Production scheduling using heuristic pour algorithm, branch and bound, and Nawaz Enscore and Ham (NEH) methods application in Butsudan industry

A B R Indah, S Asmal, S Mangngenre, and T N Istiqa

Department of Industrial Engineering, Universitas Hasanuddin, Makassar, Indonesia

Email : a.besseriyani@gmail.com

Abstract. Butsudan production companies have a major problem, namely that the production process has been delayed so that it is difficult to meet production targets on time. The problem is because the scheduling system used is not optimal. Based on these problems, the purpose of this study is to minimize the makespan value by determining the right sequence of jobs. In this study, the method used is the Heuristic Pour Algorithm, Branch and Bound, and NEH. Based on the results of the calculation using the company method, the makespan is 9.958,4 minutes, and the Heuristic Pour method is obtained by the makespan value of 9.461,1 minutes with job sequence J3-J2-J1-J4, the Branch and Bound method obtained by the makespan value of 9.717,3 minutes with job sequence J3-J1-J2-J4, the NEH method obtained the makespan result of 9.846,5 minutes with job sequence J2-J4-J1-J3. So using the proposed method can minimize makespan by 111,9 minutes to 497,3 minutes or about 8 hours.

1. Introduction

One of the crucial problems in the production system is how to arrange and schedule jobs so that orders can be completed following the contract. One effort to achieve the above objectives is to schedule a planned production process [1]. Proper scheduling of production processes can reduce idle time in production units and minimize work in process. One measure of success in scheduling is the completion time (makespan) to be more minimum [2]. Some of the objectives of scheduling activities are to increase the use of resources or reduce waiting time, so that total processing time can be reduced and productivity can be increased, reducing the inventory of half-finished goods or reducing the number of jobs waiting in the queue when the existing resources are still doing other tasks, reduce some delays in work that has a deadline to complete so that it will minimize penalty costs (delay costs), and help make decisions about planning plant capacity and the type of capacity needed so that the addition of expensive costs can be avoided [3]. Right scheduling criteria are minimizing completion time, which can be seen from its makespan, maximizing utility by minimizing its idle timeline, minimizing WIP (work in process) with minimizing flow time and minimizing earliness, minimizing customer waiting time with several tardy jobs, mean lateness, maximum lateness and mean queue time [2].

Butsudan is a ritual place for Japanese people that are used to make offerings to the ancestors of Japanese people [4]. Butsudan's production will be exported to various distributors and consumers in Japan. The problem that is often faced by Butsudan producing companies is that the production process is often delayed, so it is difficult to meet production targets on time. With the increase in the number of requests, proper scheduling is needed, so that production results are more optimal. The purpose of this
study is to apply the Heuristic Pour, Branch and Bound, and NEH methods in optimizing job scheduling, determine the comparison of scheduling methods applied by the company and the proposed method, and determine the suggestions and recommendations for improvement that are most suitable for Butsudan production companies.

In a previous study, Sulaksmi et al. [5] used the Heuristic Pour Method to minimize the makespan of 8.09 hours or 19.25% of the company's current method. Amjadi et al. [6] used two heuristic algorithms to estimate the electricity demand of the target years. These two algorithms are Genetic algorithm (GA) and Particle Swarm Optimization (PSO). GA algorithm is capable of containing discrete or discontinuous variables and non-linear constraints, and PSO is an optimization tool that provides a population-based search procedure in which individuals, called particles, change their positions with time. The obtained result reveals that the PSO can be used as an alternative solution algorithm to estimate the future demand values of the energy. Gmys et al. [7] used Branch and Bound algorithm for the permutation of flowshop problem (PFSP) with the makespan objective. Branch-and-Bound (B&B) is the method used to solve combinatorial optimization problems. The recursion algorithm describes the initial problem starting with building and exploring the search tree dynamically, the root node represents the initial problem. In contrast, the leaf node is a possible solution and the internal node is a subproblem of the initial problem. Soto et al. [8] solved the multi-objective flexible job shop scheduling problem using Branch and Bound (B&B) algorithm. B&B is used to solve the multi-purpose flexible workshop scheduling problem, where three objectives must be minimized, namely makespan, maximum workload, and the total workload of all machines.

2. Research Method
The research was carried out in Butsudan producing companies in February-March 2019. The types of Butsudan studied were Ain-L (16-48), Yuunagi-L (18-40), Kakomi-L (18-40), and Seera-L (16-40). The method used in this study is the Heuristic Pour Algorithm, Branch and Bound, and NEH. The scheduling criterion is to determine the value of the makespan and the sequencing of the work process as well as the Make to Stock (MTS) scheduling with flowshop flow.

In this study, the parameter used to measure the performance of scheduling is the value of makespan. Therefore, the best scheduling is the scheduling method that produces the smallest makespan. The discussion of the analysis of the final results is a comparison of the makespan of each method of scheduling, which is the corporate method (intuitive) and the three proposed methods.

3. Results and Discussion

3.1. Name and Number of Machines
The following Table 1 shows the name and number of machines used in the production process at each work station.

| No. | Machine         | Capacity (Pcs) | Qty |
|-----|-----------------|----------------|-----|
| 1   | Panel Saw (M1) | 1              | 1   |
| 2   | Hot Press (M2) | 6              | 1   |
| 3   | Yokosuri (M3)  | 1              | 1   |
| 4   | Double Saw (M4)| 1              | 1   |
| 5   | NC Router (M5) | 1              | 2   |
| 6   | Kazaban (M6)   | 1              | 1   |
| 7   | Panel Press (M7)| 3              | 3   |
| 8   | Strock Belt Sander (M8)| 1    | 2   |
| 9   | Painting 1 (M9) | 3              | 4   |
| 10  | Assambly (M10) | 1              | 1   |
Table 2 shows the types of meatballs which will be produced for February 2019.

Table 2. Types and Requests of Butsudan in February 2019

| Type        | Size (Inch) | Demand (Unit) | Number of Parts |
|-------------|-------------|---------------|-----------------|
| Ain-L (J1)  | 16-48       | 20            | 59              |
| Yuunagi-L (J2) | 18-40      | 16            | 52              |
| Kakomi-L (J3) | 18-40      | 16            | 49              |
| Seera-L (J4) | 16-40       | 20            | 62              |

Table 3 shows the total time for all parts of each job.

Table 3. Processing Time for Butsudan Production Unit

| Job | M1     | M2     | M3     | M4     | M5     | M6     | M7     | M8     | M9     | M10    |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| J1  | 2,824.7| 2,401.9| 359.7  | 764.0  | 7,234.5| 3,886.6| 6,059.9| 3,377.5| 12,667.3| 3,137.5|
| J2  | 2,384.9| 1,729.4| 428.2  | 685.1  | 5,575.5| 3,380.6| 5,080.4| 2,937.8| 10,288.2| 3,138.6|
| J3  | 2,312.9| 1,633.6| 328.3  | 657.9  | 6,434.1| 3,297.7| 5,025.7| 2,926.4| 9,808.9 | 3,072.7|
| J4  | 2,829.8| 2,497.3| 393.1  | 765.1  | 11,369.1| 4,642.8| 6,106.5| 3,857.7| 13,207.8| 3,069.9|
Figure 1. Production Process Flow Scheme

The data in Table 3 is the time for each process and from M1 to M10. Where:

a. M1, M3, M4, M6, M10 are obtained from the product of demand x standard time

b. M2 is obtained from (demand x standard time) / engine capacity

c. M5, M8 is obtained from (demand x standard time) / number of machines

d. M6 is obtained from the product of demand x standard time

e. M7, M9 is obtained from the result (demand/engine capacity) x (standard time/number of machines)

The following is the calculation time for each process in the form of minutes obtained from the analysis results above.

Table 4. Calculation Process Time Data for All Butsudan Production Requests

| Type (Job)  | Standard Time (Minutes) |
|-------------|-------------------------|
|             | M1  | M2  | M3  | M4  | M5  | M6  | M7  | M8  | M9  | M10 |
| Ain-L (J1)  | 941.6 | 133.4 | 119.9 | 254.7 | 1205.8 | 1295.5 | 224.4 | 562.9 | 351.9 | 1045.8 |
| Yunagi-L (J2) | 636.0 | 76.9  | 114.2 | 182.7 | 743.4  | 901.5  | 150.5  | 391.7  | 228.6  | 837.0 |
| Kakomi-L (J3) | 616.8 | 72.6  | 87.5  | 175.4 | 857.9  | 879.4  | 148.9  | 390.2  | 218.0  | 819.4 |
| Seera-L (J4) | 943.3 | 138.7 | 131.0 | 255.0 | 1894.9 | 1547.6 | 226.2  | 642.9  | 366.9  | 1023.3 |

3.2. Company Scheduling Method

Table 5 is a form of simplification made to calculate the production time by the company method. The time interval between one engine and another is not calculated because parts that are worked on one machine must be finished by batch. Based on the above table, the makespan obtained is 9,958.4 minutes and the Gantt chart model are obtained as follows.

Table 5. Time Calculation Data with the Company Method

| Job | Machine (Minutes) |
|-----|-------------------|
|     | M1   | M2   | M3   | M4   | M5   | M6   | M7   | M8   | M9   | M10  |
| 1   | Start | 0.0  | 941.6| 1,075.| 1,194.| 1,449.| 2,655.| 3,950.9| 4,175.3| 4,738.2| 5,090. |
|     | Finish| 941.6| 1,075.| 1,194.| 1,449.| 2,655.| 3,950.9| 4,175.3| 4,738.2| 5,090.1| 6,135. |
| 2   | Start | 941.6| 1,577.| 1,654.| 1,768.| 2,655.| 3,950.9| 4,852.4| 5,002.9| 5,394.6| 6,135.9 |
|     | Finish| 1,577.| 1,654.| 1,768.| 1,951.| 3,398.| 4,852. | 5,002.9| 5,394.6| 5,623.2| 6,972.9 |
| 3   | Start | 1,577.| 2,194.| 2,266.| 2,354.| 3,398.| 4,852. | 5,731.8| 5,880.7| 6,270.9| 6,972.9 |
Table 5 is a form of simplification made to calculate the production time by the company method. The time interval between one engine and another is not calculated because parts that are worked on one machine must be finished by batch. Based on the above table, the makespan obtained is 9,958.4 minutes and the Gantt chart model is obtained as follows.

![Gantt Chart with Company Scheduling Method](image)

### 3.3. Heuristic Pour Scheduling Method

The heuristic pour algorithm can solve flow shop scheduling to minimize the makespan based on a combination approach. The following are the results of the steps for completing the scheduling using the pour method.

Job 1 was chosen as the main job. Job 1 is the job that ranks first so that the Job 1 processing time on all machines is considered 0. Place a Job other than Job 1 as the first in the next sequence.

| Job | Machine (Minutes) |
|-----|-------------------|
|     | M1    | M2    | M3    | M4    | M5    | M6    | M7    | M8    | M9    | M10   |
| J1  | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   |
| J2  | 636.0 | 76.9  | 114.2 | 182.7 | 743.4 | 901.5 | 150.5 | 391.7 | 228.6 | 837.0 |
| J3  | 616.8 | 72.6  | 87.5  | 175.4 | 857.9 | 879.4 | 148.9 | 390.2 | 218.0 | 819.4 |
| J4  | 943.3 | 138.7 | 131.0 | 255.0 | 1,894 | 1,547 | 226.2 | 642.9 | 366.9 | 1,023.3 |

The shortest processing time for each machine is:

M1=J3 ; M2=J3 ; M3=J3 ; M4=J3 ; M5=J3 ; M6=J3 ; M7=J3 ; M8=J3 ; M9=J3 ; M10=J3

Additional processing time for each $P_{ij}$ starting from the smallest to the largest.

| Job | Machine (Minutes) |
|-----|-------------------|
|     | M1    | M2    | M3    | M4    | M5    | M6    | M7    | M8    | M9    | M10   |
| J1  | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   |
| J2  | 1,252 | 149.5 | 201.7 | 358.1 | 1,601 | 1,780 | 299.4 | 781.9 | 446.6 | 1,656.3 |
| J3  | 616.8 | 72.6  | 87.5  | 175.4 | 743.4 | 879.4 | 148.9 | 390.2 | 218.0 | 819.4 |
M1J1=0 ; M1J3= J3 ; M1J2= J3+J2 ; M1J4= J3+J2+J4. Do the same for the next machine.

C_i calculation for every existing Job

| Job | Machine (Minutes) |
|-----|-------------------|
| J1  | 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 |
| J2  | 1,252.7 149.5 201.7 358.1 1601.3 1,780.9 299.4 781.9 446.6 1,656.3 8,525.8 |
| J3  | 616.8 72.6 87.5 175.4 743.4 879.4 148.9 390.2 218.0 819.4 4151.6 |
| J4  | 2,196.0 288.2 332.8 613.2 3,496.1 3,328.5 525.6 1,424.8 813.5 2,679.6 15,698.3 |

Ci starts from the smallest to the largest where Job 1 is the first order, the job sequence is obtained C1<C3<C2<C4

Table 8. C_i Time Data for Every Job

| Job | Machine (Minutes) |
|-----|-------------------|
| J1  | 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 |
| J2  | 1,252.7 149.5 201.7 358.1 1601.3 1,780.9 299.4 781.9 446.6 1,656.3 8,525.8 |
| J3  | 616.8 72.6 87.5 175.4 743.4 879.4 148.9 390.2 218.0 819.4 4151.6 |
| J4  | 2,196.0 288.2 332.8 613.2 3,496.1 3,328.5 525.6 1,424.8 813.5 2,679.6 15,698.3 |

Based on Table 9, the Makespan number of 9,958.4 minutes can be obtained.

For job 2, job 3 and job 4 performed the same steps as the first order to get the smallest makespan results. The following are the results of recapitulation calculations using the Heuristic Pour method. So that the results obtained by the job sequence are J3-J2-J1-J4.

Table 10. Heuristic Pour Calculation Results

| Job Sequence | Job | Makespan  |
|--------------|-----|-----------|
| 1            | 1   | 9,958.4   |
| 2            | 1   | 9,609.8   |
| 3            | 1   | 9,364.1   |
| 4            | 1   | 10,172.0  |

The following are the results of the calculation of the makespan using the heuristic pour method. The results of the makespan are 9,461.1 minutes.
### Table 11. Makespan Value Results of the Pour Algorithm Method

| Job | Machine (Minutes) |
|-----|-------------------|
| Start | 0.0 | 616.8 | 689.4 | 776.9 | 952.4 | 1,810.1 | 2,689.2 | 2,838.5 | 3,228.7 | 3,446.7 |
| Finish | 616.8 | 689.4 | 776.9 | 952.4 | 1,810.1 | 2,689.2 | 2,838.5 | 3,228.7 | 3,446.7 |
| Start | 1,252.7 | 1,329.6 | 1,443.8 | 1,810.1 | 2,689.2 | 2,838.5 | 3,228.7 | 3,446.7 | 4,133.1 | 4,362.0 |
| Finish | 1,252.7 | 1,329.6 | 1,443.8 | 1,810.1 | 2,689.2 | 2,838.5 | 3,228.7 | 3,446.7 | 4,133.1 | 4,362.0 |
| Start | 2,194.3 | 2,327.6 | 2,447.7 | 5,054.9 | 5,279.1 | 5,842.6 | 6,194.9 | 7,240.8 |
| Finish | 2,194.3 | 2,327.6 | 2,447.7 | 5,054.9 | 5,279.1 | 5,842.6 | 6,194.9 | 7,240.8 |
| Start | 3,137.0 | 3,276.3 | 3,407.4 | 3,759.5 | 5,654.3 | 7,201.8 | 7,428.0 | 8,070.8 |
| Finish | 3,137.0 | 3,276.3 | 3,407.4 | 3,759.5 | 5,654.3 | 7,201.8 | 7,428.0 | 8,070.8 |

![Figure 3. Gantt Chart with Heuristic Pour Scheduling Method](image)

#### 3.4. Branch and Bound Scheduling Method

The Branch and Bound Algorithm, or commonly abbreviated as B&B, is a method of finding solutions in a systematic solution space, which is implemented into a dynamic status space tree. Based on the completion steps using the BB method, the results are:

**Iteration 1**

| Partial Sequence | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 |
|------------------|----|----|----|----|----|----|----|----|----|-----|
| q1               | 941.6 | 1,075.0 | 1,194.9 | 1,449.6 | 2,655.4 | 3,950.9 | 4,175.4 | 4,738.2 | 5,090.1 | 6,135.9 |
| q2               | 636.0 | 712.8 | 827.0 | 1,009.7 | 1,753.1 | 2,654.6 | 2,805.3 | 3,196.8 | 3,425.5 | 4,262.4 |
| q3               | 616.8 | 689.4 | 776.9 | 952.4 | 1,810.2 | 2,689.6 | 2,838.5 | 3,228.7 | 3,446.7 | 4,266.1 |
| q4               | 943.3 | 1,082.0 | 1,213.1 | 1,468.1 | 3,363.0 | 4,910.5 | 5,136.7 | 5,779.7 | 6,146.5 | 7,169.8 |
Table 13. Lower Bound Iteration 1 (Minutes)

| Partial Sequence | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | Lmax |
|------------------|----|----|----|----|----|----|----|----|----|-----|------|
| l1               | 6.764 | 4.912 | 4.963 | 5.315.5 | 8.667 | 8.855 | 6.128 | 5 | 7.200.4 | 6.723 | 8.815.6 | 8.855.8 |
| l2               | 6.786 | 4.634 | 4.654 | 2.718.2 | 8.167 | 7.953.6 | 4.832 | 2 | 5.830.2 | 5.181 | 7.150.9 | 8.167.4 |
| l3               | 6.764 | 4.588 | 4.577 | 4.897.5 | 8.163 | 8.042.0 | 4.897 | 0 | 5.891.9 | 5.231 | 7.172.2 | 8.163.5 |
| l4               | 6.764 | 4.914 | 4.970 | 5.333.6 | 8.625 | 9.563.4 | 7.088 | 1 | 8.161.8 | 7.764 | 9.872.0 | 9.8720  |

Based on Table 12, the smallest lower bound value is partial sequence 3 (L3), which is equal to 8,163.5 minutes so that for the next iteration, L3 is removed.

Iteration 2

Table 14. Partial Sequence Iteration 2 (Minutes)

| Partial Sequence | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 |
|------------------|----|----|----|----|----|----|----|----|----|-----|
| q1               | 1.558 | 1.691 | 1.811 | 2.066.4 | 3.272 | 4.567.6 | 4.792 | 1 | 5.355.0 | 5.706 |
| q2               | 1.252 | 1.329 | 1.443 | 1.626.5 | 2.369 | 3.271.4 | 3.421 | 9 | 3.813.6 | 4.042 |
| q4               | 1.560 | 1.698 | 1.829 | 2.084.9 | 3.979 | 5.527.3 | 5.753 | 5 | 6.396.4 | 6.763 |

Based on Table 14, the smallest lower bound value is partial sequence 1 (L1), which is 8,624.5 minutes, so that for the next iteration, L3 is removed. Job Sequence: J3

Iteration 3

Table 15. Lower Bound Iteration 2 (Minutes)

| Partial Sequence | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | Lmax |
|------------------|----|----|----|----|----|----|----|----|----|-----|------|
| l1               | 6.764 | 5.457.0 | 5.492.0 | 5.756.8 | 8.419.7 | 8.624.5 | 6.626.1 | 7 | 7.455.2 | 7.139 | 8.613.0 | 8.624.5 |
| l2               | 8.331 | 6.662.7 | 6.635.8 | 6.822.5 | 8.951.1 | 8.299.5 | 5.833.1 | 1 | 6.409.6 | 5.784 | 6.948.3 | 8.951.1 |
| l4               | 6.764 | 5.458.7 | 5.499.3 | 5.774.9 | 8.438.2 | 9.332.1 | 7.585.7 | 7 | 8.416.6 | 8.367 | 9.669.4 | 9.669.4 |

Based on Table 14, the smallest lower bound value is partial sequence 1 (L1), which is 8,624.5 minutes, so that for the next iteration, L1 is removed. Job Sequence: J3-J1

Iteration 3

Table 16. Partial Sequence Iteration 3 (Minutes)

| Partial Sequence | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 |
|------------------|----|----|----|----|----|----|----|----|----|-----|
| q2               | 2.194 | 2.271.2 | 2.385.4 | 2.568.1 | 3.311.5 | 4.212.9 | 4.363.5 | 4.755.2 | 4.983.8 | 5.820.8 |
| q4               | 2.501 | 2.640.4 | 2.640.4 | 2.771.4 | 4.921.3 | 6.468.9 | 6.695.1 | 7.338.0 | 7.704.9 | 8.728.2 |
Table 17. Lower Bound Iteration 3 (Minutes)

| Partial Sequence | M1    | M2    | M3    | M4    | M5    | M6    | M7    | M8    | M9    | M10   |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 12               | 6,456.7 | 5,897.6 | 5,935.0 | 6,003.4 | 6,564.0 | 6,722.2 | 5,971.3 | 6,212.5 | 6,049.4 | 6,657.7 | 6,722.2 |
| 14               | 7,071.3 | 6,328.7 | 6,337.8 | 6,534.2 | 9,325.4 | 9,624.3 | 8,378.5 | 9,046.5 | 8,908.7 | 9,751.5 | 9,751.5 |
| Lmax             | 6,722.2 |        |        |        |        |        |        |        |        |        |

Based on Table 16, the smallest lower bound value is partial sequence 2 (L2), which is 6,722.2 minutes. So for the job sequence is J3-J1-J2-J4. The following are the results of the makespan value, which is 9,717.3 minutes.

Table 18. Makespan Value Results using the Branch and Bound Method

| Job | Start | M1   | M2   | M3   | M4   | M5   | M6   | M7   | M8   | M9   | M10  |
|-----|-------|------|------|------|------|------|------|------|------|------|------|
| 3   | 0.0   | 616.8| 689.4| 776.9| 952.4| 1,810.2| 2.6| 2,689.2| 2,838.7| 3,228.7| 3,446.7| 7.7 |
| Finish | 616.8 | 689.4 | 776.9 | 952.4 | 1,810.2 | 2.6 | 2,689.2 | 2,838.7 | 3,228.7 | 3,446.7 | 7.7 |
| 1   | 1,558.3| 1,691.8 | 1,811.7 | 2,066.3 | 3,272.1 | 4,567.6 | 4,792.7 | 5,355.0 | 5,706.9 | 6,752.1 | 7.9 |
| Finish | 1,558.3| 1,691.8 | 1,811.7 | 2,066.3 | 3,272.1 | 4,567.6 | 4,792.7 | 5,355.0 | 5,706.9 | 6,752.1 | 7.9 |
| 2   | 1,558.3| 2,194.3| 2,271.4 | 2,385.3 | 3,272.1 | 4,567.6 | 5,469.7 | 5,619.0 | 6,011.7 | 6,752.1 | 7.9 |
| Finish | 2,194.3| 2,271.4 | 2,385.3 | 2,568.4 | 4,015.5 | 5,469.7 | 5,619.0 | 6,011.7 | 6,240.4 | 7,589.3 | 7.9 |
| 4   | 2,194.3| 3,137.3| 3,276.4 | 3,407.3 | 4,015.5 | 5,910.4 | 7,458.7 | 7,684.6 | 8,327.3 | 8,694.3 | 9.7 |
| Finish | 3,137.3| 3,276.4 | 3,407.3 | 3,662.4 | 5,910.4 | 7,458.7 | 7,684.6 | 8,327.3 | 8,694.3 | 9.7 |

Figure 4. Gantt Chart Using the Branch and Bound Method
The following are the results of the tree diagram using branch and bound.

![Figure 5. Branch and Bound Method Tree Diagrams](image)

In Figure 5 above, shows the branching tree generated from the lower bound (LB) value of each current job. The number 0 indicates the branching root. Figures 1, 2, 3, 4 show the work done. Among these jobs, selected jobs that have LB values are then used as the next branch.

### 3.5. Enscore and Ham (NEH) Scheduling Method

Nawaz Enscore and Ham (NEH) is a method used to produce the best solution in terms of changing the position of work (job) so that it will produce several possible sequences of workmanship to achieve the most optimal results and a good job sequence. The steps for completion using the NEH method are as follows. Determining the number of iterations performed

\[
(n\times(n+1)/2)-1 \text{ iteration} \\
(4\times(4+1)/2)-1=9 \text{ iteration}
\]

Calculating the amount of processing time for each job

| Table 19. Total Processing Time |
|-------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Job | M1 (Minutes) | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | Total Time |
|-----|--------------|----|----|----|----|----|----|----|----|-----|------------|
| J1  | 941.6        | 133.4 | 119.9 | 254.7 | 1,205.8 | 1,295.5 | 224.4 | 562.9 | 351.9 | 1,045.8 | 6,135.9 |
| J2  | 636.0        | 76.9 | 114.2 | 182.7 | 743.4 | 901.5 | 150.5 | 391.7 | 228.6 | 837.0 | 4,262.4 |
| J3  | 616.8        | 72.6 | 87.5 | 175.4 | 857.9 | 879.4 | 148.9 | 390.2 | 218.0 | 819.4 | 4,266.1 |
| J4  | 943.3        | 138.7 | 131.0 | 255.0 | 1,894.9 | 1,547.6 | 226.2 | 642.9 | 366.9 | 1,023.3 | 7,169.8 |

Next step is sorting by LPT (Longest Processing Time), i.e., J4-J1-J2-J3. Trying the first two sequences of LPT that have been calculated and select the smallest makespan. Then the order will be used as a reference for further calculations.
Table 20. Iteration Results 1

| Machine | Job | Duration | Start | Finish |
|---------|-----|----------|-------|--------|
| 1       | 4   | 943.3    | 0     | 943.3  |
| 1       | 1   | 941.6    | 943.3 | 1,884.9|
| 2       | 4   | 138.7    | 943.3 | 1,082.0|
| 2       | 1   | 133.4    | 1,884.9| 2,018.3|
| 3       | 4   | 131.0    | 1,082.0| 1,213.1|
| 3       | 1   | 119.9    | 2,018.3| 2,138.2|
| 4       | 4   | 255.0    | 1,213.1| 1,468.1|
| 4       | 1   | 254.7    | 1,468.1| 1,722.8|
| 5       | 4   | 1,894.9  | 1,468.1| 3,363.0|
| 5       | 1   | 1,205.8  | 1,722.8| 2,928.5|
| 6       | 4   | 1,547.6  | 3,363.0| 4,910.5|
| 6       | 1   | 1,295.5  | 2,928.5| 4,224.1|
| 7       | 4   | 226.2    | 4,910.5| 5,136.7|
| 7       | 1   | 224.4    | 4,448.5| 4,448.5|
| 8       | 4   | 642.9    | 5,136.7| 5,779.7|
| 8       | 1   | 562.9    | 4,448.5| 5,011.4|
| 9       | 4   | 366.9    | 5,779.7| 6,146.5|
| 9       | 1   | 351.9    | 5,011.4| 5,363.3|
| 10      | 4   | 1,023.3  | 6,146.5| 7,169.8|
| 10      | 1   | 1,045.8  | 5,363.3| 6,409.1|

| Cmax    | 6,409.1 |
|---------|---------|
| $F$     | 10,374.40|

The same steps are carried out to get the smallest makespan value. The following is the result of the recapitulation of the value of the makespan (Cmax) and mean flow time ($F$) by using the NEH method to determine the optimal job sequence

Table 21. Calculation Result of NEH Method

| Iteration | Job Sequence  | Cmax    | $F$       |
|-----------|---------------|---------|-----------|
| 1         | J4-J1         | 6,409.1 | 10,374.40 |
| 2         | J1-J4         | 7,406.4 | 9,839.13  |
| 3         | J4-J1-J3      | 4,864.8 | 13,531.08 |
| 4         | J4-J3-J1      | 6,349.0 | 13,470.25 |
| 5         | J3-J4-J1      | 6,022.5 | 13,470.25 |
| 6         | J4-J1-J3-J2   | 4,914.5 | 17,848.10 |
| 7         | J4-J1-J2-J3   | 4,941.6 | 17,852.36 |
| 8         | J4-J2-J1-J3   | 4,941.6 | 17,795.78 |
| 9         | J2-J4-J1-J3   | 4,634.3 | 18,685.26 |

To calculate the makespan based on processing time with job sequence J2-J4-J1-J3 is as follows.
Table 22 is a form of simplification that is done to calculate the production time by the NEH method so that the results of the makespan is 9,846.5 minutes. The time interval between one engine and another is not calculated because parts that are worked on one machine must be finished by batch. Thus, the calculation of the time on the next machine will continue based on the previous engine. Figure 6 shows the Gantt chart model as follows.

3.6. Scheduling Performance Parameters
The comparison of time calculations between company methods, heuristic pour algorithm methods, branch and bound, and NEH is as follows.
Table 23. Comparison of Makespan for Each Method

| Performance level | Company Method | Heuristic Pour Method | Branch and Bound Method | NEH Method |
|-------------------|----------------|-----------------------|-------------------------|------------|
| Job Sequence      | J1-J2-J3-J4    | J3-J2-J1-J4           | J3-J1-J2-J4             | J2-J4-J1-J3 |
| Makespan (Minute) | 9,958.4        | 9,461.1               | 9,717.3                 | 9,846.5    |

a. The efficiency of the company's makespan method and the Heuristic Pour Algorithm

\[ E_i = \left| \frac{\text{Makespan}_{\text{Company}}}{\text{Makespan}_{\text{Pour}}} \right| = \frac{9,958.4}{9,461.1} = 1.05 \]

Ei value > 1, then the Pour method performance is good.

\[ RE = \frac{\text{Makespan}_{\text{Company}} - \text{Makespan}_{\text{Pour}}}{\text{Makespan}_{\text{Company}}} \times 100\% \]

\[ RE = \frac{9,958.4 - 9,461.1}{9,958.4} \times 100\% = 4.9\% \]

b. The efficiency of the Company and Branch and Bound methods makespan

\[ E_i = \left| \frac{\text{Makespan}_{\text{Company}}}{\text{Makespan}_{\text{BB}}} \right| = \frac{9,958.4}{9,717.3} = 1.02 \]

Ei value > 1, then the BB method performance is good.

\[ RE = \frac{\text{Makespan}_{\text{Company}} - \text{Makespan}_{\text{BB}}}{\text{Makespan}_{\text{Company}}} \times 100\% \]

\[ RE = \frac{9,958.4 - 9,717.3}{9,958.4} \times 100\% = 2.4\% \]

c. The efficiency of the Company and NEH makespan methods

\[ E_i = \left| \frac{\text{Makespan}_{\text{Company}}}{\text{Makespan}_{\text{NEH}}} \right| = \frac{9,958.4}{9,846.5} = 1.01 \]

Ei value > 1, then the NEH method performance is good.

\[ RE = \frac{\text{Makespan}_{\text{Company}} - \text{Makespan}_{\text{NEH}}}{\text{Makespan}_{\text{Company}}} \times 100\% \]
Figure 7 is the result of comparing the makespan values using the company method, Heuristic Pour, Branch and Bound, and the NEH method.

Figure 7. Makespan Comparison Results

\[ RE = \frac{9,958.4 - 9,461.1}{9,958.4} \times 100\% = 1.1\% \]

4. Conclusion

Based on the results of the analysis conducted above, it can be concluded that:

1. By applying the Heuristic Pour Algorithm method Branch and Bound method and the Nawaz Enscore Ham method, we get different job sequence results, namely Heuristic Pour with the sequence J3-J2-J1-J4, Branch and Bound with the order J3-J2-J1-J4, and NEH in the order J2-J4-J1-J3. Each method has a priority order for each job.

2. The makespan value generated for the company method is 9,958.4 minutes. Whereas using the Heuristic Pour proposed method has a makespan value of 9,461.1 minutes and RE of 4.9%, Branch and Bound has a makespan value of 9,717.3 minutes and RE of 2.4%, and NEH has a value of makespan of 9,846.5 minutes and RE at 1.1%.

3. From some of the methods that have been proposed, it can be seen that the most optimal method is the Heuristic Pour method with a makespan value of 9,461.1 minutes, job sequence J3-J2-J1-J4 and efficiency of 4.9%. Thus this method can be used as a recommendation for companies to improve company performance and productivity.

References

[1] Haming, M., & Nurnajamuddin M 2007 Manajemen Produksi Modern (Jakarta: Bumi Aksara)
[2] Ginting R 2009 Penjadwalan Mesin (Yogyakarta: Graha Ilmu)
[3] Bedworth, D. D., & Bailey J E 1987 Integrated Production Control Systems (Singapore: John Willey and Sons Inc)
[4] Ogino M 2015 Collective Memory, Sociology of vol 4 (Elsevier)
[5] Sulaksmi A, Kesy Garside A and Hadziqah F 2014 Penjadwalan Produksi Dengan Algoritma Heuristik Pour (Studi Kasus: Konveksi One Way – Malang) J. Tek. Ind. 15 35
[6] Amjadi M H, Nezamabadi-pour H and Farsangi M M 2010 Estimation of electricity demand of Iran using two heuristic algorithms Energy Convers. Manag. 51 493–7
[7] Gmys J, Mezmaz M, Melab N and Tuyttens D 2020 A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem Eur. J. Oper. Res.
[8] Soto C, Dorronsoro B, Fraire H, Cruz-Reyes L, Gomez-Santillan C and Rangel N 2020 Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm Swarm Evol. Comput. 53 100632