

Bottleneck Adjacent Matching 3 (BAM3) Heuristic For Re-Entrant Flow Shop With Dominant Machine

S. A. Bareduan, S. H. Hasan

Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia

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 first process at M1 has high tendency of exhibiting dominant 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 medium to high job numbers, this heuristic shows slightly better makespan performance compared to the NEH. However, for smaller job numbers, NEH is superior.

Keywords – Bottleneck, dominant machine, heuristic, re-entrant flow shop, Scheduling

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] 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 [4]. Mixed integer heuristic algorithms was later on elaborated by [5] in minimizing the makespan of a permutation flow shop scheduling problem. Significant works on re-entrant hybrid flow shop can be found in [6,7,8] while hybrid techniques which combine lower

bound-based algorithm and idle time-based algorithm was reported by [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 [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 develops a makespan minimization heuristic using bottleneck approach known as bottleneck adjacent matching 3 (BAM3) heuristic. This procedure is specifically intended for the cyber manufacturing centre 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 [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.

At the CMC, all jobs must go through six processes and four machines according to a fixed sequence. This is known as flow shop manufacturing as described by some researchers [2,15]. However, it is also noticed that some processes enter the equipments more than one time similar to the re-entrant flow shop described by [3]. Since the CMC utilised strict permutation rule in organising the schedule, the overall problem can be identified as four machine permutation re-entrant flow shop with the processing route of M1,M2,M3,M4,M3,M4 as similarly described by [16] and the bottlenecks normally present at either M1(1st process) or M4+M3+M4 (4th, 5th and 6th processes).

II. BOTTLENECK ADJACENT MATCHING 3 (BAM3) HEURISTIC

The bottleneck adjacent matching 3 (BAM3) heuristic, which is thoroughly illustrated in this section,

exploits the bottleneck limiting characteristics of the CMC process scheduling. The BAM3 considers the bottleneck exists at the first process of the CMC which is $P(1,j)$. This heuristic will generate a schedule which selects a preceding job based on the best matching index to the current job bottleneck processing time, which is the P_1 of the current job or $P(1,j)$. The BAM3 functions to minimize the discontinuity time between the bottleneck machine of the current job scheduled and its subsequent processes in order to produce near-optimal schedule arrangement. The procedures to implement the BAM3 heuristic to the CMC scheduling are as the followings:

Step 1:

Evaluate the bottleneck dominance level of $P(1,j)$ compared to $P(4,j) + P(5,j) + P(6,j)$ as described in the next section. This is to ensure that $P(1,j)$ is the dominant bottleneck because BAM3 heuristic is more appropriately applicable for this type of bottleneck. If $P(4,j) + P(5,j) + P(6,j)$ instead of $P(1,j)$ is the dominant bottleneck, BAM3 heuristic will not produce good results.

Step 2:

Select the job with the smallest value of $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$ as the last job (6^{th} job for the example problem in Table 1 since this table consists of six jobs). If more than one job are having the same smallest value of $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$, select the first job found to have the smallest value and assign it as the last job in the schedule. Other potential candidates for the last job position will be evaluated in Step 7.

Step 3:

With the selected last job (6^{th} job) as in Step 2, compute the BAM3 index for the potential 5^{th} job (second last job) by assuming one by one of the remaining jobs are to be assigned as the 5^{th} job. This index is built based on the absolute bottleneck limiting characteristics. The BAM3 index can be computed as the followings:

$$\text{MAX}[\{P(2,j-1) - P(1,j)\}, \sum_{i=2}^5 P(i,j-1) - \sum_{i=1}^2 P(i,j), \sum_{i=2}^6 P(i,j-1) - \sum_{i=1}^3 P(i,j)]$$

where j = the job that has been assigned ($j = n, n-1, \dots, 4, 3$).

For evaluating the BAM3 index for the 5^{th} job, set $j = 6$ ($j=6$ belongs to the job that has been assigned as the 6^{th} job from Step 2).

$j-1$ = the immediate preceding job that is to be assigned. Each remaining job is one by one assumed to be the candidate for $j-1$.

Step 4:

Select the job that has zero BAM3 index. If no zero BAM3 index is available, select the job that has the largest negative BAM3 index (negative BAM3 index closest to zero). If no negative BAM3 index is available, select the job with the smallest positive BAM3 index. Assign this job for the current job scheduling. If two or

more jobs are sharing the best index value, select the first found best BAM3 index from the jobs list.

Step 5:

Compute the BAM3 index for job scheduling assignment number 4, 3, and 2 one by one using the algorithm at Step 3 and select the best job allocation using Step 4.

Step 6:

Compute the makespan from the completed job scheduling arrangement.

Step 7:

Use the bottleneck scheduling performance 3 (BSP3) index to evaluate the performance of the selected schedule. This index is explained in the example implementation section. If this BSP3 index evaluation suggests that there is other possible last job candidate that may generate better job schedule arrangement, assign these new candidates one by one as the last job and repeat Step 3 to Step 6.

Step 8:

From the entire completed schedule arrangement list, select the schedule that produces the minimum makespan as the best schedule.

III. AN ILLUSTRATIVE EXAMPLE OF BAM3 HEURISTIC

In order to illustrate the implementation of the BAM3 heuristic, let's consider the six jobs CMC processes data as in Table 1. First, the P_1 bottleneck dominance level is evaluated. This dominance level is measured by detecting the number of occurrences where $P_1 + P_2 + P_3$ of any job is greater than $P_2 + P_3 + P_4 + P_5 + P_6$ of other jobs. Table 2 shows the values of $P(1,j) + P(2,j) + P(3,j)$ versus $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$ and these values are then utilised to generate the $P(1,j)$ dominance level values as in Table 3. The overall P_1 bottleneck dominance level resulting from all $P(1,j)$ can be computed by adding all values in Table 3. Therefore the overall P_1 bottleneck dominance level equals to 23. Since there are more values

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	139	12	16	8	15	24
Job B	2	92	14	11	57	5	34
Job C	3	78	8	7	42	8	18
Job D	4	106	15	11	55	5	19
Job E	5	22	6	12	10	4	11
Job F	6	134	14	10	27	16	17

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	167	75
Job B	2	117	121
Job C	3	93	83
Job D	4	132	105
Job E	5	40	43
Job F	6	158	84

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
$j=1$	-	1	1	1	0	1
$j=2$	1	-	0	1	0	1
$j=3$	1	1	-	1	0	1
$j=4$	1	1	0	-	0	1
$j=5$	1	1	1	1	-	1
$j=6$	1	1	1	1	0	-

of one in Table 3 compared to zeroes, this means that the bottleneck characteristic of $P(1,j)$ is more dominant compared to $P(4,j) + P(5,j) + P(6,j)$. As such, it is appropriate to use BAM3 to solve the scheduling problem. (Step 1)

From Table 2, it is noticed that the smallest $P(2,j) + P(3,j) + P(4,j) + P(5,j) + P(6,j)$ value belongs to Job E. Therefore, Job E is selected as the last job. (Step 2)

In the next step, the BAM3 index for the 5th job (second last job) is computed for each of the remaining jobs. This is shown in Table 5. By setting $j=6$ ($j=6$ belongs to Job E since it has been assigned as the last job), and assuming Job A is to be assigned as 5th job ($j-1=5$ =Job A), the value of $P(2,j-1) - P(1,j)$ for Job A is equal to $P(2,A) - P(1,E)$. Referring to Table 1, this value is equal to $12-22 = -10$. The value of $\sum_{i=2}^5 P(i,j-1) -$

$\sum_{i=1}^2 P(i,j)$ for Job A is equal to $\{P(2,A) + P(3,A) + P(4,A) + P(5,A)\} - \{P(1,E) + P(2,E)\} = (12+16+8+15)-(22+6) = 23$. Finally, the value of $\sum_{i=2}^6 P(i,j-1) -$

$\sum_{i=1}^3 P(i,j)$ for Job A is equal to $\{P(2,A) + P(3,A) + P(4,A) + P(5,A) + P(6,A)\} - \{P(1,E) + P(2,E) + P(3,E)\} = (12+16+8+15+24)-(22+6+12) = 35$. Since the largest numbers for the BAM3 index components belonging to Job A is equal to 35, therefore the BAM3 index for Job A to be assigned as the 5th job is 35. Using the same method, the 5th job BAM3 index for Job B can be computed by assuming $j-1=$ Job B. Similarly, the BAM3 indexes for Jobs C, D and F can be computed and this is summarised in Table 5. Since there is no zero or negative BAM3 index value, therefore the positive values are to be considered. From this table, the smallest positive value for the BAM3 index belongs to Job A. This means Job A is assigned as the 5th job. (Steps 3 and 4)

Table 5 : BAM3 Index Computation for 5th Job

Job	$P(2,j-1) - P(1,j)$	$\sum_{i=2}^5 P(i,j-1) - \sum_{i=1}^2 P(i,j)$	$\sum_{i=2}^6 P(i,j-1) - \sum_{i=1}^3 P(i,j)$	BAM3 Index
Job A	-10	23	35	35
Job B	-8	59	81	81
Job C	-14	37	43	43
Job D	-7	58	65	65
Job F	-8	39	44	44
Job E	-	-	-	-

With the assignment of Job A as the 5th job, the next steps are to compute the BAM3 index for the 4th, 3rd, and 2nd job respectively (Step 5). The remaining job is ultimately assigned to the 1st job. The recommended job sequence by using BAM3 index is therefore CFDBAE. The makespan for this sequence can be computed using the conventional start and stop time analysis corresponding to the CMC re-entrant flow shop with strict permutation rule as described by [17]. The result indicates that CFDBAE job sequence generates a makespan of 649 hours. (Step 6)

The seventh step in implementing BAM3 heuristic is the scheduling performance evaluation using the BSP3 index. This index is measured as the followings:

$$\text{BSP3 index} = \text{Makespan} - \sum_{j=1}^n P(1,j)$$

Therefore, for the CFDBAE scheduling arrangement:

$$\begin{aligned} \text{BSP3 index} &= \text{Makespan} - \{P(1,1) + P(1,2) + P(1,3) + \\ &\quad P(1,4) + P(1,5) + P(1,6)\} \\ &= 649 - \{78 + 134 + 106 + 92 + 139 + 22\} \\ &= 78 \end{aligned}$$

If there exist from any job a $\sum_{i=2}^6 P(i,j)$ value which

is less than the current BSP3 index and it belongs to the job which is not assigned as the last job in the current schedule, then there is a possibility that assigning this candidate as the last job may result to a better schedule. It is worth to try this new job arrangement. Table 2 shows

the value of $\sum_{i=2}^6 P(i,j)$ for all jobs and it is noted that

the only job (other than Job E) that is having $\sum_{i=2}^6 P(i,j)$

less than the current BSP3 index of 78 is Job A. Therefore, a new schedule arrangement has to be established with Job A is to be assigned as the last job (Step 7). The Step 3 to 6 of BAM3 heuristic has to be repeated.

With Job A assigned as the 6th job, the next step is to select the 5th, 4th, 3rd, 2nd, and 1st job candidate using BAM3 index. The second recommended job sequence by

using BAM3 index is therefore ECFDBA. The makespan for this sequence is equal to 646 hours. Comparing to the first BAM3 recommended scheduling arrangement of CFDBAE which resulted to a makespan of 649 hours, the new ECFDBA scheduling arrangement produces better makespan result. As such, BAM3 heuristic will select ECFDBA as its best scheduling solution (Step 8). This BAM3 heuristic result can be verified by comparing its makespan value to the minimum makespan value obtained using complete enumeration representing all 720 possible sequences for 6 jobs schedule. This enumeration is found resulting to a minimum makespan of 646 hours. This means that for the example 6 job problems discussed in this section, the BAM3 heuristic is capable to produce sound and accurate result.

IV. BAM3 HEURISTIC PERFORMANCE EVALUATION

This section discusses the simulated results of BAM3 heuristic performance under a few selected operating conditions. Similar to [12], the results are categorised into three levels of weak, medium and strong $P1$ dominance as shown in Table 8 where n equals number of jobs.

The performance evaluation was first simulated using groups of 6 jobs waiting to be scheduled at the CMC. The processing time for each process is randomly generated using uniform distribution pattern on the realistic data ranges as in Table 9. During each simulation, data on $P1$ dominance level, minimum makespan from BAM3 heuristic and makespan from NEH heuristic (heuristic from Nawaz, Enscore and Ham) [18] were recorded. The ratio between BAM3 heuristic makespan and the NEH makespan was then computed for performance comparisons. A total of 3000 simulations were conducted. Table 10 shows the makespan performance comparison between BAM3 and NEH in solving the CMC scheduling for 6 job problems. It can be seen that BAM3 produces highest accuracy result at strong $P1$ dominance level. Here, 89.77% of BAM3 results are the same with NEH, 0.3% of BAM3 results are better than NEH while 9.92% of BAM3 results are worse than NEH. Since this study considers NEH as the best known heuristic for flow-shop scheduling [12,18] and appropriate tool for BAM3 performance verification, it can be said that at strong $P1$ dominance level, BAM3 produces 89.77% + 0.3% or 90.07% accurate result. This dominance level also produces average BAM3 makespan performance of 0.172% above the NEH makespan. Observations at Table 10 also suggest that BAM3 is less accurate in solving the CMC scheduling problem at both medium and weak $P1$ dominance level. Medium $P1$ dominance level registers 36.12% + 0.16% accurate BAM3 results while weak $P1$ dominance level experiences 26.52% + 0.81% accurate BAM3 results.

Table 9 : 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 8 : $P1$ Dominance Level Groups

$P1$ Dominance Description	Equivalent $P456$ Dominance Interpretation	Ranges of $P1$ Dominance Level ($P1DL$)
Weak	Strong	$0 \leq P1DL \leq n(n-1)/3$
Medium	Medium	$n(n-1)/3 < P1DL \leq 2n(n-1)/3$
Strong	Weak	$2n(n-1)/3 < P1DL \leq n(n-1)$

Table 10 : BAM3 vs NEH Makespan Performance for 6 Job Problems

$P1$ Dominance Level	Average BAM3/NEH Ratio	BAM3 < NEH (%)	BAM3 = NEH (%)	BAM3 > NEH (%)
Weak	1.032591	0.809717	26.518219	72.672065
Medium	1.032334	0.162955	36.121673	63.715372
Strong	1.001718	0.300752	89.774436	9.924812
Overall	1.025590	0.3	46.433333	53.266667

The BAM3 performance evaluation was also simulated using groups of 10 jobs. The simulation result analysis is presented in Table 12.

Table 12 : BAM3 vs NEH Makespan Performance for 10 Job Problems

$P1$ Dominance Level	Average BAM3/NEH Ratio	BAM3 < NEH (%)	BAM3 = NEH (%)	BAM3 > NEH (%)
Weak	1.028596	0.120048	16.086435	83.793517
Medium	1.035157	4.0	21.142857	74.857143
Strong	0.999690	9.579230	88.630260	1.790510
Overall	1.020129	5.0	44.866667	50.133333

From Table 12, it can be seen that for 10 job problems, BAM3 produces highest accuracy result at strong $P1$ dominance level. Comparing to 6 job problems, here BAM3 produces better accuracy results. Overall, at the strong $P1$ dominance level BAM3 produces 88.63% + 9.58% or 98.21% results that match or better than NEH makespan results. This dominance level also produces average BAM3 makespan performance of 0.031% below the NEH makespan.

A new simulation was conducted to evaluate the capability of the BAM3 heuristic in estimating near optimal job sequences for CMC 20 job problems. The simulation result analysis is presented in Table 13.

Table 13 : BAM3 vs NEH Makespan Performance for 20 Job Problems

$P1$ Dominance Level	Average BAM3/NEH Ratio	BAM3 < NEH (%)	BAM3 = NEH (%)	BAM3 > NEH (%)
Weak	1.017797	0	4.123711	95.87628
Medium	1.022831	0.99502	17.57877	81.42620
Strong	0.999997	1.57170	98.03536	0.392927
Overall	1.013781	0.933333	41.4	57.666667

From Table 13, it can be seen that at strong $P1$ dominance level, BAM3 heuristic produces 98.03%

makespan results equal to NEH, 1.57% results better than NEH while 0.39% of BAM3 results are worse than NEH. Overall, at the strong P_1 dominance level BAM3 produces 99.6% (98.03% + 1.57%) results that are equal or better than NEH makespan results. This dominance level also produces average BAM3 makespan performance of 0.0003% less than the NEH makespan.

V. 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_1 dominance level, the BAM3 heuristic is capable to produce near optimal results for all the problem sizes studied. At strong P_1 dominance level and medium to high job numbers ($n=10$ and 20), 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 BAM3 shows slightly better average makespan performance compared to the NEH. However, for smaller job numbers ($n=6$), 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 BAM3 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_1 dominance level.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] 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
- [2] Pinedo M (2002), "Scheduling: Theory, algorithms, and systems", 2nd ed., Upper Saddle River, N.J., Prentice-Hall
- [3] 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
- [4] Demirkol E, Uzsoy R (2000), "Decomposition methods for reentrant flow shops with sequence dependent setup times", Journal of Scheduling, Vol. 3, pp. 115-177
- [5] Pan JC, Chen JS (2003), "Minimizing makespan in re-entrant permutation flow-shops", Journal of Operation Research Society, Vol. 54, pp. 642-653
- [6] Yura K (1999), "Cyclic scheduling for re-entrant manufacturing systems", International Journal of Production Economics, Vol. 60(61), pp. 523-528
- [7] 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
- [8] 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
- [9] Choi SW, Kim YD (2008), "Minimizing makespan on an m -machine re-entrant flowshop", Computers & Operations Research, Vol. 35(5), pp. 1684-1696
- [10] Adams J, Balas E, Zawack D (1988), "The shifting bottleneck procedure for job shop scheduling", Management Science, Vol. 34, pp. 391-401
- [11] Mukherjee S, Chatterjee AK (2006), "Applying machine based decomposition in 2-machine flow shops", European Journal of Operational Research, Vol. 169, pp. 723-741
- [12] 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
- [13] 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
- [14] 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
- [15] 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
- [16] 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
- [17] 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.
- [18] 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