Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), March 3 – 5, 2015

Numerical Assessment on Makespan Minimization by Adopting NEH Heuristics in Permutation Flow Shop

Chong Zheng Ziao
School of Mechanical Engineering
Engineering Campus, Universiti Sains Malaysia
Nibong Tebal, Pulau Pinang 14300, Malaysia
zz_circle@hotmail.com

Shahrul Kamaruddin
Department of Mechanical Engineering
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Perak Darul Ridzuan
Malaysia
shahrul.k@petronas.com.my

***Abstract*— Permutation flow shop problem (PFSP) is one of the important subclass of scheduling, where the order in each process remains unchanged for all machines. Few algorithms have been developed for deciding the sequence of n jobs and m machines to minimize makespan in the flow shop. Throughout the last 30 years, the NEH heuristic developed by Nawaz, Enscore and Ham has been commonly regarded as the best heuristic for minimizing the makespan in permutation flow shops. This heuristic is basically divided into two main stages: generate an initial sequence with decreasing total processing time and insert into the schedule accordingly. In this paper, total of 50 flow-shop problems were solved with the number of machines and jobs being set at a range of 4 to 25. Whereas the process times of the jobs were randomly generated with range 1 to 10 using Microsoft Excel spreadsheet. Performance of the heuristic was validated using the relative percentage deviation (RPD) formula based on the makespan generated by CDS heuristic proposed by Campbell, Dudek and Smith. The generated results through Excel modeling showed that NEH heuristic outperforms CDS heuristic. This statement correspond with the previous studies. Therefore, the Excel modelling is said to be an encouraging direction for solving scheduling problems in future research.**

***Keywords*—scheduling; NEH heuristic; permutation flow shop; makespan; idle time**

I. INTRODUCTION

Scheduling can be defined as a decision making process that concerned with the allocation of resources attached to the activities with the objectives of optimizing one or more performance measures. One of the active researched topics in scheduling literature is the permutation flow shop problem (PFSP) [1]. It refers to determination of the order of n jobs on m machines while all jobs have the same machine sequence. Among the desired objectives, minimization of makespan has gained the most attention [2]. This is due to, by reducing the makespan, maximizing utilization of machines can be achieved at the same time. Thus, this gives a big impact in fixed unit cost reduction.

The earliest known heuristic for the PFSP will be the Johnson's algorithm[3]. The proposed idea was restricted to two machines optimization problem. However, by grouping the m machines into two "virtual" machines, it can be used as a heuristic for the m machine case. The general idea of Johnson's rule is adopted by the other authors in their algorithms. One of the examples will be paper proposed by Dudek and Teuton [4]. The idea is based on developing a m-stage rule for the PFSP that minimizes the idle time accumulated on the last machine when processing each job by using Johnson's approach. In 1970, Campbell et al. [5] proposed a heuristic algorithm which is also an extension of Johnson's algorithm. In this case, several schedules are constructed and the best one is given as a result. The heuristic is known as CDS and built m - 1 schedule by clustering the m original machines into two virtual machines and solve the generated two machine problem by repeatedly using Johnson's rule. Since all the jobs in a PFSP form a permutation, many proposed methods work with the idea of exchanging the position of the jobs in the sequence or inserting jobs at different locations to obtain better results. The idea is to obtain a good ordering of the jobs and to subsequently improve this order by means of job exchanges. In the early of 80's, Nawaz et al. [6] has proposed a heuristic named NEH heuristic. It is based on the idea that jobs with high processing times on all the machines should be scheduled as early in the sequence as possible. Throughout the last 30 years, the NEH heuristic has been often considered as the leading heuristic for minimizing the makespan in permutation flow shops [7]. As the NEH is considered as one of the best heuristics for the PFSP, recently there lot of work can be found on the basis of NEH, focusing on extending or modifying the original algorithm, such as [8-10].

Generally, in part of metaheuristics that is commonly reported as the best results, the initial sequence is developed from NEH. For example, Reeves [11] stated that the effect of "seeding" the initial population often gave a good solution which is formulated by a constructive heuristics, and the best is the NEH algorithm is an accepted agreement. On the other hand, Solimanpur et al. [12] stated that in order to find an initial sequence, a constructive algorithm is adopted. NEH is selected among all the algorithm where Taillard [13] has come out with the conclusion from his experiment that NEH algorithm performs better than other constructive methods. Besides that, a NEH based heuristics has been proposed by Bhongade and Khodke [10]. It is an

978-1-4799-6065-1/15/$31.00 ©2015 IEEE

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.

idea of branch and bound heuristic combined with well-known NEH heuristics for flow shop problem. The procedure is divided in two phases. In the first phase, the sequence of parts for minimum makespan is decided for every individual product and in the second phase, the sequence of product is decided for minimum makespan. Besides the situation where the problem size is large, the heuristic in [8] basically gave a better solution. However, due to the disproportionately high computation time of the heuristic, it is not recommended for obtaining initial solution.

In the paper proposed by Kalczynski and Kamburowski [14] the authors discussed about the issue of a fair differentiation of the NEH results with those acquired by metaheuristics. In the end of the study they proved that NEH is still the best solution for the problem $F_m|\text{prmu}|C_{max}$. Base on the analysis carried out, the authors stated that the main features of NEH advantages are the initial arrangement of the jobs and the job insertion phase. On the other hand, the major weakness of NEH is the random tie breaking rule. Therefore in order to extend and prove that NEH is giving a numerical assessment on makespan minimization in a permutation flow shop will be the subject matter of this paper. It will be evaluated with another well accepted rule known as CDS.

The rest of the paper is organized as follows. In Section 2 the theory behind the well-known NEH and CDS heuristics are elaborated. Section 3 deals with numerical assessment prepared for NEH heuristics using Microsoft Excel 2010. Comparisons of data generated by NEH and CDS heuristics can be seen in Section 4. Finally, in Section 5 some conclusions are given.

II. THE HEURISTIC

Fundamentally, the flow-shop sequencing problem is focusing on the arranging the sequence of a list of *n* jobs to be processed on the *m* machines in the same order. The processing time of job *i* on machine *j* is usually written as $t_{i,j}$. Where *i* = 1, 2,…, *n; j* = 1, 2,…, *m.* While sequencing the jobs, minimization of the total time taken to complete the processes (makespan) will be the main concern. Based on the problem stated above, it is possible to have *n!* solutions. In another words, the minimum makespan could only be determined if all the *n!* sequences are listed out. Apparently, this procedure is expensive and impractical when the number of jobs gets larger.

A. NEH Heuristic

Following the NEH algorithm developed by Nawaz et al. [6], based on the assumption made, job with a longer total processing time is more preferable to be inserted into the schedule than the job with shorter processing time. In the other words, the two jobs with the highest total processing time are selected from the *n*-jobs. By considering the two possible partial schedules, the best partial sequence for the first two jobs is found. The relative positions of these two jobs with respect to each other with the minimum makespan are unchanged in the remaining steps of the algorithm throughout the process. The algorithm will continue to find the job with the third longest total processing time. This time, three partial sequences are generated by placing job at the beginning, middle and end of the partial sequence found in the previous step. The partial sequence with the shortest makepsan will be unchanged in the remaining steps as well. Similar steps will be continued until all the jobs have inserted into the schedule. The minimum number of enumerations of this algorithm can be calculated using the formula:

$$\frac{n\,(n+1)}{2} - 1 \qquad (1)$$

The procedural steps of the NEH insertions [6] are shown below:

Step 1 For each job *i* calculate

$$T_i = \sum_{j=1}^{m} t_{i,j} \qquad (2)$$

Where $t_{i,j}$ is the process time of job *i* on machine *j*.

Step 2 Arrange the jobs in descending order of T_i.

Step 3 Two jobs are selected from the first and second position of the list of Step 2, and find the best sequence for these two jobs by calculating makespan for the two possible sequences. Remain the relative positions of these two jobs with respect to each other in the remaining steps of the algorithm. Set *i* = 3.

Step 4 Select the job in the *i*th position in the list generated in Step 2 and determine the best sequence by locating it at all possible *i* positions in the partial sequence found in the previous step, without changing the relative positions to each other of the already assigned jobs, the number of the insertions at this step equals *i*.

Step 5 If *n* = *i*, STOP. Otherwise set *i* = *i* +1 and go to step 4

According to the heuristic, if ties exist in partial sequences, each of the tie sequence has to be examined. This causes the number of enumeration cannot be calculated using the formula given above. In this paper, whenever there is ties exist, only the first solution with minimum makespan is selected to continue with the remaining steps.

B. CDS Heuristic

CDS rule is transformed from the Johnson's Rule. *m*-1 of artificial two machines were generated from the *m* machines. Each of the problems is then solve using the Johnson's Rule. The simplified steps of CDS are shown:
First of all, calculate the processing time of the two machines.

$$\theta_{j1}^{k} = \sum_{i=1}^{k} t_{ji} \qquad (3)$$

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.

Equation (3) is the sum of processing time for the j^{th} job on machine 1 (M1).

$$\theta_{j2}^{k} = \sum_{i=m+1-k}^{m} t_{ji} \quad (4)$$

Equation (4) is the sum of processing time for the j^{th} job on machine 2 (M2)[5]. Apply Johnson's Rule to solve the two machines problem. In the end of the solution, select the minimum makespan sequence as the best sequence.

In order to evaluate the applicability of NEH and CDS for solving the makespan minimization in a permutation flow shop. The motivation of using CDS in the comparison is due to the study of Setiaputra [15], the author stated that CDS is performing better than the other five well known heuristics. The comparison is done through a numerical assessment approach that is decoded into a simple Excel model.

III. Numerical Assessment

A total of 50 flow shop problems were solved with the number of machines and jobs being set at a range of 4 to 25. 10 sets of data were created for each category. This means, the problems with the same number of jobs and machines will be classified under the same category. In another words, there are 5 categories in this numerical assessment. The classifications of categories are tabulated in TABLE 1. In Category I, the problems involved 4 jobs with 5 machines. In Category II, there are 5 jobs with 4 machines, whereas Category III have 5 jobs with 10 machines. Category IV have 10 jobs with 5 machines and finally Category V have 25 jobs with 25 machines.

TABLE 1

Category	Number of Job	Number of Machine
I	4	5
II	5	4
III	5	10
IV	10	5
V	25	25

The processing times of the jobs were randomly generated with range 1 to 10 using the Excel function. An example can be seen in TABLE 2. This "RANDBETWEEN(TOP, BOTTOM)" function returns a random integer number between the numbers specified, which are "1" and "10". A new random integer number is returned every time the worksheet is calculated. The modeling of Excel worksheets were based on the NEH algorithm operating steps. Therefore, whenever there is new data inserted, the worksheet is able to generate the sequence and the makespan by adopting NEH heuristic. In this section, the makespan generated in Excel worksheet following the NEH heuristic is compared with the makespan generated following the CDS heuristic by using the same sets of processing time. The working steps of CDS heuristic are not going to show in this paper. Refer to [5] for the similar calculation steps.

Step 1: As shown in TABLE 2, once the process time of each job are inserted into the grey boxes, the total processing time, T_i will be calculated in the last column.

Step 2: The particular column is then sorted with "largest to smallest" sequence in order to have a descending order of T_i.

Step 3: According to the example given, the two jobs from the first and second position in the list, which are "Job 1" and "Job 3" were selected for calculating the partial makespan as in TABLE 3. As sequence 1 – 3 gives a shorter makespan, therefore this sequence will be remain and bring forward to the remaining steps.

TABLE 2

Machine, j / Job, i	A	B	C	D	E	Total Processing Time, T_i
1	8	6	8	9	9	40
3	5	8	10	10	4	37
4	10	6	10	9	1	36
2	4	3	8	7	9	31

TABLE 3

Job	Time	A	B	C	D	E	Makespan
1	Process	8	6	8	9	9	46
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
3	Process	5	8	10	10	4	
	In	8	14	22	32	42	
	Out	13	22	32	42	46	
3	Process	5	8	10	10	4	51
	In	0	5	13	23	33	
	Out	5	13	23	33	37	
1	Process	8	6	8	9	9	
	In	5	13	23	33	42	
	Out	13	19	31	42	51	

Step 4: The job located in the third place in TABLE 2, which is "Job 4" is inserted into front, middle and back of the sequence 1 – 3. Same as the previous step, the sequence with the shortest makespan 52 is chosen. The sequence is 1 – 3 – 4. This is shown in TABLE 4.

Step 5: The sequence 1 – 3 – 4 is fixed and the job in the fourth place in TABLE 2, which is "Job 2" is inserted in to 5 different locations as shown in TABLE 5. This step will continue to look for the minimum makespan as the previous steps. The first enumeration 2 – 1 – 3 – 4 with the makespan of 56 is chosen.

Step 6: Now $i = n$, all the jobs are inserted into the schedule. Therefore the calculations will stop here with the final outcome of makespan = 56. The sequence is 2 – 1 – 3 – 4.

The total 9 enumerations are all generated in a single Excel worksheet. Whenever there is any change in numerical input, the same procedural steps will be taken and calculate the final makespan. However, if ties exist, only the first solution that gave us a minimum makespan is selected instead of examining each tied sequence. With the functions inserted in

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.

each cell, the worksheet will follow the rules of NEH heuristic.

TABLE 4

Job	Time	A	B	C	D	E	Makespan
4	Process	10	6	10	9	1	58
	In	0	10	16	26	35	
	Out	10	16	26	35	36	
1	Process	8	6	8	9	9	
	In	10	18	26	35	44	
	Out	18	24	34	44	53	
3	Process	5	8	10	10	4	
	In	18	24	34	44	54	
	Out	23	32	44	54	58	
1	Process	8	6	8	9	9	58
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
4	Process	10	6	10	9	1	
	In	8	18	24	34	43	
	Out	18	24	34	43	44	
3	Process	5	8	10	10	4	
	In	18	24	34	44	54	
	Out	23	32	44	54	58	
1	Process	8	6	8	9	9	52
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
3	Process	5	8	10	10	4	
	In	8	14	22	32	42	
	Out	13	22	32	42	46	
4	Process	10	6	10	9	1	
	In	13	23	32	42	51	
	Out	23	29	42	51	52	

TABLE 5

Job	Time	A	B	C	D	E	Makespan
2	Process	4	3	8	7	9	56
	In	0	4	7	15	22	
	Out	4	7	15	22	31	
1	Process	8	6	8	9	9	
	In	4	12	18	26	35	
	Out	12	18	26	35	44	
3	Process	5	8	10	10	4	
	In	12	18	26	36	46	
	Out	17	26	36	46	50	
4	Process	10	6	10	9	1	
	In	17	27	36	46	55	
	Out	27	33	46	55	56	
1	Process	8	6	8	9	9	60
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
2	Process	4	3	8	7	9	
	In	8	14	22	31	40	
	Out	12	17	30	38	49	
3	Process	5	8	10	10	4	
	In	12	17	30	40	50	
	Out	17	25	40	50	54	
4	Process	10	6	10	9	1	
	In	17	27	40	50	59	
	Out	27	33	50	59	60	
1	Process	8	6	8	9	9	60
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
3	Process	5	8	10	10	4	
	In	8	14	22	32	42	
	Out	13	22	32	42	46	
2	Process	4	3	8	7	9	
	In	13	22	32	42	49	
	Out	17	25	40	49	58	
4	Process	10	6	10	9	1	
	In	17	27	40	50	59	
	Out	27	33	50	59	60	
1	Process	8	6	8	9	9	60
	In	0	8	14	22	31	
	Out	8	14	22	31	40	
3	Process	5	8	10	10	4	
	In	8	14	22	32	42	
	Out	13	22	32	42	46	
4	Process	10	6	10	9	1	
	In	13	23	32	42	51	
	Out	23	29	42	51	52	
2	Process	4	3	8	7	9	
	In	23	29	42	51	58	
	Out	27	32	50	58	67	

IV. RESULTS

The performance of the NEH heuristic was tested on randomly generated integers from 1 to 10 using Excel worksheet as the processing times. This is important in validating the capability of Excel worksheet in solving the scheduling problems. There are 5 categories (m,n) and N= 10 problems in each category. Extracting the idea from Campbell et al. [5], Taillard [13] and Woo and Yim [16], the processing time were drawn randomly. Modification made in the case of instead of taking the integers from 1 to 99, the processing time between the ranges of 1 to 10 is used. The makespan obtained by adopting NEH and CDS heuristics in Category I where n = 4 and m = 5 is shown in TABLE 6.

TABLE 6

Job x Machine : 4 x 5				
Test	**Makespan**			
	CDS	**NEH**	**RPD**	**Condition**
1	56	56	0.00	Tie
2	53	54	-1.89	Lose
3	42	42	0.00	Tie
4	42	42	0.00	Tie
5	43	43	0.00	Tie
6	46	46	0.00	Tie
7	51	51	0.00	Tie
8	41	41	0.00	Tie
9	53	49	7.55	Win
10	58	58	0.00	Tie
ARPD			**0.57**	**Win**

The effectiveness of the NEH heuristics was validated by comparing the makespan with those obtained by applying the CDS algorithm. For each instances, a computation approach to validate and prove the effectiveness is carried out. If the outcome of this validation is correspond with the previous studies, this prove that the Excel modelling are working well. This test bed that used in this validation was introduced by [16]. The formula of Relative Percentage Deviation (RPD) used for the basis is shown in (5), where $NEH_{makespan}$ is the solution obtained for the instance by the NEH heuristic while $CDS_{makepsan}$ is the solution obtained for the instance by CDS heuristic. The Average Relative Percentage Deviation (ARPD) values were calculated by averaging the RPD for each instance size.

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.

$$RPD = \frac{NEH_{makespan} - CDS_{makepsan}}{CDS_{makespan}} \quad (5)$$

The values of ARPD indicate how well the NEH heuristics is performing. Usually ARPD values falls into these three ranges as shown in TABLE 7.

TABLE 7

Range	Condition	Definition
Positive integers (APRD > 0)	WIN	NEH is performing **better than** CDS
Zero (ARPD = 0)	TIE	NEH is performing **as good as** CDS
Negative integers (ARPD < 0)	LOSE	NEH is performing **worse than** CDS

TABLE 8 shows the comparison of ARPD computed by adopting NEH heuristics with CDS heuristic. From the results shown, the implementation of heuristics in Excel worksheet is able to fulfill the statements of NEH author [6]. The statements are as below:

1. NEH does not perform better than CDS when number of job, n is small.
2. NEH does not outperform CDS when number of jobs, n is greatly smaller than number of machine, m.

These two statements are proved by Category I and Category III respectively. Category I was built up by 4 jobs and 5 machines, this represent example with small number of jobs. Whereas Category III is built up by 5 jobs with 10 machines, this is an example problem that the number of job is greatly smaller than number of machine. The ARPD calculated are 0.57 and 0.04. Although NEH prevail compare to CDS in these two categories, the small values of ARPD indicated that it is just slightly performing better as compared to the other categories that are giving remarkable better results. As these results are in line with the statement found in earlier investigates, Excel modelling is a worth exploring path for scheduling in the upcoming study.

TABLE 8

No. of Job, *n*	No. of Machine, *m*	No. of Win	No. of Tie	No. of Lose	ARPD	Condition
4	5	1	8	1	0.57	Win
5	5	4	4	2	2.22	Win
5	10	3	4	3	0.04	Win
10	5	10	0	0	2.80	Win
25	25	10	0	0	5.22	Win

V. CONCLUSION

In this paper, numerical assessment is designed for a total of 50 flow-shop problems with the number of machines and jobs being set at a range of 4 to 25. Modification is made in the processing times of the jobs with range of 1 to 10 using excel spreadsheet instead of 1 to 99. Performance of the heuristic was computed using the relative percentage deviation (RPD) formula based on the makespan generated by CDS heuristic proposed by Campbell, Dudek and Smith. The generated results through Excel modeling showed that NEH heuristic outperforms CDS heuristic. This proved that the Excel modeling is able to obey procedural rules in NEH heuristic. The numerical assessment is successful.

There is very limited research on Excel modeling for scheduling. As the numerical assessment gives a positive outcome and studies must take into consideration with practical and economical application of the solution technique instead of only obtaining the optimal solution. Therefore, additional research is needed on Excel modeling for solving scheduling problems. Microsoft Excel is a common data sharing tools in industrial field. It is believed that by inserting proper functions in selected cells, Excel modelling is foreseen as a capable tool to solve scheduling problems that concern with time prediction.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the CREST grant scheme from the Government of Malaysia for funding this research. Also the authors are sincerely grateful to the anonymous referees who provide very valuable comments in improving this paper.

REFERENCES

1. Ruiz, R. and T. Stützle, *A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem.* European Journal of Operational Research, 2007. 177(3): p. 2033-2049.
2. Ruiz, R. and C. Maroto, *A comprehensive review and evaluation of permutation flowshop heuristics.* European Journal of Operational Research, 2005. 165(2): p. 479-494.
3. Johnson, S.M., *Optimal two- and three-stage production schedules with setup times included.* Naval Research Logistics Quarterly, 1954. 1(1): p. 61-68.
4. Dudek, R.A. and O.F. Teuton Jr, *Development of m-stage decision rule for scheduling n jobs through m machines.* Operations Research, 1964. 12(3): p. 471-497.
5. Campbell, H.G., R.A. Dudek, and M.L. Smith, *A heuristic algorithm for the n job, m machine sequencing problem.* Management science, 1970. 16(10): p. B-630-B-637.
6. Nawaz, M., E.E. Enscore, and I. Ham, *A heuristic algorithm for the< i> m</i>-machine,< i> n</i>-job flow-shop sequencing problem.* Omega, 1983. 11(1): p. 91-95.
7. Kalczynski, P.J. and J. Kamburowski, *An improved NEH heuristic to minimize makespan in permutation flow shops.* Computers & Operations Research, 2008. 35(9): p. 3001-3008.

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.

8. Jena, S., M. Poggi, and D. Sotelo, *Competitive Deterministic Heuristics for Permutation Flow Shop Scheduling*. 2009.
9. Dong, X., H. Huang, and P. Chen, *An improved NEH-based heuristic for the permutation flowshop problem.* Computers & Operations Research, 2008. 35(12): p. 3962-3968.

10. Bhongade, A. and P. Khodke, *Heuristics for production scheduling problem with machining and assembly operations.* International Journal of Industrial Engineering Computations, 2012. **3**(2).
11. Reeves, C.R., *A genetic algorithm for flowshop sequencing.* Computers & Operations Research, 1995. **22**(1): p. 5-13.
12. Solimanpur, M., P. Vrat, and R. Shankar, *A neuro-tabu search heuristic for the flow shop scheduling problem.* Computers & Operations Research, 2004. **31**(13): p. 2151-2164.
13. Taillard, E., *Some efficient heuristic methods for the flow shop sequencing problem.* European journal of Operational research, 1990. **47**(1): p. 65-74.
14. Kalczynski, P.J. and J. Kamburowski, *On the NEH heuristic for minimizing the makespan in permutation flow shops.* Omega, 2007. **35**(1): p. 53-60.
15. Setiaputra, W., *A Survey of Flow-shop Permutation Scheduling Techniques and Evaluation of Heuristic Solution Methods*. 1980, Pennsylvania State University.
16. Woo, H.-S. and D.-S. Yim, *A heuristic algorithm for mean flowtime objective in flowshop scheduling.* Computers & Operations Research, 1998. **25**(3): p. 175-182.

BIOGRAPHY

Chong Zheng Ziao is a MSc research student at the School Mechanical Engineering, Universiti Sains Malaysia. She received the B.Eng.(Hons) degree from Universiti Sains Malaysia in 2013.

Shahrul Kamaruddin is an Associate Professor at the Mechanical Engineering Department, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Perak Darul Ridzuan, Malaysia. He received the B.Eng.(Hons) degree from University of Strathclyde, Glasgow, Scotland in 1996, the M.Sc. degree from University of Birmingham, U.K., in 1998, and the PhD from University of Birmingham, in 2003. Prior to the Universiti Teknologi PETRONAS, he is with the School Mechanical Engineering (under the manufacturing engineering with management programme),Universiti Sains Malaysia as an Associate Professor. He also has various past experiences with manufacturing industries from heavy to electronics industries especially in the field of industrial engineering, manufacturing processes and product design. He has more than 150 publications in reputed international and national journals/conferences. His current research interests include simulation and modelling of manufacturing systems, production planning and control, maintenance engineering and management and manufacturing process related to the oil and gas industries.

Authorized licensed use limited to: UNIVERSIDADE DE SAO PAULO. Downloaded on August 09,2021 at 22:36:22 UTC from IEEE Xplore. Restrictions apply.