Research Article

Hybrid Metaheuristics and Linear Programming for Finite Capacity MRP in Multi-Stage Flexible Flow Shop with Permutation and Non-permutation Scheduling Options

Watchara Songserm and Teeradej Wuttipornpun*
Department of Industrial Engineering, Faculty of Engineering, King Mongkut’s University of Technology North Bangkok, Bangkok, Thailand

Chorkaew Jaturanonda
Department of Production Engineering, Faculty of Engineering, King Mongkut’s University of Technology Thonburi, Bangkok, Thailand

* Corresponding author. E-mail: teeradejw@kmutnb.ac.th DOI: 10.14416/j.ijast.2018.05.004
Received: 11 May 2017; Accepted: 13 July 2017; Published online: 22 May 2018
© 2018 King Mongkut’s University of Technology North Bangkok. All Rights Reserved.

Abstract
This paper presents a new algorithm for Finite Capacity MRP (FCMRP) in a multi-stage flexible flow shop. The proposed algorithm consists of four conventional metaheuristics namely, Genetic Algorithm (GA), Tabu Search (TS), Variable Neighborhood Search (VNS), and Simulated Annealing (SA) hybridized with Linear Programming (LP). The objective is to minimize the total cost, which is the sum of tardiness, earliness, and flow-time costs. There are two main steps of the proposed algorithm. Firstly, an efficient sequence of orders is generated by the proposed metaheuristics in a way that reduce the total cost. In this step, the required operations of the orders are scheduled based on two scheduling options called permutation and non-permutation. Secondly, the total cost is minimized by the LP model. The required parameters of the metaheuristics are tuned by using real data from automotive companies. The result shows that the proposed algorithm significantly outperforms the existing algorithm, and GA obtains the best total cost.

Keywords: Finite capacity MRP, Flexible flow shop, Metaheuristic, Linear programming, Permutation scheduling, Non-permutation scheduling

1 Introduction
Material Requirement Planning (MRP) is a powerful planning tool used to generate production and purchasing plans. Unfortunately, the production plan obtained from the MRP system is reported that it could not be implemented on shop floor. A reason for this is that MRP assumes infinite resource capacity or constant lead-time [1]–[3]. This problem is later called Finite Capacity MRP (FCMRP). Since the FCMRP problem for industrial scale instances is normally the NP-hard class, the metaheuristic algorithm is one of the appropriate approaches to solve the problem [4]–[6]. In the literature, there are two approaches of metaheuristic algorithms developed for the FCMRP problems summarized in Table 1. The first approach is called population search algorithm. It includes Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Cuckoo Search (CS) [7]–[18]. The second approach is called non-population search algorithm. It includes TS, Iterated Local Search (ILS), Variable Neighborhood Search (VNS), and Simulated Annealing (SA) [19]–[34]. Based on the principles of each metaheuristic, the population search algorithm...
seems to be more efficient than the other. However, there is no strong conclusion that the population search approach is always better than the non-population one. It depends on mostly about the problem characteristics.

This paper presented four metaheuristic algorithms, which are GA, TS, SA, and VNS, hybridized with an LP model to solve the industrial scale FCMRP problem. The metaheuristics are used to generate the efficient sequence of orders on machines based on permutation and non-permutation scheduling concepts, whereas the LP model is used to schedule the required operations of orders to the optimal start times. The proposed algorithm is intently developed to improve the solution obtained from the existing FCMRP algorithm developed by Wuttipornpun and Yenradee [6]. The objective is to minimize the total cost defined as the sum of tardiness, earliness, and flow-time costs. The production shop used to evaluate the performance of the proposed algorithm is a multi-stage flexible flow shop. The planning horizon is one month without overtimes and preemptive options.

The remaining of the paper is organized as follows. Section 2 deals with details of the proposed algorithm. Case studies and experiments for parameter tuning and performance evaluation are explained in Section 3. Results and discussion are provided in Section 4. Finally, the conclusion of this paper and recommendations for future research are given in Section 5.
2 Details of the Proposed Algorithm

The proposed algorithm has two main steps as shown in Figure 1. They are explained as follows.

Step 1: Construct the efficient sequence of orders

The objective of this step is to construct the efficient sequence of orders by the conventional GA, SA, VNS, and TS algorithms. The conventional procedures of each algorithm are shown in Figures 2, 3, 4, and 5 respectively. Note that details of these metaheuristics are available in Artificial Intelligent (AI) books [35], [36].

In each iteration of a given metaheuristic algorithm, the sequence of orders is evaluated to obtain the total cost calculated by Equation (1), where \( Q_i \), \( t_i \), \( e_i \), \( e_i \), and \( f_i \) are the order quantity, tardiness, earliness, and flow-time of the \( i \)th order. \( CT_i \), \( CE_i \), and \( CF_i \) are the cost per unit of tardiness, earliness, and flow-time of the \( i \)th order.
To calculate the total cost, the required operations of orders in the sequence must be scheduled to machines in order to calculate the tardiness, earliness, and flow-time. It can be done by the following steps: (1) the orders in the sequence are exploded by variable lead-time MRP (VMRP) to obtain the required operations (see Wuttipornpun and Yenradee [6] for details of VMRP), (2) these operations are scheduled to less tardiness machines by forward scheduling based on permutation and non-permutation scheduling options, and (3) the tardiness, earliness, and flow-time of each order are calculated by the start time, due time, and completion time.

Details of the required operations \( O_{ij} \) of the four orders \( O_i \) after VMRP explosion shown in Figure 6 are used to demonstrate this step. Suppose that a sequence of orders is \( O_4 \Rightarrow O_1 \Rightarrow O_2 \Rightarrow O_3 \). When the permutation option is selected, all operations must be scheduled based on the sequence of orders in a way that the operations of the first order in the sequence are scheduled to less tardiness machines first, and the operations of the second order in the sequence are scheduled next and so on. By this way, it is obvious that all machines have the same sequence of operations that complies with the permutation sequencing concept (see second column of Table 2, \( M_1 \) to \( M_6 \)). When the non-permutation option is selected, the operation that is scheduled later based on the sequence of orders may be started earlier than the operation that is scheduled before. It results in different sequence of operations of each machine as shown in the fourth column of Table 2. It can be seen that the sequence of operations on \( M_6 \) is different from other machines. The total costs of these sequences are shown in Table 2. Note that the permutation and non-permutation scheduling concepts are applied and evaluated separately.

\[
\text{Total cost} = \sum_{i=1}^{n} CT_i Q_i t_i + \sum_{i=1}^{n} CE_i Q_i e_i + \sum_{i=1}^{n} CF_i Q_i f_i \quad (1)
\]

Figure 6: Details of operations after applying the VMRP explosion.
Step 2: Optimize the start times of the operations

From the first step, the operations are scheduled to machines by forward scheduling. As a result, the start times of some operations may not be optimal. This step tries to determine the optimal start times of all operations by the LP model in order to minimize the total cost. In the LP model, only the start times of operations are optimized, whereas the sequence of operations on machines obtained from the first step is still maintained. The LP model is formulated by using indices, parameters, and variables explained as follows.

Indices

- \( i \) = index of order starting from 1 to \( n \)
- \( j \) = index of operation of order \( i \) starting from 1 to \( m \)
- \( j^* \) = index of the last operation of order \( i \)
- \( k \) = index of machine starting from 1 to \( s \)
- \( j' \) = index of child operations of order \( i \) operation \( j \)
- \( j'' \) = index of the first operations of order \( i \)
- \( q \) = index of operations on machine starting from 1 to \( t \). For example, there are three operations on machine 4 \( (M_4) \), which are \( O_{4,1} \), \( O_{4,2} \), \( O_{4,3} \) (see second column in Table 2), when \( q = 2 \), it refers to the second operation on this machine \( (O_{4,2}) \).

Sets and Parameters

- \( p_{ij} \) = processing time of order \( i, j \) (order / operation)
- \( p_{ij,k} \) = processing time of order \( i, j \) (order / operation)
- \( d_i \) = due time of order \( i \)
- \( C_{Ti} \) = cost per unit of tardiness of order \( i \)
- \( CE_i \) = cost per unit of earliness of order \( i \)
- \( CF_i \) = cost per unit of flow-time of order \( i \)
- \( Q_i \) = quantity of order \( i \)
- \( Seq_{k} \) = set of sequence of operation \( i, j \) on machine \( k \). For example, there are three operations on machine 4 \( (M_4) \), which are \( O_{4,1} \), \( O_{4,2} \), \( O_{4,3} \) (see second column in Table 2), so \( Seq_{4} = \{4,3, (3,3), (1,3)\} \).

\[ CH_{ij} = \text{set of child operations of operation } i, j \text{ (index } j) \]. For example, \( O_{1,4} \) consists of \( O_{1,2} \) and \( O_{1,3} \) (see Figure 6), so \( CH_{1,4} = \{2, 3\} \).

\[ FO_i = \text{set of the first operations of order } i \]. For example, the first operations of order 4 are \( O_{4,1} \) and \( O_{4,2} \) (see Figure 6), so \( FO_4 = \{1, 2\} \).

Decision Variables

- \( x_{ij} \) = start time of operation \( i, j \)
- \( x_{ij,k} \) = start time of operation \( i, j \) on machine \( k \)
- \( c_i \) = completion time of order \( i \)
- \( c_{ij} \) = completion time of operation \( i, j \)
- \( f_i \) = flow-time of order \( i \)
- \( e_i \) = earliness of order \( i \)
- \( t_i \) = tardiness of order \( i \)
- \( Z \) = total cost

Objective function [Equation (2)]

\[
\text{Minimize } Z = \sum_{i=1}^{n} C_{Ti} x_i + \sum_{i=1}^{n} CE_i e_i + \sum_{i=1}^{n} CF_i f_i \tag{2}
\]

Constraints

Constraint (3) is used to maintain the sequence of operations on each machine

\[
x_{q,k} \geq x_{q,k} + p_{q,k} \quad \forall k, \forall q, \text{ and } q^* > q \tag{3}
\]

Constraint (4) is used to maintain the precedence relationships of operations

\[
x_{i,j} \geq c_{i,j} \quad \forall i, \forall j, \text{ and } j' \in CH_{i,j} \tag{4}
\]

Constraints (5) – (11) are used to calculate completion time, tardiness, earliness, and flow-time

For completion time:

\[
c_i \geq x_{i,j^*} + p_{i,j^*} \quad \forall i, \forall j^* \tag{5}
\]

\[
c_{i,j} \geq x_{i,j} + p_{i,j} \quad \forall i, \forall j \tag{6}
\]

Table 2: Total cost of a sequence of orders based on permutation and non-permutation schedules

<table>
<thead>
<tr>
<th>Sequence of Orders ((O_i))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Permutation)</th>
<th>Total Cost ((\text{Permutation}))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Non-permutation)</th>
<th>Total Cost ((\text{Non-permutation}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_1 \otimes O_2 \otimes O_3 \otimes O_4 )</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,522.00</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,325.20</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Sequence of Orders ((O_i))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Permutation)</th>
<th>Total Cost ((\text{Permutation}))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Non-permutation)</th>
<th>Total Cost ((\text{Non-permutation}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_1 \otimes O_2 \otimes O_3 \otimes O_4 )</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,522.00</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,325.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence of Orders ((O_i))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Permutation)</th>
<th>Total Cost ((\text{Permutation}))</th>
<th>Sequence of Operations ((O_{ij})) on Machines (Non-permutation)</th>
<th>Total Cost ((\text{Non-permutation}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_1 \otimes O_2 \otimes O_3 \otimes O_4 )</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,522.00</td>
<td>( M_1: O_{1,1} \otimes O_{1,2} \otimes O_{1,3} \otimes O_{1,4} )</td>
<td>$1,325.20</td>
</tr>
</tbody>
</table>
For tardiness:
\[ t_i \geq c_i - d_i \quad \forall i \]  
\[ t_i \geq 0 \quad \forall i \]  

For earliness:
\[ e_i \geq d_i - c_i \quad \forall i \]  
\[ e_i \geq 0 \quad \forall i \]  

For flow-time:
\[ f_i \geq c_i - x_{i,j}^* \quad \forall i, \forall j^* \in FO_i \]  

Non-negativity constraint [Constraint (12)]
\[ x_{i,j,k} \geq 0 \quad \forall i, \forall j, \forall k \]  

Figures 7 and 8 show the results after applying the LP model. It can be seen that the total costs shown in Table 2 are dramatically reduced. This shows that our LP model is very efficient. Also, it observes that the total cost obtained from non-permutation scheduling is slightly better than that obtained from permutation scheduling. However, it can be happened in the opposite direction depending on data sets.

3 Case Studies and Experiments

This section consists of three parts. The first part deals with the details of case studies. The second part deals with the experiment for parameters tuning of GA, SA, TS, and VNS. The final part is the experiment to evaluate the performance of the proposed algorithm. They are explained as follows.

3.1 Details of case studies

Our case studies are derived from three automotive-part companies. The different characteristics are summarized in Table 3, whereas the common characteristics are summarized as follows.

- The production shop is a multi-stage flexible flow shop
- The planning horizon is one month
- Lot sizing rule is lot-for-lot since it results in lowest inventory
- No preemptive schedule and overtime
Table 3: Different characteristics of case studies

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of finished products</td>
<td>16</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>Total order quantities</td>
<td>1,512</td>
<td>1,735</td>
<td>1,986</td>
</tr>
<tr>
<td>Min/Max levels in bill of materials</td>
<td>2/5</td>
<td>3/5</td>
<td>4/6</td>
</tr>
<tr>
<td>Number of operations</td>
<td>520</td>
<td>610</td>
<td>690</td>
</tr>
<tr>
<td>Number of machines</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

3.2 Experiments for parameters tuning of GA, SA, TS, and VNS

The required parameters for GA, SA, TS, and VNS are summarized in Tables 4 and 5. These parameters are considered as independent variables for a factorial experiment. The experiment is conducted with five replicates for all case studies to obtain the response variable, which is the total cost before applying the LP model. The stopping criterion for each run is 120 minutes (set by the planner). Note that data in Tables 4 and 5 are obtained by a set of screening experiments beforehand.

Table 4: Parameters for GA

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$p_s$</th>
<th>Crossover</th>
<th>$p_c$</th>
<th>Mutation</th>
<th>$p_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>5, 10</td>
<td>PMX, PBX</td>
<td>0.6, 0.8</td>
<td>SWAP, INSERT</td>
<td>0.005, 0.01</td>
</tr>
</tbody>
</table>

$p_s$: population size, $p_c$: crossover probability, $p_m$: mutation probability.

To obtain the practical solution for the planner, the best common parameter setting for all case studies is required rather than seeking the best setting for an individual case study. To obtain the best common setting for a given algorithm, the relative percentage deviation over the best total cost of each case study ($RPD_v$) and the average relative percentage deviation for all case studies ($ARPD$) are determined by using Equations (13) and (14), where $Sol_{Alg}$ is the total cost obtained from each run, $Sol_{Best}$ is the minimum total cost across all runs, $v$ is the index of case study, and $V$ is the number of case studies. The best common setting is a setting obtained at the minimum $ARPD$, since it guarantees that the total cost from this setting is very close to its best total cost (near best total cost).

$$RPD_v = \frac{Sol_{Alg(v)} - Sol_{Best(v)}}{Sol_{Best(v)}} \times 100\%$$ (13)

$$ARPD = \frac{1}{V} \sum_{v=1}^{V} (RPD)_v$$ (14)

3.3 Experiments to evaluate the performance of the proposed algorithm

To evaluate the performance, we compare the total cost obtained from the proposed algorithm with a benchmark algorithm (existing FCMRP algorithm by Wuttipornpun and Yenradee [6]). There are 8 options available in the proposed algorithm. They all use the best common settings obtained from section 3.2, and use the following notations.

GA-P: GA/Permutation
GA-NP: GA/Non-permutation
SA-P: SA/Permutation
SA-NP: SA/Non-permutation
TS-P: TS/Permutation
TS-NP: TS/Non-permutation
VNS-P: VNS/Permutation
VNS-NP: VNS/Non-permutation

All options in the proposed algorithm are set as independent variables for a one-way ANOVA experiment. The response variable is the total cost after applying the LP model. To determine the performance improvement of the proposed algorithm,

Table 5: Parameters for TS, SA, and VNS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>No. of NBH Sequences</th>
<th>NBH Operators</th>
<th>TL Size</th>
<th>No. of Iterations for each Temperature</th>
<th>Initial Temperature</th>
<th>Alpha</th>
<th>Shaking Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>10, 15</td>
<td>SWAP, INSERT</td>
<td>30, 40, 50</td>
<td>-</td>
<td>80, 100</td>
<td>0.8, 0.9</td>
<td>-</td>
</tr>
<tr>
<td>SA</td>
<td>5, 10</td>
<td>SWAP, INSERT</td>
<td>-</td>
<td>1, 10</td>
<td>-</td>
<td>-</td>
<td>SWAP, INSERT</td>
</tr>
<tr>
<td>VNS</td>
<td>70, 90</td>
<td>SWAP and INSERT</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SWAP, INSERT</td>
</tr>
</tbody>
</table>

NBH: Neighborhood, TL: tabu list, Alpha: temperature reduction.
the relative percentage improvement over the best total cost obtained from the benchmark algorithm (RPI) is calculated using Equation (15), where $Sol_{Alg}$ is the total cost of each algorithm, $Sol_{BM}$ is the best total cost from the benchmark algorithm.

$$RPI_v = \frac{|Sol_{Alg} - Sol_{BM}|}{Sol_{BM}} \times 100\% \quad (15)$$

### 4 Results and Discussions

The experiments mentioned in section 3 are run by using C# language on a Core i5 Processor 2.5 GHz with 4 GB of RAM desktop computer. The results are analyzed by using a statistical software called Minitab 17.

Since we evaluate the proposed algorithm using three different case studies, the best setting for each case study is then varied depending on the different characteristics. Therefore, the best common setting for all case studies is more practical for the planner. Tables 6 and 7 show the best common setting of each algorithm recommended for all case studies. It is a setting selected from the minimum value of $ARPD$. It is also observed that the permutation option and non-permutation have different parameter settings.

#### Table 6: Best common parameter settings for GA

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$p_s$</th>
<th>Crossover</th>
<th>$p_c$</th>
<th>Mutation</th>
<th>$p_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA-P</td>
<td>10</td>
<td>PBX</td>
<td>0.6</td>
<td>SWAP</td>
<td>0.005</td>
</tr>
<tr>
<td>GA-NP</td>
<td>10</td>
<td>PBX</td>
<td>0.6</td>
<td>INSERT</td>
<td>0.01</td>
</tr>
</tbody>
</table>

$p_s$: population size, $p_c$: crossover probability, $p_m$: mutation probability.

Table 8 shows the best and near best total costs obtained from all available options in the proposed algorithm. Based on $RPD$, it shows that there is only a narrow range of 0–3.85% between the near best total cost and its best total cost. From the $ARPD$ value, it can be seen that VNS-P and VNS-NP obtain the minimum deviation of 0.61%, whereas the SA-NP obtains the maximum deviation of 1.58%. However, the deviations are very small. This proves that our method to select the near best total cost works very well.

Table 9 shows the improved total cost, Relative Percentage Improvement (RPI), and the rank after applying the LP model. The rank numbers shown in this table are obtained from Tukey’s comparison test. The lower rank means better performance than the higher rank. It can be seen that the near best total cost in Table 8 is substantially reduced when applying the LP model.
This proves that the LP model is very efficient. It allows some operations to start prior to the release times from VMRP to reduce the tardiness cost, which is the main cost in our objective function.

Based on the RPI value, it is obvious that all available options in the proposed algorithm significantly outperform the benchmark algorithm. The improvement gap over the benchmark algorithm is in a range of 20.03–31.66%. GA-NP offers the highest improvement of 31.66%, whereas SA-P offers the lowest improvement of 20.03%. For our problem characteristics, GA, SA, and TS with non-permutation option outperform the permutation option, while it happens in the opposite direction for VNS.

Comparing the required runtimes between the proposed algorithm and the benchmark algorithm, which are 120 and 10 minutes, respectively, the proposed algorithm requires substantially longer runtime than the benchmark algorithm. However, the total cost obtained from the proposed algorithm is dramatically lower than that obtained from the benchmark algorithm. Therefore, it is worth for the planner to wait slightly longer.

### 5 Conclusions

This paper presents a new algorithm for the FCMRP problem in automotive multi-stage flexible flow shop. It is a hybrid of metaheuristic algorithms (GA, SA, TS, and VNS) and linear programming model. The objective is to improve the solution obtained from the existing FCMRP algorithm done by Wuttipornpun and Yenradee [6] and also offer a practical alternative method for the planner. There are two main steps in the proposed algorithm. In the first step, the efficient sequence of orders is determined by the proposed metaheuristic algorithms. After obtaining the sequence of orders, the required operations of these orders are scheduled to the less tardiness machines by using the forward heuristic with permutation and non-permutation scheduling options. In the second step, the start times of the required operations are optimally determined using the LP model to minimize the total cost, which is the sum of tardiness, earliness, and flow-time costs.

There are 8 options available in the proposed algorithm. They are controlled by the parameters of metaheuristic algorithms. Since we evaluate the proposed algorithm by different case studies, the best parameter setting for each case study may be set differently. To offer a practical setting for the planner, the best common setting for all case studies is required rather than the best setting for an individual case. Based on the best common setting, it offers a solution which slightly deviates from its best solution in a deviation range of 0–3.85%.

For the effectiveness perspective, all options in the proposed algorithm significantly outperform the existing algorithm for all case studies. GA-NP offers the best improvement of 31.66%. Based on the different characteristics of case studies, we can conclude that the proposed algorithm can be applied to various industrial situations. Although the proposed algorithm requires a significant longer runtime than the existing algorithm, it is worth for the planner to wait since the solution quality of the proposed algorithm is much better.

### Table 9: Total costs after applying the LP model

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Total Cost ($)</th>
<th>RPI</th>
<th>Ranking</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 3</td>
<td></td>
</tr>
<tr>
<td>Existing FCMRP [6]</td>
<td>9,424 (Benchmark)</td>
<td>16,016 (Benchmark)</td>
<td>19,280 (Benchmark)</td>
<td>14,906.58 (Benchmark)</td>
</tr>
<tr>
<td>GA-P</td>
<td>7,300 / 22.54% / 4</td>
<td>11,960 / 25.33% / 4</td>
<td>13,707 / 28.91% / 6</td>
<td>10,988.99 / 25.59% / 3</td>
</tr>
<tr>
<td>GA-NP</td>
<td>7,016 / 25.55% / 3</td>
<td>10,265 / 35.91% / 1</td>
<td>12,814 / 33.54% / 1</td>
<td>10,031.70 / 31.66% / 1</td>
</tr>
<tr>
<td>SA-P</td>
<td>8,305 / 11.87% / 6</td>
<td>12,770 / 20.27% / 6</td>
<td>13,894 / 27.94% / 4</td>
<td>11,656.33 / 20.03% / 4</td>
</tr>
<tr>
<td>SA-NP</td>
<td>7,056 / 25.13% / 3</td>
<td>10,489 / 34.51% / 2</td>
<td>13,787 / 28.49% / 6</td>
<td>10,444.25 / 29.94% / 2</td>
</tr>
<tr>
<td>TS-P</td>
<td>7,540 / 19.99% / 5</td>
<td>12,480 / 22.07% / 4</td>
<td>14,116 / 26.79% / 5</td>
<td>11,378.91 / 22.95% / 3</td>
</tr>
<tr>
<td>TS-NP</td>
<td>7,057 / 25.12% / 3</td>
<td>12,155 / 24.10% / 5</td>
<td>13,681 / 29.04% / 3</td>
<td>10,964.69 / 26.44% / 3</td>
</tr>
<tr>
<td>VNS-P</td>
<td>6,629 / 29.66% / 1</td>
<td>11,476 / 28.35% / 3</td>
<td>13,583 / 29.55% / 3</td>
<td>10,562.67 / 29.18% / 2</td>
</tr>
<tr>
<td>VNS-NP</td>
<td>6,845 / 27.37% / 2</td>
<td>11,961 / 25.32% / 4</td>
<td>13,183 / 31.62% / 2</td>
<td>10,663.34 / 28.47% / 2</td>
</tr>
</tbody>
</table>
There are some interesting research gaps for future investigations. The lot sizing policy is only lot-for-lot. Other lot-sizing techniques should be studied. Since it is especially designed for the flexible flow shop, the performance of the proposed algorithm in other manufacturing shops such as job shop, open shop should be investigated. The production overtime and order preemption are also of interest because they can significantly reduce the total cost. Therefore, a new algorithm should be developed to address these problems.

Acknowledgments

This research has been supported by Faculty of Engineering, King Mongkut’s University of Technology North Bangkok.

References


