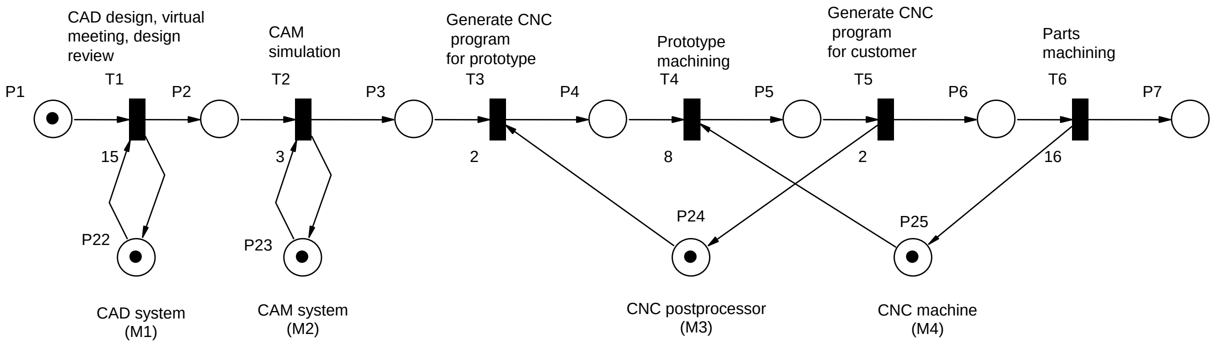

Figure 1 Petri Net Model of CMC activities

(Kalir and Sarin, 2001)(Wang et al., 2006). However, not much progress is reported on bottleneck approach in solving re-entrant flow shop problem. Among the few researches are Dermirkol and Uzsoy (2000) 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 develops a makespan minimization heuristic using bottleneck approach known as bottleneck adjacent matching 1 (BAM1) heuristic. This procedure is specifically intended for the cyber manufacturing centre at Universiti Tun Hussein Onn Malaysia (UTHM).

2 CYBER MANUFACTURING CENTRE

UTHM has recently developed a cyber manufacturing system (CMS) 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 (Bareduan et al., 2006). 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 CMS 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. 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 by Onwubolu (1996) and Pinedo (2002). However, it can be noticed from the PN model that two of the resources are being shared by two different processes. 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 by Graves et al. (1983). 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 by Yang et al. (2008).

3 BOTTLENECK ADJACENT MATCHING 1 (BAM1) HEURISTIC

The bottleneck adjacent matching 1 (BAM1) heuristic, which is thoroughly illustrated in this section, exploits the bottleneck limiting characteristics of the CMC process scheduling. The BAM1 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 $P4 + P5 + P6$ of the previous job (refer example problem in Table 1). Ultimately, this minimizes the discontinuity time between the bottleneck machines and thus produces near-optimal schedule arrangement. The procedures to implement the BAM1 heuristic to the CMC scheduling are as the followings:

Step 1:
Evaluate the $P1$ dominance level of $P(1,j)$ bottleneck as compared to $P(4,j) + P(5,j) + P(6,j)$ as described in the next section. This is to ensure that $P(4,j) + P(5,j) + P(6,j)$ is the dominant bottleneck because BAM1 heuristic is more appropriately applicable for this type of bottleneck. If $P(1,j)$ is the dominant bottleneck instead of $P(4,j) + P(5,j) + P(6,j)$, BAM1 heuristic will not produce good results.

Step 2:
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.

Step 3:
With the selected first job from Step 2, compute the BAM1 index for the potential second job selection by assuming one by one of the remaining jobs are to be set as the second job. This index is built based on the absolute bottleneck limiting characteristics. The BAM1 index can be computed as the followings:

$$\text{MAX}[\{P(3,j) - P(6,j\text{-}1)\};\ \sum_{i=1}^{3} P(i,j) - \sum_{i=2}^{6} P(i,j-1);\ \sum_{i=2}^{3} P(i,j) - \sum_{i=3}^{6} P(i,j-1)]$$

where j = remaining jobs to be selected one by one
j-1 = the immediate preceding job that has been assigned

Step 4:
Select the job that has zero BAM1 index. If no zero BAM1 index is available, select the job that has the largest negative BAM1 index (negative BAM1 index closest to zero). If no negative BAM1 index is available, select the job with the smallest positive BAM1 index. Assign this job for the current job scheduling. If more than one job are having the same best index value, select the first job found to have the best index value from the jobs list.

Step 5:
Compute the BAM1 index for job scheduling assignment number 3, 4….n-1 one by one using algorithm at Step 3 and select the best job allocation using Step 4.

Step 6:
Compute the makespan of the completed job scheduling sequence.

Step 7:
Use the bottleneck scheduling performance 1 (BSP1) index to roughly evaluate the performance of the selected schedule. This index is explained in the next section. If this BSP1 index evaluation suggests that there is other possible first job candidate that may generate better job schedule arrangement, assign this new candidate as the first job and repeat Step 3 to Step 6.

Step 8:
From the whole list of scheduling sequences established using the above procedure, select the sequence that produces the minimum makespan.

4 AN IILUSTRATIVE EXAMPLE OF BAM1 HEURISTIC

In order to illustrate the implementation of the BAM1 heuristic, let's consider the six jobs CMC processes data as in Table 1. First, the $P1$ dominance level is evaluated. This dominance level is measured by detecting the number of occurrences where $P1 + P2 + P3$ of any job is greater than $P2 + P3 + P4 + P5 + P6$ of other jobs. For example, the dominance level of $P(1,1)$ is measured by determining the number of occurrences where the value of $P(1,1) + P(2,1) + P(3,1)$ is greater than $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$ of j = 2,3,4,5 and 6. From Table 2, it can be noted that $P(1,1) + P(2,1) + P(3,1)$ is greater than $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$ at j = 2,3,5 and 6. This provide a dominance value of one each for $P(1,1)$ at j = 2,3,5 and 6 as indicated at column $P(1,1)DL$ in Table 3. The other dominance level belonging to $P(1,2)$, $P(1,3)$, $P(1,4)$, $P(1,5)$ and $P(1,6)$ can be computed using the same approach. The results are tabulated in Table 3. The overall $P1$ dominance level resulting from all $P(1,j)$ can be computed by adding all value in Table 3. Therefore the overall $P1$ dominance level equals to 11. Since there are more zeroes in Table 3 compared to value of one, this means that the bottleneck characteristic of $P(4,j) + P(5,j) + P(6,j)$ is more dominant compared to $P(1,j)$. As such, BAM1 can appropriately be used to solve this example problem. (Step 1)

Table 1 Process Time Data

Job	j	$P(1,j)$	$P(2,j)$	$P(3,j)$	$P(4,j)$	$P(5,j)$	$P(6,j)$
Job A	1	96	9	11	58	7	14
Job B	2	21	9	4	34	14	9
Job C	3	51	4	14	31	4	29
Job D	4	71	8	10	50	12	58
Job E	5	45	14	9	26	5	36
Job F	6	145	16	12	23	11	33

Table 2 Comparison of $\sum_{i=1}^{3} P(i,j)$ and $\sum_{i=2}^{6} P(i,j)$

Job	j	$\sum_{i=1}^{3} P(i,j)$	$\sum_{i=2}^{6} P(i,j)$
Job A	1	116	99
Job B	2	34	70
Job C	3	69	82
Job D	4	89	138
Job E	5	68	90
Job F	6	173	95

Table 3 Occurrence of $\sum_{i=1}^{3} P(i,j)$ greater than $\sum_{i=2}^{6} P(i,j)$ of other job

	$P(1,1)$ *DL*	$P(1,2)$ *DL*	$P(1,3)$ *DL*	$P(1,4)$ *DL*	$P(1,5)$ *DL*	$P(1,6)$ *DL*
j=1	-	0	0	0	0	1
j=2	1	-	0	1	0	1
j=3	1	0	-	1	0	1
j=4	0	0	0	-	0	1
j=5	1	0	0	0	-	1
j=6	1	0	0	0	0	-

The dominant bottleneck can also be decided by comparing the $P1$ dominance level value to the value of $n(n\text{-}1)/2$ where n equals the number of jobs available. If the $P1$ dominance level value is greater than $n(n\text{-}1)/2$, this indicates that $P(1,j)$ is the dominant bottleneck. Otherwise, $P(4,j) + P(5,j) + P(6,j)$ becomes the dominant bottleneck. Next, Table 4 is produced in order to detect the most suitable job to be assigned as first job in the schedule. Job B is selected as the first job because it has the smallest value $P(1,j) + P(2,j) + P(3,j)$. (Step 2)

In the next step, the BAM1 indexes for the second job selection are computed. The remaining jobs (Job A, C, D, E and F) which have not been assigned yet are being tested one by one in order to identify the best candidate for the second job position.

Table 4 Data For $P(1,j) + P(2,j) + P(3,j)$

Job	$P(1,j) + P(2,j) + P(3,j)$
Job A	116
Job B	34
Job C	69
Job D	89
Job E	68
Job F	173

In evaluating the BAM1 index for the second job candidate, the value of j=2 and j-1=1 are used and each of the remaining jobs is assigned as j=2 one at a time. Since Job B has been assigned to the first job, therefore j-1 belongs to Job B. As such, the example of second job BAM1 index for Job A is computed as the followings: (Step 3)

$$\mathrm{MAX}[\{P(3,j) - P(6,j\text{-}1)\};\ \sum_{i=1}^{3} P(i,j) - \sum_{i=2}^{6} P(i,j-1);\ \sum_{i=2}^{3} P(i,j) - \sum_{i=3}^{6} P(i,j-1)]$$

$$= \mathrm{MAX}[\{P(3,j)\text{-}P(6,j\text{-}1)\};\ \{P(1,j)+P(2,j)+P(3,j)\}\text{-}\{P(2,j\text{-}1)+P(3,j\text{-}1)+P(4,j\text{-}1)+P(5,j\text{-}1)+P(6,j\text{-}1)\};\ \{P(2,j)+P(3,j)\}\text{-}\{P(3,j\text{-}1)+P(4,j\text{-}1)+P(5,j\text{-}1)+P(6,j\text{-}1)\}]$$

$$= \mathrm{MAX}[\{P(3,\text{Job A})\text{-}P(6,\text{Job B})\};\ \{P(1,\text{Job A})+P(2,\text{Job A})+P(3,\text{Job A})\}\text{-}\{P(2,\text{Job B})+P(3,\text{Job B})+P(4,\text{Job B})+P(5,\text{Job B})+P(6,\text{Job B})\};\ \{P(2,\text{Job A})+P(3,\text{Job A})\}\text{-}\{P(3,\text{Job B})+P(4,\text{Job B})+P(5,\text{Job B})+P(6,\text{Job B})\}]$$

$$= \mathrm{MAX}[\{11\text{-}9\};\ \{96+9+11\}\text{-}\{9+4+34+14+9\};\ \{9+11\}\text{-}\{4+34+14+9\}]$$

$$= \mathrm{MAX}[2;\ 46;\ \text{-}41] = 46$$

The second job BAM1 index for other remaining jobs (Job C, D, E and F) can be computed by substituting Job A with Job C, D, E and F respectively in the above formulation. The results of computing all the second job BAM1 indexes are shown in Table 5. Since the zero BAM1 index belongs to Job E, therefore Job E is selected to be assigned as the second job. (Step 4)

Table 5 BAM1 Index Computation for Second Job

Job	$P(3,j) - P(6,j\text{-}1)$	$\sum_{i=1}^{3} P(i,j)$ - $\sum_{i=2}^{6} P(i,j-1)$	$\sum_{i=2}^{3} P(i,j)$ - $\sum_{i=3}^{6} P(i,j-1)$	BAM1 Index
Job B	-	-	-	-
Job A	2	46	-41	46
Job C	5	-1	-43	5
Job D	1	19	-43	19
Job E	0	-2	-38	0
Job F	3	103	-33	103

Since the second job has been assigned to Job E, then the next move is to evaluate the BAM1 index for third job assignment. Now, j-1 belongs to Job E and the example of the third job BAM1 index for Job A is computed as the followings: (Step 5)

= MAX[{P(3,j)-P(6,j-1)}; {P(1,j)+P(2,j)+P(3,j)}-{P(2,j-1)+P(3,j-1)+P(4,j-1)+P(5,j-1)+P(6,j-1)}; {P(2,j)+P(3,j)}-{P(3,j-1)+P(4,j-1)+P(5,j-1)+P(6,j-1)}]

= MAX[{P(3,Job A)-P(6,Job E)}; {P(1,Job A)+P(2,Job A)+P(3,Job A)}-{P(2,Job E)+P(3,Job E)+P(4,Job E)+P(5,Job E)+P(6, Job E)}; {P(2,Job A)+P(3, Job A)}-{P(3,Job E)+P(4,Job E)+P(5,Job E)+P(6, Job E)}]

= MAX[{11-36}; {96+9+11}-{14+9+26+5+36}; {9+11}-{9+26+5+36}]

= MAX[-25; 26; -56] = 26

The third job BAM1 index for other remaining jobs (Job C, D and F) can be computed by substituting Job A with Job C, D and F respectively in the above formulation. The results of computing all the third job BAM1 indexes are shown in Table 6. Because there is no zero BAM1 index value, therefore the largest negative value (closest to zero) is selected. This means that Job D is assigned as the third job.

With the assignment of Job D as the third job, the next steps are to compute the BAM1 index for fourth and fifth job respectively using the same previously described approach. The last remaining job is ultimately assigned to the sixth job. The recommended job sequence by using BAM1 index is therefore BEDACF. The makespan for this sequence can be computed using the conventional start and stop time analysis corresponding to the Petri net model in Figure 1 with strict permutation rule as described by Bareduan et al. (2008). The result is shown in Table 7 and it indicates that BEDACF job sequence generates a makespan of 524 hours. (Step 6)

Table 6 BAM1 Index Computation for Third Job

Job	$P(3,j) - P(6,j\text{-}1)$	$\sum_{i=1}^{3} P(i,j)$ - $\sum_{i=2}^{6} P(i,j-1)$	$\sum_{i=2}^{3} P(i,j)$ - $\sum_{i=3}^{6} P(i,j-1)$	BAM1 Index
Job B	-	-	-	-
Job E	-	-	-	-
Job A	-25	26	-56	26
Job C	-22	-21	-58	-21
Job D	-26	-1	-58	-1
Job F	-24	83	-48	83

The seventh step in implementing the BAM1 heuristic is the scheduling performance evaluation using the BSP1 index. Using the data from Table 8, the BSP1 index for BEDACF job arrangement is measured as the followings:

Table 7 Start And Stop Time for BEDACF Job Sequence

Job	j	Start $P(1,j)$	Stop $P(1,j)$	Start $P(2,j)$	Stop $P(2,j)$	Start $P(3,j)$	Stop $P(3,j)$	Start $P(4,j)$	Stop $P(4,j)$	Start $P(5,j)$	Stop $P(5,j)$	Start $P(6,j)$	Stop $P(6,j)$
Job B	1	0	21	21	30	30	34	34	68	68	82	82	91
Job E	2	21	66	66	80	82	91	91	117	117	122	122	158
Job D	3	66	137	137	145	145	155	158	208	208	220	220	278
Job A	4	137	233	233	242	242	253	278	336	336	343	343	357
Job C	5	233	284	284	288	343	357	357	388	388	392	392	421
Job F	6	284	429	429	445	445	457	457	480	480	491	491	524

$$\text{BSP1 index} = \text{Makespan} - \sum_{j=1}^{n}\sum_{i=4}^{6} P(i,j)$$
$$= 524 - [(34+14+9) + (26+5+36) + (50+12+58) + (58+7+14) + (31+4+29) + (23+11+33)]$$
$$= 524 - 454$$
$$= 70$$

Table 8 Process Time Data For BEDACF Job Sequence

Job	j	$P(1,j)$	$P(2,j)$	$P(3,j)$	$P(4,j)$	$P(5,j)$	$P(6,j)$
Job B	1	21	9	4	34	14	9
Job E	2	45	14	9	26	5	36
Job D	3	71	8	10	50	12	58
Job A	4	96	9	11	58	7	14
Job C	3	51	4	14	31	4	29
Job F	6	145	16	12	23	11	33

Referring to the BSP1 index evaluation in Table 9, it can be noticed that other than Job B (which is currently assigned as the first job), Job E and Job C are having $\sum_{i=1}^{3} P(i,j)$ values which are less than the current BSP1 index value of 70. Since there exist some $\sum_{i=1}^{3} P(i,j)$ values which are less than the current BSP1 index and they belong to the jobs which are not assigned as the first job in the current schedule, then there is a possibility that assigning these new candidates as the first job may result to better schedules. It is worth to try these new job arrangements. As such, two new schedule arrangements have to be established with Job C and Job E assigned as the first job.

Table 9 BSP1 Index Evaluation

Job	j	$\sum_{i=1}^{3} P(i,j)$
Job B	1	34
Job E	2	68
Job D	3	89
Job A	4	116
Job C	5	69
Job F	6	173
BSP1 index = 70		

With the new first job assignment for Job C and E, the next steps are to compute the BAM1 index and selecting appropriate job for the 2nd, 3rd, 4th, 5th and 6th job. This means steps 3 to 5 of the BAM1 heuristic procedure have to be repeated for both arrangements with Job C and E assigned as the first job. The second recommended job sequence by using BAM1 index is therefore CEDABF while the third sequence is EDACBF. The makespan for these sequences are 526 and 525 hours respectively.

The final step in implementing the BAM1 heuristic procedure is to select the best scheduling sequence from the entire recommended scheduling list. This is done by comparing the makespan of all the BAM1 listed scheduling sequences and selecting the schedule that produces the minimum makespan. The entire BAM1 listed scheduling sequences and their respective makespan for the example problem discussed in this section is illustrated in Table 10. From this table, the minimum makespan belongs to the scheduling sequence of BEDACF and therefore the BAM1 heuristic selects this sequence as the best solution. (Step 8)

Table 10 BAM1 List of Scheduling Sequences

No	J1	J2	J3	J4	J5	J6	Makespan
1	B	E	D	A	C	F	524
2	C	E	D	A	B	F	526
3	E	D	A	C	B	F	525

The results of the BAM1 heuristic above is verified by using complete enumeration of 720 different job sequences representing all the possible sequences for 6 job schedule. This enumeration resulted to a minimum makespan of 524 hours. 4.44% of the 720 job sequences were found resulting to this minimum makespan.

5 BAM1 HEURISTIC PERFORMANCE EVALUATION

This section discusses the simulated results of the BAM1 heuristic performance under a few selected operating conditions. Since the $P1$ dominance level is the major characteristic being considered in developing the BAM1 heuristic, therefore it is appropriate to test the performance of this heuristic under various groups of dominance level values. Similar to Kalir and Sarin (2001), the dominance level groups are divided into levels of weak $P1$ dominance, medium $P1$ dominance and strong $P1$ dominance. The determination of the groups $P1$ dominance level range are solely depended on the value of the maximum possible $P1$ dominance level divided by 3. For the experimentation that uses 6 job analysis, the maximum possible $P1$ dominance level equals to $(n-1)n = (6-1)6 = 30$. The $P1$ dominance level range values and its equivalent interpretations are summarised in Table 11. The equivalent interpretations is included in Table 11 because there is a relationship between dominance levels of *P*1 and *P*4+*P*5+*P*6 (*P*456) in which, weak *P*1 dominance can also be interpreted as strong *P*456 or vice-versa.

Table 11 *P*1 Dominance Level Groups

*P*1 Dominance Descriptions	Equivalent *P*456 Dominance Interpretation	Ranges of P1 Dominance Level (*P*1*DL*)
Weak	Strong	$0 \le P1DL \le (n-1)n/3$
Medium	Medium	$((n-1)n/3)+1 \le P1DL \le 2(n-1)n/3$
Strong	Weak	$(2(n-1)n/3)+1 \le P1DL \le (n-1)n$

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 BAM1 heuristic result. The processing time for each process is randomly generated using uniform distribution pattern on the realistic data ranges as in Table 12. During each simulation, data on *P*1 dominance level, minimum makespan from BAM1 heuristic and optimum makespan from complete enumeration are recorded. The ratio between BAM1 heuristic makespan and the optimum makespan from enumeration is then computed for performance measurement. A total of 3000 simulations were conducted using the randomly generated data in order to achieve an accuracy of 0.5% and confidence level of 99.7% on the average performance ratio. The results of the simulation are tabulated in Table 13.

Table 12 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 13 BAM1 Heuristic Performance for 6 Job Problem

*P*1 Dominance Level	*P*456 Dominance Level	Average Makespan Ratio	Perfect result (%)
Weak	Strong	1.001827	83.603239
Medium	Medium	1.028848	32.862575
Strong	Weak	1.01817	41.804511
Overall		1.022032	43.2

The average makespan ratio in Table 13 represents the average ratio of the makespan from BAM1 heuristic to the optimum makespan from complete enumeration. The perfect result column registers the percentage of occurrences in which the makespan from BAM1 heuristic exactly match the optimum makespan from complete enumeration. The general results indicate that the BAM1 heuristic produces near-optimal solutions for all categories of dominance level. This is shown by the overall makespan ratio value which means that on the average the BAM1 heuristic produces solutions that are 2.20% above the optimum. However, the result also suggested that the BAM1 heuristic is very effective in solving the scheduling problems within the strong *P*456 dominance (weak *P*1 dominance) level range. This is indicated by the average makespan ratio of 0.18% above the optimum. Moreover, it was also noted that at this dominance range, 83.60% of the solution generated by the heuristic are actually the optimum solutions. The percentage of perfect results decreases at the medium *P*456 dominance (32.86%) and the weak *P*456 dominance (41.80%).

For comparison purposes, a similar test was also conducted using the NEH heuristic, which is the best known heuristic for flow-shop scheduling (Kalir and Sarin, 2001)(Kalczynski and Kamburowski, 2007) in predicting the job sequence that produces optimum makespan for the CMC. The result of this test is illustrated in Table 14.

Table 14 NEH Heuristic Performance for 6 Job Problem

*P*1 Dominance Level	*P*456 Dominance Level	Average Makespan Ratio	Perfect result (%)
Weak	Strong	1.00041	93.319838
Medium	Medium	1.000146	98.044541
Strong	Weak	1.000011	99.699248
Overall		1.00016	97.633333

Comparing Table 14 and 13, it can be clearly seen that NEH heuristic produces good results and is superior to BAM1 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 BAM1 performance.

The BAM1 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 problem is randomly generated using uniform distribution pattern on the realistic data ranges as in Table 12. During each simulation, data on *P*1 dominance level, makespan from BAM1 heuristic and makespan from NEH heuristic are recorded. The ratio between BAM1 heuristic makespan and the NEH makespan is then computed for performance comparisons. A total of 3000 simulations of 10 job problems using the randomly generated data were conducted. The simulation result analysis is presented in Table 15.

From Table 15, it can be seen that for 10 job problems, BAM1 also produces highest accuracy result at strong *P*456 dominance level. Comparing to 6 job problems, here BAM1 produces better accuracy results with 85.25% of BAM1 results are the same with NEH, 4.10% of BAM1 results are better than NEH while 10.66% of BAM1 results are worse than NEH. Overall, at the strong *P*456 dominance level BAM1 produces 85.25% + 4.10% or 89.35% results that match or better than NEH makespan results. This dominance level also produces average BAM1 makespan performance of 0.0297% above the NEH makespan. Observations at Table 15 also suggest that BAM1 is less accurate in solving the CMC 10 job scheduling problem at both medium and weak P456 dominance level. Medium P456 dominance level registers 34.13% (30.03% + 4.10%) accurate BAM1 results while weak P456 dominance level experiences 24.41% (17.06% + 7.35%) accurate BAM1 results.

Table 15 BAM1 vs NEH Makespan Performance for 10 Job Problem

*P*456 Dominance Level	Average BAM1/NEH Ratio	BAM1 < NEH (%)	BAM1 = NEH (%)	BAM1 > NEH (%)
Strong	1.000297	4.09836	85.24590	10.65573
Medium	1.019532	4.09764	30.03487	65.86748
Weak	1.016573	7.35294	17.05882	75.58823
Overall	1.016851	4.46666	35.3	60.23333

Better BAM1 results at the CMC 10 job problems compared to the 6 job problems motivates further analysis for larger scheduling problems. A new simulation was conducted to evaluate the capability of the BAM1 heuristic in estimating near optimal job sequences for CMC 20 job problems. The simulation result analysis is presented in Table 16.

Table 16 BAM1 vs NEH Makespan Performance for 20 Job Problem

*P*456 Dominance Level	Average BAM1/NEH Ratio	BAM1 < NEH (%)	BAM1 = NEH (%)	BAM1 > NEH (%)
Strong	0.9999904	1.28866	97.93814	0.77319
Medium	1.009927	3.9801	41.45937	54.56053
Weak	1.009991	0.58939	8.44793	90.96267
Overall	1.007378	2.13333	44.86666	53

From Table 16, it can be seen that at strong *P*456 dominance level, BAM1 heuristic produces 97.94% makespan results equal to NEH, 1.29% results better than NEH while 0.77% of BAM1 results are worse than NEH. Overall, at the strong *P*456 dominance level BAM1 produces 99.23% (97.94% + 1.29%) results that are equal or better than NEH makespan results. This dominance level also produces average BAM1 makespan performance of 0.001% less than the NEH makespan.

6 CONCLUSION

In this paper, we explore and investigate the potential development of a bottleneck-based heuristic to minimise the makespan of a four machine permutation re-entrant flow shop with the process routing of M1,M2,M3,M4,M3,M4. It was shown that especially at strong *P*456 dominance level, the BAM1 heuristic is capable to produce near optimal results for all the problem sizes studied. At strong *P*456 dominance level and large job numbers (n=20), this heuristic generates results which are very much compatible to the NEH. To some extent, in the specific 20 job problems simulation conducted during the study, the BAM1 shows slightly better average makespan performance compared to the NEH. However, for smaller job numbers (n=6 and 10), NEH is superior. 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 BAM1 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 *P*456 dominance level.

ACKNOWLEDGMENTS

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

REFERENCES

Adams J, Balas E, Zawack D (1988), "The shifting bottleneck procedure for job shop scheduling", Management Science, Vol. 34, pp. 391-401

Bareduan, S.A., Hasan, S.H. and Ariffin, S. (2008), "Finite Scheduling of Collaborative Design and Manufacturing Activity: a Petri Net Approach", Journal of Manufacturing Technology Management, Vol. 19(2), pp. 274-288.

Bareduan SA, Hasan SH, Rafai NH and Shaari MF (2006), "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

Choi SW, Kim YD, Lee GC, (2005), "Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop", International Journal of Production Research, Vol. 43, pp.2049-2067

Choi SW, Kim YD (2008), "Minimizing makespan on an *m*-machine re-entrant flowshop", Computers & Operations Research, Vol. 35(5), pp. 1684-1696

Demirkol E, Uzsoy R (2000), "Decomposition methods for reentrant flow shops with sequence dependent setup times", Journal of Scheduling, Vol. 3, pp. 115-177

Graves SC, Meal HC, Stefek D, Zeghmi AH (1983), "Scheduling of re-entrant flow shops", Journal of Operations Management, Vol. 3(4), pp. 197-207

Kalczynski PJ, Kamburowski J (2007), "On the NEH heuristic for minimizing the makespan in permutation flow shops", The International Journal of Management Science (Omega), Vol. 35, pp. 53-60

Kalir AA, Sarin SC (2001), "A near optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots", The International Journal of Management Science (Omega), Vol. 29, pp. 577-584

Lian Z, Gu X, Jiao B (2008), "A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan", Chaos, Solitons and Fractals, Vol. 35(5), pp. 851-861

Mukherjee S, Chatterjee AK (2006), "Applying machine based decomposition in 2-machine flow shops", European Journal of Operational Research, Vol. 169, pp. 723-741

Onwubolu GC (1996), "A flow-shop manufacturing scheduling system with interactive computer graphics" International Journal of Operations & Production Management, Vol. 16(9), pp. 74-84

Pan JC, Chen JS (2003), "Minimizing makespan in re-entrant permutation flow-shops", Journal of Operation Research Society, Vol. 54, pp. 642-653

Pearn WL, Chung SH, Chen AY, Yang MH (2004), "A case study on the multistage IC final testing scheduling problem with reentry", International Journal of Production Economics, Vol. 88, pp. 257-267

Pinedo M (2002), "Scheduling: Theory, algorithms, and systems", 2nd ed., Upper Saddle River, N.J., Prentice-Hall

Wang JB, Shan F, Jiang B, Wang LY (2006), "Permutation flow shop scheduling with dominant machines to minimize discounted total weighted completion time", Applied Mathematics and Computation, Vol. 182(1), pp. 947-957

Yang DL, Kuo WH, Chern MS (2008), "Multi-family scheduling in a two-machine re-entrant flow shop with setups", European Journal of Operational Research, Vol. 187(3), pp. 1160-1170

Yura K (1999), "Cyclic scheduling for re-entrant manufacturing systems", International Journal of Production Economics, Vol. 60(61), pp. 523-528