Development of Flow Shop Scheduling Method to Minimize Makespan Based on Nawaz Enscore Ham (NEH) & Campbell Dudek and Smith (CDS) Method

Latief Anggar Kurniawan, Farizal F
Department of Industrial Engineering, Faculty of Engineering
Universitas Indonesia
Jakarta, Indonesia
latief.anggar@ui.ac.id, farizal@eng.ui.ac.id

Abstract

Every company engaged in the manufacturing industry must implement a productivity improvement program. In avoiding the occurrence of queues in the manufacturing process, an overall solution is needed. One that is often used in scheduling systems is the flow shop pattern. FCFS are the most widely used methods, this method schedules work without paying attention to makespan, so that with this method there are often delays and inefficient production costs. According to Utami (2020) the CDS method produces a better solution than the FCFS but Jin et. Al (2007) state that the NEH method produces a more optimal solution than the CDS method, but the NEH method provides a longer computational time. The purpose of this study was to develop a new method for solving flow shop scheduling problems based on the concept of the CDS and NEH methods namely NEHLPD, NEHLPD1, NEHLPD2 and then tested them on random cases. Based on the results of running on 30 cases, it was found that: NEHLPD produces more efficient makespan than the CDS method and the NEH method where the average efficiency level is 3.56% and 0.28%, respectively; NEHLPD1 produces 9.67% faster computational time than the CDS method and 74.72% than the NEH method; NEHLPD2 produces a processing time of 12.41% faster than the NEH method and produces an average makespan solution of 1.84% better than the CDS method and only 1.49% worse than the makespan of NEH method.

Keywords
Flow shop scheduling, CDS, NEH, New method development, Random case.

1. Introduction

Every company engaged in the manufacturing industry must implement a productivity improvement program. In carrying out its activities, the company must have a good manufacturing process (Utami et al., 2020). The increasing use of operations research in various fields of science is a result of industrial progress. In avoiding the occurrence of queues in the manufacturing process, an overall solution is needed. One that is often used in scheduling systems is the flow shop pattern (Gozali et al., 2019)

Flow shop is the process of determining the sequence of jobs that have the same product path. The flow shop model is a job that is considered as a collection of operations where a special precedence structure is applied. The arrangement of a flow shop type production process can be applied precisely to products with stable designs and produced in large volumes, so that special-purpose investments used are quickly returned (Pinedo, 2001).

The Campbell Dudeck and Smith (CDS) method is the most important heuristic method for the makespan problem or the time to produce all jobs to completion in the flow shop scheduling problem (Campbell et al., 1970). According to Baker (1974) the Campbell Dudeck and Smith (CDS) method has advantages in two respects: The use of Johnson's rules is used to find job sequences involving 2 groups of machines as a processing tool for incoming jobs. Jobs that are processed must go through two groups of machines, the M1 machine and continue on the M2 machine until it is finished. CDS usually produces several sequences that can be selected as the best solution.

According to Utami (2020) the CDS method produces a better solution than the FCFS, Palmer and Dannenbring methods but the CDS method has weaknesses:

a. The CDS method uses the number of machines as the iteration limit so that if a few machines are used but there are many jobs to be done, the CDS iteration does not produce an optimal solution,
b. The CDS method uses processing time as a reference in scheduling and ignores the makespan whereas, the optimal scheduling is determined from the total makespan required to complete all jobs.
c. Sometimes it produces the exact same sequence of jobs as the previous iteration.

Nawaz Enscore Ham (NEH) is a heuristic method developed by Nawas and colleagues (1983). This method produces solutions in a different order (Nawaz et al., 1983). From the research results, the NEH method produces a more optimal solution than the CDS method, but the NEH method provides a longer computational time (Jin et al., 2007).

Over time, the NEH method has undergone several developments. NEHNM sorts the difference between the total processing time and the lower bound of delay time (Nagano & Moccellin, 2002), MNEH sorts the amount of artificial processing time in descending order (Low et al., 2004), NEHKK the sequence is selected with the least maximum completion time of the sequence between two positions, NEHKK1 and NEHKK2 uses an unincreasing number of weighted processing times (Kalczynski & Kamburowski, 2008), NEH - D sorts the mean and standard deviation of processing time in descending order (Dong et al., 2008), NEHFF selects the order with the least delay and idle time (Fernandez-Viagas & Framinan, 2014), NEHLJP1 combining the new priority rule and tie-breaking rule (Liu et al., 2017)

Jin et al., (2007) developed the NEH method to minimize computation time but ignore the makespan in each iteration. In the process of finding a solution using the CDS method, if there is the same processing time, then we can choose the order arbitrarily (Campbell et al., 1970) so that the less the difference in processing time the easier it is to sort, but no one has developed the NEH method using total time difference for each process. Therefore, it is necessary to develop a new method by combining the NEH and CDS methods to reduce computation time and still pay attention to makespan.

1.1 Objectives

This study aims to develop and test a new flow shop scheduling method which is carried out on random cases to find a better solution (smaller makespan) and faster computation time. In addition, this research is also to find out the level of efficiency of the resulting new method compared to the existing methods, namely CDS and NEH.

2. Literature Review

Scheduling theory is largely concerned with mathematical models that apply to the scheduling process. The ongoing interaction between theory and practice has been the creation of usable models, which leads to solution methodologies and practical insights. The theoretical approach is likewise essentially quantitative, attempting to represent issue structure in mathematical form. This quantitative method, in particular, begins with a description of resources and tasks, as well as the translation of decision-making goals into an explicit objective function. Ideally, the goal function should include all expenses that are affected by scheduling decisions. In reality, however, such expenses are sometimes difficult to quantify, if not totally define. The planning determines the principal running costs, as well as the most easily recognized ones (Baker & Trietsch, 2009).

The CDS algorithm was developed since the 1970s and Campbell Dudek Smith (CDS) tried their algorithm and tested its performance on various problems, and found that this algorithm is effective for small problems or large problems. The steps for scheduling the CDS algorithm are (Alharkan, 2019):

1. Take the first and last work stations or machines (other machines are considered non-existent), and then arrange the scheduling order using Johnson's rule.
2. Take the work station or machine 1, 2 and the last work station or machine (m), m-1 then add up the processing time between machines 1, 2 and m, m-1 and then arrange the scheduling sequence with Johnson's rule.
3. Take work stations or machines 1, 2, 3 and work stations or machines m, m-1, m-2 add up the processing time between machines 1, 2, 3 and m, m-1, m-2 and then arrange the scheduling sequence by Johnson's rule.
4. Repeat Step 3 until machine 1,2,3…m-1

5. For each resulting schedule, calculate the total completion time and then select the scheduling sequence with the smallest total completion time

The Nawaz Enscore Ham (NEH) method was developed by Muhammad Nawaz, E. Emory Enscore Jr., and Inyong Ham in 1983. The Nawaz Enscore Ham method is an incremental construction algorithm that has been awarded as the best heuristic method in the flow shop problem (Taillard, 1993). The NEH method assumes that the job that has a larger total processing time for all machines must take precedence over the job with a smaller total processing time. NEH initializes the order of jobs in descending order based on the total processing time of each job. Then the partial sequence process is carried out, which is to determine the best order of each possible job position. NEH Steps (Alharkan, 2019):

1. Add up all processing time for each job.
2. Sort jobs by number of processing times in descending order.
3. Choose the first two jobs from the list and create two partial sequences by swapping the positions of the two tasks. Compute the partial sequences' makespan values. The partial sequence with the shortest makespan is referred to as the incumbent sequence.
4. Select the next job from the work content list and assign it to all positions in the incumbent sequence. Determine the value of makespan for each sequence.
5. Keep the sequence with the shortest makespan as the incumbent sequence and reject all other sequences.
6. STOP if there are no jobs remaining in the work content list to be added to the incumbent sequence. Otherwise, go to step (4).

3. Methods

This research is focused on developing methods using a combination of cds and neh methods and performing computations on these methods. Computing is done using Macros Visual Basic for Applications on Ms. Excel 2013, laptop with windows 11, intel core i5-1135G7, 8GB RAM, VGA Intel Iris Xe Graphics. The steps in this research are as follows:

Step 1 : Conceptualize the developed method

Step 2 : Computerize the CDS, NEH and development methods, this method is needed in order to calculate the computational time for each method using the same laptop specification.

Step 3 : Verify and validate, verification is carried out to ensure that the coding in the computer is appropriate and validation is carried out using cases that are done manually compared to the computational results.

Step 4 : Testing methods and comparing methods using hypothetical data as many as 30 cases, this test is intended to get a solution based on each method and calculate the computation time of each method in solving the case.

Step 5 : Make improvements to the developed method

3.1 NEHLPD

NEHLPD (Least Process Deviation) is the first development method where this method combines the concept of the CDS method with the same process time and combines it with the concept of the NEH method. The steps of this method are as follows:

Step 1: Add up the total runtime for each job and sort the jobs from smallest to largest runtime, $y_i = \sum_{j=1}^{m} t_{ij}$.

Step 2: Calculate the time difference of each process (absolute value) for jobs in sequence 1 and 2, 2 and 3, 3 and 4 and so on.

Step 3: Add up the difference in processing time for each job, $S_i = \sum_{j=1}^{m}\{abs\ (t_{J_i j} - t_{J_{i+1} j})\}$.

Step 4: Iteration 1 (K=1), select jobs in $Min\ S_i$ as initial iteration, sort the jobs so that the smallest partial makespan is obtained then take the jobs order with the smallest partial makespan as partial solution.

Step 5: Compare the K values, if K=(n-1) then the iteration stops with a partial solution as the optimum solution, if K<(n-1) then the iteration continues with K=(K+1).

Step 6: Next iteration (K= (K+1)), select a job in the sequence immediately below or above the K-1 jobs with the smallest S value and place them in any order without changing the position of the K-1 jobs until partial smallest makespan then take the order of jobs with the smallest partial makespan as the partial solution.

Step 7: Repeat steps 5 and 6 until K= (n-1).

Where:

K	= iteration
y	= total processing time of a job
t	= processing time
J	= job order
S	= total absolute difference in processing time
i	= number of jobs (i=1…n)
j	= number of machines (j=1…m)

3.2 NEHLPD1
NEHLPD1 is the first improvement from NEHLPD to speed up computing time. In NEHLPD1, jobs are grouped into 2 jobs from the value of S to the largest and then sort 2 jobs in each group to produce the smallest makespan. It uses the assumption of the CDS concept where the more identical a process is, the easier it will be to sort (Campbell et al., 1970). The steps of this method are as follows:
Step 1: Add up the total runtime for each job and sort the jobs from smallest to largest runtime, $y_i = \sum_{j=1}^{m} t_{ij}$.

Step 2: Calculate the time difference of each process (absolute value) for jobs in sequence 1 and 2, 2 and 3, 3 and 4 and so on.

Step 3: Add up the difference in processing time for each job, $S_i = \sum_{j=1}^{m}\{abs\ (t_{J_i j} - t_{J_{i+1} j})$

Step 4: sort jobs by value S from smallest to largest, then group every 2 adjacent jobs. If the number of jobs is odd then leave 1 last job. $C_i = (J_{ix2-1}, J_{ix2})$

Step 5: Find the sequence of jobs with the smallest makespan value for each C.

Step 6: define the sequence of jobs in Step 5 as the optimal solution for each C.

Step 7: iteration 1 (K=1) select C_1 and C_2 from Step 6, then sort C until the smallest makespan is obtained as a partial solution.

Step 8: Compare the K values, if K ≥ $(\frac{n}{2} - 1)$ then the iteration stops with a partial solution as the optimum solution, if K < $(\frac{n}{2} - 1)$ then the iteration continues with K= (K+1).

Step 9: Select C_3 or C_{K+1} then place them in any order without changing the position of jobs K-1 until the smallest partial makespan is obtained then take the order of jobs with the smallest partial makespan as partial solution.

Step 10: Repeat steps 8 and 9 until K ≥ $(\frac{n}{2} - 1)$.

Where:

C	= group of jobs

3.3 NEHLPD2
NEHLPD2 is the second improvement from NEHLPD to optimize the makespan of the resulting solution. NEHLPD2 uses steps similar to NEHLPD1 except that in NEHLPD2 jobs can be placed anywhere without changing the job position in each C sequence. The steps of this method are as follows:
Step 1: Add up the total runtime for each job and sort the jobs from smallest to largest runtime, $y_i = \sum_{j=1}^{m} t_{ij}$.

Step 2: Calculate the time difference of each process (absolute value) for jobs in sequence 1 and 2, 2 and 3, 3 and 4 and so on.

Step 3: Add up the difference in processing time for each job, $S_i = \sum_{j=1}^{m}\{abs\ (t_{J_i j} - t_{J_{i+1} j})$

Step 4: sort jobs by value S from smallest to largest, then group every 2 adjacent jobs. If the number of jobs is odd then leave 1 last job. $C_i = (J_{ix2-1}, J_{ix2})$

Step 5: Find the sequence of jobs with the smallest makespan value for each C.

Step 6: define the sequence of jobs in Step 5 as the optimal solution for each C.

Step 7: iteration 1 (K=1) select C_1 and C_2 from Step 6, then place jobs in any position without changing the position of internal order of C until the smallest makespan is obtained as a partial solution.

Step 8: Compare the K values, if K ≥ $(\frac{n}{2} - 1)$ then the iteration stops with a partial solution as the optimum solution, if K < $(\frac{n}{2} - 1)$ then the iteration continues with K= (K+1).

Step 9: Select C_3 or C_{K+1} then place jobs on the C_{K+1} without changing the position of internal order of C_{K+1} in any order without changing the position of jobs K-1 until the smallest partial makespan is obtained then take the order of jobs with the smallest partial makespan as partial solution.

Step 10: Repeat steps 8 and 9 until K $\geq \left(\frac{n}{2} - 1\right)$.

4. Results and Discussion

Based on testing of 30 cases with hypothetical data, the results are as shown in table 1. Table 1 shows the total makespan and computational time to produce a solution in each case using the CDS, NEH and methods being developed. From table 1, hypothesis testing is carried out using the 95% confident level and get the results as shown in table 2. Based on table 2, the method being developed, namely NEHLPD produces a significantly different makespan solution from the CDS method with an average efficiency of 3.56%. but does not provide a different solution with the NEH method and only produces an average efficiency of 0.28%. Meanwhile, the computational time for the NEHLPD method was significantly different from the CDS and NEH methods, which slowed down by 263.68% compared to the CDS method and 1.77% compared to the NEH method. Because the NEHLPD method is still bad in terms of computational time, improvements are made and produce the NEHLPD1 method.

The results of testing the same case for the NEHLPD1 method are shown in table 1. While the results of hypothesis testing for the NEHLPD1 method are shown in table 3. Based on table 3, the NEHLPD1 method does not produce a significantly different makespan solution from the CDS method with an average efficiency of only -0.24 % but produces a different makespan solution with the NEH method with an average efficiency of -3.65%. Meanwhile, the computational time for the NEHLPD1 method was significantly different from the CDS method and the NEH method with the computational time efficiency of 9.67% and 74.72%, respectively. Because the NEHLPD1 method only produces better computational time, it is re-improved and produces the NEHLPD2 method.

Table 4 shows the results of hypothesis testing using the NEHLPD2 method against the CDS and NEH methods. Based on the results of makespan, the NEHLPD2 method is significantly different from the two methods where the NEHLPD2 method produces a solution with an average efficiency of 1.84% compared to the CDS method and 1.49% worse than the NEH method. Based on the computational time, the NEHLPD2 method is also significantly different from the CDS method and the NEH method where compared with the CDS method produces a longer computational time of 213% but produces a faster computation time than the NEH method of 12.41%.

Table 1. Method Testing Results

CASE	JOB	MACHINE	MAKESPAN					TIME				
			CDS	NEH	NEHLPD	NEHLPD1	NEHLPD2	CDS	NEH	NEHLPD	NEHLPD1	NEHLPD2
1	4	4	86	86	86	86	86	4	4	4	1	3
2	5	3	80	80	80	83	83	3	8	7	3	10
3	10	5	276	279	276	282	279	7	39	41	10	33
4	5	8	257	257	257	257	257	13	7	8	3	8
5	10	15	452	437	437	452	443	31	40	41	12	31
6	12	8	380	379	374	385	378	16	64	68	19	54
7	14	10	369	364	364	382	371	24	101	102	25	80
8	14	14	587	572	568	587	567	37	99	103	24	82
9	19	15	479	463	464	470	470	48	236	237	64	222
10	21	17	709	674	670	703	680	63	305	315	86	298
11	18	9	335	308	309	337	329	27	208	208	47	164
12	15	7	542	534	538	558	535	19	122	126	34	119
13	15	24	725	708	714	742	728	99	129	140	35	115
14	25	24	1331	1287	1285	1353	1299	133	537	563	135	462
15	34	29	1218	1157	1146	1243	1184	168	1188	1182	269	960
16	18	25	926	904	905	945	918	77	204	197	44	165
17	21	15	645	611	604	631	607	44	300	302	83	286
18	20	35	1033	999	999	1015	1009	114	258	268	67	215
19	16	23	452	448	446	459	447	65	146	146	33	120
20	11	20	496	495	495	516	499	47	51	55	16	50
21	21	38	794	778	780	797	789	136	307	302	79	281
22	29	41	1561	1490	1465	1518	1518	186	721	737	187	664
23	23	16	906	868	873	930	921	58	390	387	104	357
24	18	8	510	481	483	496	489	22	193	201	47	160
25	11	14	236	238	238	246	234	35	52	55	16	51
26	24	18	816	760	739	788	778	72	428	441	102	357
27	16	30	395	380	389	399	394	105	140	147	32	117
28	17	23	1245	1218	1218	1240	1234	81	172	172	45	161
29	15	25	504	483	485	486	484	88	119	124	35	115
30	13	13	522	509	509	527	509	39	82	89	24	85

Table 2. Hypothesis Testing for The NEHLPD Method

	MAKESPAN				*TIME*			
	NEHLPD	*CDS*	*NEHLPD*	*NEH*	*NEHLPD*	*CDS*	*NEHLPD*	*NEH*
Mean	606.5333333	628.9	606.5333333	608.2333	225.6	62.03333	225.6	221.6667
Variance	121955.3609	135583.6103	121955.3609	124050.5	61145.21	2363.206	61145.21	60620.02
Observations	30	30	30	30	30	30	30	30
Pearson Correlation	0.999087352		0.99984054		0.772466		0.999627	
Hypothesized Mean Difference	0		0		0		0	
df	29		29		29		29	
t Stat	-5.019986726		-1.34189699		4.226225		3.15773	
P(T<=t) one-tail	1.19977E-05		0.095021203		0.000108		0.001848	
t Critical one-tail	1.699127027		1.699127027		1.699127		1.699127	
P(T<=t) two-tail	2.39954E-05		0.190042406		0.000216		0.003696	
t Critical two-tail	2.045229642		2.045229642		2.04523		2.04523	

Table 3. Hypothesis Testing for The NEHLPD1 Method

	MAKESPAN				TIME			
	NEHLPD1	CDS	NEHLPD1	NEH	NEHLPD1	CDS	NEHLPD1	NEH
Mean	630.4333333	628.9	630.4333333	608.2333	56.03333	62.03333	56.03333	221.6667
Variance	134025.5644	135583.6103	134025.5644	124050.5	3403.62	2363.206	3403.62	60620.02
Observations	30	30	30	30	30	30	30	30
Pearson Correlation	0.999098692		0.9992239		0.779567		0.995303	
Hypothesized Mean Difference	0		0		0		0	
df	29		29		29		29	
t Stat	0.533839385		6.133680319		-0.8961		-4.81971	
P(T<=t) one-tail	0.298760638		5.52246E-07		0.188789		2.09E-05	
t Critical one-tail	1.699127027		1.699127027		1.699127		1.699127	
P(T<=t) two-tail	0.597521276		1.10449E-06		0.377578		4.19E-05	
t Critical two-tail	2.045229642		2.045229642		2.04523		2.04523	

Table 4. Hypothesis Testing for The NEHLPD2 Method

	MAKESPAN				TIME			
	NEHLPD2	CDS	NEHLPD2	NEH	NEHLPD2	CDS	NEHLPD2	NEH
Mean	617.3	628.9	617.3	608.2333	194.1667	62.03333	194.1667	221.6667
Variance	129148.4931	135583.6103	129148.4931	124050.5	43097.18	2363.206	43097.18	60620.02
Observations	30	30	30	30	30	30	30	30
Pearson Correlation	0.999453456		0.999609694		0.779682		0.99626	
Hypothesized Mean Difference	0		0		0		0	
df	29		29		29		29	
t Stat	-4.255923549		4.052935695		4.197822		-3.48015	
P(T<=t) one-tail	9.94446E-05		0.000173218		0.000117		0.000803	
t Critical one-tail	1.699127027		1.699127027		1.699127		1.699127	
P(T<=t) two-tail	0.000198889		0.000346435		0.000233		0.001606	
t Critical two-tail	2.045229642		2.045229642		2.04523		2.04523	

6. Conclusion

After carrying out a valid computerization process and testing the method using hypothesis data on random cases, the development methods, namely NEHLPD, NEHLPD1 and NEHLPD2, have their respective advantages and disadvantages. If we are concerned with the most optimal solution, then NEHLPD is very suitable to be used in finding solutions to flow shop scheduling problems. Where this method produces an average makespan is more efficient than the CDS method and the NEH method where the average efficiency level is 3.56% and 0.28%, respectively.

However, if we are concerned with the speed of computation time in generating solutions to the flow shop scheduling problem, the NEHLPD1 method is suitable for use because it produces 9.67% faster computational time than the CDS method and 74.72% than the NEH method.

Meanwhile, if we are concerned with the makespan value and computation time, the NEHLPD2 method is suitable to be applied because it produces a processing time of 12.41% faster than the NEH method and produces an average makespan solution of 1.84% better than the CDS method and only 1.49% worse than the makespan of NEH method.

References

Alharkan, I. M., *Algorithms for Sequencing and Scheduling*, 2019.

Baker, K. R., & Trietsch, Dan., *Principles of sequencing and scheduling*. John Wiley, 2009.

Campbell, H. G., Dudek, R. A., & Smith, M. L., A Heuristic Algorithm for the n Job, m Machine Sequencing Problem. *Management Science*, *16*(10), B630–B637. http://www.jstor.org/stable/2628231, 1970.

Dong, X., Huang, H., & Chen, P., An improved NEH-based heuristic for the permutation flowshop problem. *Computers and Operations Research*, *35*(12), 3962–3968. https://doi.org/10.1016/j.cor.2007.05.005, 2008.

Fernandez-Viagas, V., & Framinan, J. M., On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. *Computers and Operations Research*, *45*, 60–67. https://doi.org/10.1016/j.cor.2013.12.012 2014.

Gozali, L., Kurniawan, V., & Nasution, S. R., Design of Job Scheduling System and Software for Packaging Process with SPT, EDD, LPT, CDS and NEH algorithm at PT. ACP. *IOP Conference Series: Materials Science and Engineering*, *528*(1). https://doi.org/10.1088/1757-899X/528/1/012045, 2019.

Jin, F., Song, S., & Wu, C., An improved version of the NEH algorithm and its application to large-scale flow-shop scheduling problems. *IIE Transactions (Institute of Industrial Engineers)*, *39*(2), 229–234. https://doi.org/10.1080/07408170600735553, 2007.

Kalczynski, P. J., & Kamburowski, J., An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers and Operations Research*, *35*(9), 3001–3008. https://doi.org/10.1016/j.cor.2007.01.020, 2008.

Liu, W., Jin, Y., & Price, M., A new improved NEH heuristic for permutation flowshop scheduling problems. *International Journal of Production Economics*, *193*, 21–30. https://doi.org/10.1016/j.ijpe.2017.06.026, 2017.

Low, C., Yeh, J. Y., & Huang, K. I., A robust simulated annealing heuristic for flow shop scheduling problems. *International Journal of Advanced Manufacturing Technology*, *23*(9–10), 762–767. https://doi.org/10.1007/s00170-003-1687-x, 2004.

Nagano, M. S., & Moccellin, J. v., A high quality solution constructive heuristic for flow shop sequencing. *Journal of the Operational Research Society*, *53*(12), 1374–1379. https://doi.org/10.1057/palgrave.jors.2601466, 2002.

Nawaz, M., Enscore, E. E., & Ham, I., A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9, 1983.

Pinedo, M. L., *Scheduling* (Vol. 5). https://doi.org/10.1007/978-3-319-26580-3, 2001.

Taillard, E., Benchmarks for basic scheduling problems. *European Journal of Operational Research*, *64*(2), 278–285. https://doi.org/https://doi.org/10.1016/0377-2217(93)90182-M, 1993.

Utami, I. D., Kuswandi, I., & Wibowo, D. E., Comparation of Schedulling Methods: Campbell Dudek Smith, Palmer and Dannenbring to Minimize Makespan. *Journal of Physics: Conference Series*, *1569*(3). https://doi.org/10.1088/1742-6596/1569/3/032019, 2020.

Biographies

Latief Anggar Kurniawan is a master's degree student in the Industrial Engineering Department, Faculty of Engineering Universitas Indonesia, and head of the marketing analysis department at one of the SOEs in Indonesia. He has research interests in the field of market research and the development of new methods.

Farizal is a senior lecturer in Management System in the Industrial Egineering Department, Faculty of Engineering Universitas Indonesia. He earned Bachelor of Engineering degree from Universitas Indonesia, Master degree from Oklahoma State University and Doctoral degree in from University of Toledo. His research interest in reliability design optimization, renewable energy, supply chain management and techno-economy.