Bottleneck-Based Heuristic for Permutation Flowshop Scheduling

N.A. Isa*\textsuperscript{a}, S.A. Bareduan\textsuperscript{b} and A.S. Zainudin\textsuperscript{c}

\textsuperscript{1} Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

E-mail: *nooramira1992@gmail.com, \textsuperscript{b}saleh@uthm.edu.my, \textsuperscript{c}amirasyu27@gmail.com

Abstract. This paper addresses the permutation flowshop scheduling problem with objective of makespan minimization. It proposes a heuristic that apply bottleneck-based identification technique and Nawaz, Enscore and Ham (NEH) insertion technique. This experiment was carried out using measurement of four-machines and six-jobs simulated in Microsoft Excel Simple Programming. The computational result show that bottleneck-based (BNB) achieved 60 % and more of optimum solution and heuristic validation result prove that the proposed heuristic resulting to be much better than existing heuristic by achieving more best solution of 6 sets out of 10 sets replication.

1. Introduction

Scheduling is an act of planning and prioritizing activities with the time limit for the job completion in order to meet with certain requirement, constraints or to achieve the goal of objectives [1]. Since the time always be the greatest constraint, it is why the scheduling is important to organize the activities efficiently and optimally. In worldwide industries, scheduling plays an important role in the production system since the resources are becoming more critical to be controlled. The resources are referred to the machine, manpower, material and many more. The greatest outcome can be gained if the scheduling of the resources is successfully organized [2]. It will be a great advantage to the manufacturer in worldwide industries if the latest research were successful to perform, easy to understand and easily applied to the current manufacturing system.

Permutation flowshop scheduling problem (PFSP) is one of the best known production scheduling problems with the same job order on all machines which has an intense engineering background [18]. Johnson [4] becomes the pioneer by introducing the Johnson’s rule with two machine problem. However, the general problem has been proved to be strongly NP-hard when involving more than two machine problem, m > 2 [4-8]. Minimization of makespan becomes one of the most attractive objective where it leads to the development of many approximate algorithms. NEH heuristic appeared around three decades ago are known as the highest performing method in minimizing the makespan for permutation flowshop scheduling problem. NEH algorithm was proposed by Nawaz, Enscore and Ham [5-7], [17]. This algorithm gave a highest attention on the job with larger total processing time where it should have higher priority. The jobs are arranged in decreasing order of the total processing time and
the 3rd job in the job list is inserted into the previous partial sequence at the position that gives the best mean flow time or makespan. Many researchers have used NEH heuristic for comparison purpose since it had appeared to be the best heuristic if the makespan criterion is considered [6-7]. Some of them also modified the NEH heuristic following to their objective function, [5], [12], [15-17], [19], [21-22].

In the scheduling environment, shifting bottleneck model has been widely used in sorting the machine with the highest increase of objective function value [9]. According to Zhenqiang and Peng [10], identification of bottleneck resources helps in scheduling the jobs rationally in reducing the difficulty in follow-up scheduling and also ensures the feasibility and effectiveness of scheduling result. Basically, the main idea is to schedule the jobs at the bottleneck stage where it may affect the performance of a heuristic for scheduling jobs in all the stages [11]. In the study of Modrak and Pandian [12] bottleneck machines were detected based on the machine workload, utilization rate or idle time length which bottleneck involvement significantly affect the quality of final solution. Utilization bottleneck heuristic approach is known to be among the most successful methods in solving shop scheduling problem including the shifting bottleneck heuristic and bottleneck minimal idleness heuristic [13].

In this paper, we investigated the bottleneck based concept in finding a good job arrangement for flowshop scheduling problem with the objective of makespan minimization. This study performed the concept by using a small problem size of four-machines and six-jobs and was carried out by using Microsoft Excel simple programming in Visual Basic for Application (VBA). This concept was applied on 100 sets of randomly generated data in 10 replications. The effectiveness of this concept was evaluated against the optimum solution based on the average percentage error, standard deviation and the number of best solution obtained [4], [15], [20-21], [23]. This proposed heuristic was also compared with existing heuristic, NEH heuristic.

2. Methodology
The bottleneck-based heuristic firstly identifies the bottleneck stage and then determines the schedule of the jobs in the bottleneck stage [11]. The job processing times are used as indicator to determine the bottleneck stage. The procedures of the BNB heuristic are presented as below:

2.1 Bottleneck Machine Identification Procedure
Based on the four machines problem, there will be four possible dominant machines and each of the dominance will have their own initial sequences. The initial sequence for each of the machine dominance was arranged based on the result of machine dominant identification. Bottleneck identification is important to classify the criticality of process machines. The machine criticality or called as dominant machine is classified based on the processing times of job being process in each machines. 100 sets of processing times are generated randomly in VBA. All the steps to identify the machine bottleneck are explained below:
Step 1: The average processing time for all the six jobs on four machines was calculated as shown in Figure 1.

| JOB | M1  | M2  | M3  | M4  | TOTAL |
|-----|-----|-----|-----|-----|-------|
| A   | 13  | 28  | 11  | 36  | 88    |
| B   | 23  | 30  | 34  | 6   | 93    |
| C   | 11  | 28  | 7   | 32  | 78    |
| D   | 22  | 15  | 35  | 37  | 109   |
| E   | 34  | 34  | 28  | 39  | 135   |
| F   | 10  | 5   | 28  | 27  | 70    |

**Average Process** 25.150

**Figure 1.** Example of processing time data.

Step 2: The domination of machine was then determined using following equations:

\[
X = \begin{cases} 
1, & \text{if } \tau > \tau_{\text{avg}} \\
0, & \text{if } \tau \leq \tau_{\text{avg}} 
\end{cases} \tag{1.0}
\]

Where;

- \(X\) = Dominance value
- \(\tau\) = Processing time
- \(\tau_{\text{avg}}\) = Average processing time

Step 3: The total dominance value was calculated for each machine as shown in Figure 2. The machine which has the highest dominance value is identified as the bottleneck machine.

| Dominance calculation | Max. Dominance value | Dominance machine |
|-----------------------|----------------------|-------------------|
| M1 | M2 | M3 | M4 | 5 | M4 |
| 0  | 1  | 0  | 0  | 1  | 0  |
| 0  | 1  | 1  | 0  | 0  | 0  |
| 0  | 1  | 0  | 0  | 1  | 0  |
| 0  | 0  | 1  | 1  | 0  | 0  |
| 1  | 1  | 1  | 1  | 1  | 1  |

**Figure 2.** Example of dominance calculation.
2.2 Initial Sequences and Insertion Phase

In the study of Framinan and Rajendran [7], it was found that the initial job arrangement and the opportunity of inserting a job in any possible position are the strength of NEH heuristic which known as the best heuristic in solving the permutation flowshop problem with objective of makespan minimization. Thus, in this paper, the initial sequence was arranged in best arrangement based on the result of machine dominant identification so that the final solution will have the best answer.

- In cases of machine 1 dominant (M1), the processing time of jobs for machine 1 was used (M1).
- In cases of machine 2 dominant (M2), the processing time of jobs for machines 1 and 2 was sum up (M1+M2).
- In cases of machine 3 dominant (M3), the processing time of jobs for machines 3 and 4 was sum up (M3+M4).
- In cases of machine 4 dominant (M4), the processing time of jobs for machine 4 was used (M4).

From the observation, in building an initial sequence for all dominant machine, it was found that for M1 dominant, the machine criticality is depending on the processing time at machine 1. For M2 dominance, the machine criticalities are on the machine 1 and 2, thus the processing time for both machines was summed up. Next, for M3 dominance, the processing time for the second last and the last machine was considered while for M4 dominance, the processing time of last machine is the most critical. The sequence guide is the job with highest processing time must be processed first and then proceed with the job with second highest processing time and continue with the next job until the last job with lowest processing time. A job with highest processing time must be processed first so that there are not much idle time towards the end of the overall sequence.

The generalization sequences of this heuristic can be described as follows:

Input:
Suppose a standard permutation flowshop problem with $i = 1, \ldots, m$ machines, $j = 1, \ldots, n$ jobs, processing time $P_{ij}$ and the objective function is makespan minimization, $C_{max}$.

Step 1: Sort the $n$ jobs according to descending sums of processing time depending on the machine domination cases.

Step 2: The initial sequences arrangement are depending on the job sorting in Step 1. The first place was filled by a job with largest sum of processing time, $\text{Max } \Sigma P_{ij}$ followed by the second largest sum of processing time and so on until the last place, the sixth place was filled with the lowest sum of processing time, $\text{Min } \Sigma P_{ij}$.

Step 3: Then, take the first two jobs from the job arrangement in Step 2 and schedule their start-stop time. Reverse the position of jobs. Determine the first partial sequences with the lowest objective function, $C_{max}$.

Step 4: Next, for the $k$th job, $k = 3, \ldots, n$, insert the job into the place, among the position 1, 2, $\ldots$, $k$ job of partial sequences, while keeping the relative sequence of partial sequence. Choose the best sequences out of $k$ sequences as partial sequence for the next iteration.

3. Results and Discussion

This section discussed the result of the BNB heuristic performance through comparison with the optimum answer and also the NEH heuristic. A test on heuristic validation was also explained in this section. This paper performed the experiment using the problem size of four-machines and six-jobs. There are 4 potential machines dominance in this paper and the processing time are randomly generated using uniform distribution for 100 set of test problems with 10 replications.
3.1 Heuristic Performance
Table 1 summarizes the factors used to define the experimental condition: number of machines, number of jobs, number of bottleneck machine, variation of job processing time, number of test problems and number of replications. At the stage of bottleneck identification, the result shows that the machine dominance 1, M1= 35 data set, machine dominance 2, M2= 19 data set, machine dominance 3, M3= 27 data set, machine dominance 4, M4= 19 data set. The result was shown in Table 2.

| Experiment Condition | Levels |
|----------------------|--------|
| Number of machines   | 4      |
| Number of jobs       | 6      |
| Number of bottleneck machine | 4 |
| Variation of job processing time | [1,40] (low, high) |
| Number of test problems | 100 |
| Number of replications | 10 |

Table 2. Machine Domination.

| Machine Dominance | Number of data set |
|-------------------|--------------------|
| M1                | 35                 |
| M2                | 19                 |
| M3                | 27                 |
| M4                | 19                 |

In this paper, in order to evaluate the effectiveness of this proposed bottleneck-based (BNB) heuristic, the simulated result was compared with the optimum answer based on the average percentage error (Avg), standard deviation (Std) and the number of optimum solution obtained (Best). The optimum makespan was recorded from the complete 720 iterations. The error for each heuristic solution is defined as 100[(Heuristic - Best) / Best]. The average error is then computed for each dominant group and this is summarized in Table 3. The evaluations of BNB heuristic have been made against NEH heuristic. The different of BNB against NEH heuristic is defined as 100[(BNB – NEH) / NEH] and was classified based on the dominant group, the results was tabulated in Table 4.

| Machine Dominance | Avg error (%) | Std   | Best solution obtained | Optimum result (%) |
|-------------------|---------------|-------|------------------------|--------------------|
| M1                | 1.09          | 0.0196| 21                     | 60.00              |
| M2                | 1.13          | 0.0194| 12                     | 63.16              |
| M3                | 0.85          | 0.0190| 21                     | 77.78              |
| M4                | 0.32          | 0.0062| 13                     | 68.42              |
Table 4. Evaluation of BNB heuristic against NEH heuristic.

| Machine Dominance | BNB VS NEH | Average difference BNB/NEH (%) |
|-------------------|------------|-------------------------------|
|                   | WIN  | TIE  | LOSE |                   |
| M1                | 11   | 20   | 4    | -0.48             |
| M2                | 3    | 13   | 3    | -0.20             |
| M3                | 4    | 22   | 1    | -0.19             |
| M4                | 5    | 13   | 1    | -0.45             |

From the result Table 3, the average percentage errors and the standard deviation for BNB heuristic against the optimum makespan are quite small since the BNB completion times are very close to optimum result. It clearly showed that the number of best solutions obtained by BNB heuristic are impressive. Machine dominance 1 (M1) achieved 21 best solutions from 35 solutions, M2 achieved 12 best solutions from 19 solutions while M3 achieved 21 best solutions from 27 solutions and lastly M4 achieved 13 best solutions from 19 solutions. As expected, the optimum result showed that the proposed heuristic is performing well since it achieved 60% and more of optimum answer.

Table 4 shows the evaluation of performance of BNB heuristic against NEH heuristic. Since the BNB heuristic achieved good results against the optimum solution, it is expected that this heuristic can compete with existing heuristic, NEH heuristic. From the result of BNB heuristic against NEH heuristic, machine dominance M1 produced 11 better makespan (11 wins), 4 worse makespan (4 lose) and 20 equal makespan (20 tie) compared to the NEH. M2 produced 3 wins, 3 lose and 13 ties makespan result against the NEH. For M3 and M4, the results are as shown in Table 4. Overall, it was found that BNB heuristic have gained 67 number of optimum solutions compared to NEH heuristic which only gained 58 number of optimum solutions. The average different of BNB heuristic against NEH heuristic shows the negative value because the solution produce by BNB is better than the NEH solution for each of the machine dominance. So, the table clearly shows that BNB heuristic performs better than the NEH heuristic.

3.2 Heuristic Validation

In order to validate the heuristic performance, a test on 10 replications of 100 set of random data was carried out. The performance of both heuristic, BNB heuristic and NEH heuristic was measured based on the best solution obtained (Best) and also the average percentage error (Avg). The results are shown in Table 5.

Table 5. BNB heuristic against NEH heuristic performance.

| Replication Set of Random Data | Best solution obtained | Average Error (%) |
|-------------------------------|------------------------|-------------------|
|                               | BNB | NEH  | BNB VS OPT | NEH VS OPT |
| 1                             | 67  | 58   | 1.00       | 1.35        |
| 2                             | 64  | 62   | 1.02       | 0.77        |
| 3                             | 65  | 73   | 1.05       | 0.82        |
| 4                             | 65  | 64   | 0.98       | 0.93        |
| 5                             | 62  | 62   | 0.94       | 0.94        |
| 6                             | 61  | 68   | 0.95       | 0.88        |
| 7                             | 67  | 63   | 0.88       | 1.00        |
| 8                             | 61  | 60   | 1.13       | 0.96        |
| 9                             | 65  | 68   | 1.09       | 0.76        |
| 10                            | 67  | 64   | 0.98       | 0.89        |

From the result in Table 5, the best solution obtained by the BNB heuristic is much better than the NEH heuristic. BNB heuristic achieved more best solution in 6 set of replication, loss best solution
in 3 set of replication and tied in 1 set of replication. BNB heuristic produces an answer more than 60 best solution compared to the NEH. Even though the BNB heuristic obtained more best solution than NEH, however, from the overall result of average percentage error, the result shows that NEH heuristic perform a little better in producing a good solution close to optimal answer. From the observation, it was found that BNB heuristic still can be improved to produce an answer near to the optimal answer since the job arrangement are not effective to a certain sets of data where it resulted into large completion time.

4. Conclusion
This paper addresses the permutation flowshop scheduling problem with the objective of makespan minimization. Many researches and heuristics have been proposed to compete with the existing heuristic. This paper proposes a heuristic for a small size flowshop problem which the performance of the heuristic is depending on the quality of initial sequence. The problems set are first being classified into specific number of bottleneck-based machines. This paper applied the NEH insertion technique since NEH heuristic are the most famous heuristic in solving permutation problem objective function. From the results that have been shown in the previous section, bottleneck-based (BNB) achieved 60 % and more of optimum solution and heuristic validation result achieved more best solution of 6 sets out of 10 sets replication. Overall, it can be concluded that the proposed BNB heuristic performs better than the NEH heuristic.

Acknowledgement
This research is funded by Ministry of Higher Education of Malaysia (MOHE) and Universiti Tun Hussein Onn Malaysia (UTHM) under Fundamental Research Grant Scheme (FRGS, Vot 1497).

References
[1] Sule, D. R. 2008. Production Planning and Industrial Scheduling: Examples, Case Studies and Applications; Second Edition. United States of America: CRC Press, Taylor and Francis Group.
[2] Thomas, E. M. & David, W. P 1993. Heuristic Scheduling Systems. Canada: Wiley-Interscience.
[3] Johnson, S. 1954. Optimal two- and three stage production schedules with setup times included. Naval Research logistics Quarterly. 1:61.
[4] Allahverdi, A. and Aldowaisan, T. 2002. New heuristics to minimize total completion time in m-machine flowshops. International Journal Production Economics. 77: 71-83.
[5] Kalczynski, P. J. and Kamburowski, J. 2008. An improve NEH heuristic to minimize makespan in permutation flow shops. Computers & Operations Research. 35: 3001-3008.
[6] Kalczynski, P. J. and Kamburowski, J. 2007. On the NEH heuristic for minimizing the makespan in permutation flow shops. OMEGA, The International Journal of Management Science. 35: 53-60.
[7] Framinan, J. M., Leisten, R. & Rajendran C. 2003. Different intial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. International Journal of Production Research. 41(1), pp. 121-148.
[8] Modrak V. and Pandian R. S. 2010. Flow shop scheduling algorithm to minimize completion time for n-jobs m-machine problem. Technical Gazette. 17(3), pp. 273-278.
[9] Braune, R and Zapfel,G. 2015. Shifting bottleneck scheduling for total weighted tardiness minimization- A computational evaluation of subproblem and re-optimization heuristics. Computers & Operations Research. 66: 130-140.
[10] Zhenqiang, B., Weiye, W., Peng, W., and Pan, Q. 2012. Research on production scheduling system with bottleneck based on multi-agent. Physics Procedia. 24: 1903-1909.
[11] Chen, C. L. and Chen C. L. 2009. A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines. *Computers & Operations Research*. 36: 3073-3081.

[12] Zhang, R. and Wu, C. 2012. Bottleneck machine identification method based on constraint transformation for job shop scheduling with genetic algorithm. *Information Sciences*. 188: 236-252.

[13] Bareduan, S. A. and Hasan, S. 2009. Bottleneck-Based Heuristic For Flow Shop With Two Potential Dominant Machines. *Computers & Industrial Engineering*. 978: 4136-4244.

[14] Allahverdi, A. and Aldowaisan, T. 2002. New heuristics to minimize total completion time in \( m \)-machine flowshops. *International Journal Production Economics*. 77: 71-83.

[15] Aldowaisan, T. and Allahverdi, A. 2003. New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*. 30: 1219-1231.

[16] Kalczynski, P. J. and Kamburowski, J. 2005. A heuristic for minimizing the makespan in no-idle permutation flow shops. *Computers & Industrial Engineering*. 49: 146-154.

[17] Viagas, V. F. and Framinan, J. M. 2015. NEH-based heuristic for the permutation flowshop scheduling problem to minimise total tardiness. *Computers & Operations Research*. 60: 27-36.

[18] Abedinnia, H. Glock, C. H. and Brill, A. 2016. New simple constructive heuristic algorithms for minimizing total flow-time in the permutation flowshop scheduling problem. *Computers & Operations Research*. 74: 165-174.

[19] Framinan, J. M., Leisten, R. & Rajendran C. 2006. A heuristic for scheduling a permutation flowshop with makespan objective subject to maximum tardiness. 99: 28-40.

[20] Chakravarty, K. and Rajendran, C. 1999. A heuristic for scheduling in a flowshop with bicriteria of makespan and maximum tardiness minimization. *Production Planning and Control*. 10: 707-14.

[21] Aldowaisan, T. and Allahverdi, A. 2003. New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*. 30: 1219-1231.

[22] Allahverdi, A. and Al-Anzi, F.S. 2006. A branch-and-bound algorithm for three-machine flowshop scheduling problem to minimize total completion time with separate setup times. *European Journal of Operational Research*. 169: 767-780.

[23] Allahverdi, A. 2004. A new heuristic for 3-machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness. *Computers & Operations Research*. 31: 157-180.