Non-Population Search Algorithms for Capacitated Material Requirement Planning in Multi-Stage Assembly Flow Shop with Alternative Machines

Watcharapan Sukkerd, Teeradej Wuttipornpun

Abstract—This paper aims to present non-population search algorithms called tabu search (TS), simulated annealing (SA) and variable neighborhood search (VNS) to minimize the total cost of capacitated MRP problem in multi-stage assembly flow shop with two alternative machines. There are three main steps for the algorithm. Firstly, an initial sequence of orders is constructed by a simple due date-based dispatching rule. Secondly, the sequence of orders is repeatedly improved to reduce the total cost by applying TS, SA and VNS separately. Finally, the total cost is further reduced by optimizing the start time of each operation using the linear programming (LP) model. Parameters of the algorithm are tuned by using real data from automotive companies. The result shows that VNS significantly outperforms TS, SA and the existing algorithm.

Keywords—Capacitated MRP, non-population search algorithms, linear programming, assembly flow shop.

I. INTRODUCTION

ENTERPRISE resource planning (ERP) is a powerful system to manage business activities throughout the supply chain. Unfortunately, a planning tool of the ERP system called material requirement planning (MRP) is reported that it generates a capacity problem on shop floor [1]-[3]. A reason for this is that MRP assumes infinite resource capacity or constant lead-time [4]-[6]. This problem is then later called capacitated MRP. Since the capacitated MRP problem for industrial scale instances is normally the NP-hard class, the metaheuristic algorithm is one of the appropriate approaches to solve the problem [7]-[9].

There are two concepts of the metaheuristic algorithms developed for the capacitated MRP problem as shown in Table I. The first concept is called population search algorithm. It includes genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO) and cuckoo search (CS). The second concept is called non-population search algorithm. It includes TS, iterated local search (ILS), VNS, and SA [7]-[33].

Based on the literature as shown in Table I, the population search algorithm seems to be more popular 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 three non-population search algorithms, which are TS, SA and VNS to solve the industrial scale capacitated MRP problem. Our objective is to improve the solution obtained from the existing capacitated MRP algorithm by [6]. The presented algorithm is intently developed for multi-stage flow shop with two alternative machines. The planning horizon is one month without overtimes and preemptive options.

The remaining of the paper is organized as follows. Section II deals with details of the presented algorithm. The experiments for parameter tuning for our case studies are explained in Section III. Results and discussions are provided in Section IV. Finally, the conclusion of this paper and recommendations for future research are given in Section V.

II. DETAILS OF PRESENTED ALGORITHM

The presented algorithm has three main steps as shown in Fig. 1. They are explained as follows.

1. Construct Initial Sequence of Orders
2. Improve the Initial Sequence by Non-Population Search Algorithms
3. Optimize the Start Times of Operations

Fig. 1 Pseudo code of the proposed algorithm

Step 1: Construct Initial Sequence of Orders

The objective of this step is to construct an initial sequence of orders by a simple dispatching rule called minimum slack time (MST). This rule schedules the order with the MST first, and schedules the order with the relatively long slack time later (see [6] for details of applying this rule). The reason to select this rule is that it obtains a good performance for the existing capacitated MRP algorithm. Thus, applying this rule to both algorithms, a fair comparison can be made.
TABLE I
LITERATURE REVIEW

<table>
<thead>
<tr>
<th>Approach</th>
<th>References</th>
<th>Characteristics of problem</th>
<th>Objectives with minimizing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chang et al., 2013 [8]</td>
<td>Permutation flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td>PSO</td>
<td>Zhang et al., 2008 [10]</td>
<td>Flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>Eddy et al., 2016 [12]</td>
<td>Blocking flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>Yagmahan and Yenisey, 2010 [14]</td>
<td>Flow shop</td>
<td>Makespan and total flowtime</td>
</tr>
<tr>
<td></td>
<td>Zhang and Jing, 2012 [15]</td>
<td>Permutation flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td>CS</td>
<td>Marichelvam et al., 2014 [16]</td>
<td>Hybrid flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>Dasgupta and Das, 2015 [17]</td>
<td>Flow shop</td>
<td>Makespan and mean flow time</td>
</tr>
<tr>
<td></td>
<td>Wang et al., 2016 [18]</td>
<td>Flow shop scheduling</td>
<td>Makespan</td>
</tr>
<tr>
<td>Non-population search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS</td>
<td>Ekşioğlu et al., 2008 [19]</td>
<td>Flow shop scheduling</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>Chen, Pan and Wu, 2009 [21]</td>
<td>Permutation flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>Liao and Huang, 2011 [22]</td>
<td>Two machine flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td>ILS</td>
<td>Dong, Chen and Huang, 2011 [23]</td>
<td>Permutation flow shop</td>
<td>Total flow time</td>
</tr>
<tr>
<td></td>
<td>Dong et al., 2013 [24]</td>
<td>Permutation flow shop</td>
<td>Total flow time</td>
</tr>
<tr>
<td></td>
<td>Wang et al., 2014 [25]</td>
<td>Permutation flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td>VNS</td>
<td>Ribas et al., 2013 [26]</td>
<td>Blocking flow shop</td>
<td>Total tardiness</td>
</tr>
<tr>
<td></td>
<td>Li, Pan and Wang, 2014 [27]</td>
<td>Hybrid flow shop</td>
<td>Maximum completion time</td>
</tr>
<tr>
<td></td>
<td>Moslehi and Khorasanian, 2014 [28]</td>
<td>Permutation flow shop</td>
<td>Makespan</td>
</tr>
<tr>
<td></td>
<td>M’Hallah, 2014 [29]</td>
<td>Permutation flow shop</td>
<td>Earliness and tardiness</td>
</tr>
<tr>
<td></td>
<td>Lei, 2015 [30]</td>
<td>Flow shop</td>
<td>Makespan of the first and the total tardiness of the second agent</td>
</tr>
<tr>
<td>SA</td>
<td>Jungwattanakit et al., 2009 [31]</td>
<td>Flexible flow shop</td>
<td>Sum of makespan and the number of tardy jobs</td>
</tr>
<tr>
<td></td>
<td>Jaroslaw et al., 2013 [33]</td>
<td>Flow shop</td>
<td>Makespan and the sum of tardiness</td>
</tr>
<tr>
<td></td>
<td>Nikzad et al., 2015 [34]</td>
<td>Flexible flow shop</td>
<td>Maximum completion time</td>
</tr>
</tbody>
</table>

overall procedure: TS algorithm

input: orders, BOMs and machines information, TS parameters
output: solution from the TS algorithm

begin
create $S_i$ as a sequence of orders
while (not terminating condition of TS) do
generate neighbourhood sequences from $S_i$ and check with TL_i;
evaluate the neighbourhood sequences;
select the improved sequence and update TL_i;
end;
output: solution from the TS algorithm
end;

Fig. 2 TS algorithm

Step 2: Improve the Initial Sequence by Non-population Search Algorithms

This step tries to improve the sequence of orders obtained from the first step by applying non-population search algorithms. Three non-population search called TS, SA and VNS are proposed and their conventional mechanisms are shown in Fig. 2, 3 and 4, respectively. The fitness function ($FIT$) is shown in (1). Let $i$ be an order index from 1 to $n$, $Q_i$, $t_e$, $e_i$, $f_i$ be the order quantity, tardiness, earliness and flow-time of order $i$. $T_i$, $E_i$, $F_i$ be the cost per unit of tardiness, earliness and flow-time of order $i$.

$$FIT = \sum_{i=1}^{n} T_i Q_i f_i + \sum_{i=1}^{n} E_i Q_i e_i + \sum_{i=1}^{n} F_i Q_i f_i$$ (1)

To obtain the $FIT$, a sequence of orders of each iteration is exploded by variable lead-time MRP (VMRP) in order to determine details of operations. These operations are then scheduled to less tardiness machines by forward with permutation scheduling. After that the tardiness, earliness and flow-time of each order is calculated and finally the $FIT$ value can be obtained. To illustrate how this step works, details of the required operations of the four orders after VMRP explosion shown in Fig. 5 are used. Suppose that the sequence of orders of an iteration of VNS is $O_1$ $O_2$ $O_3$ $O_4$. All operations of order $O_1$ are scheduled to less tardiness machines first, and the operations of order $O_2$ are scheduled next and so on. The result is shown in Fig. 6. It is obvious that all machines have the same sequence of operations complying with the permutation sequencing concept.
### Overall procedure: SA algorithm

**Input:** orders, BOMs and machines information, SA parameters  
**Output:** solution from the SA algorithm

1. Create S\_i as a sequence of orders.  
2. Set T\_i // T: initial temperature. 
3. **While** (not terminating condition of SA) do  
   a. Generate neighborhood sequences from S\_i.  
   b. Evaluate the neighborhood sequences.  
   c. Select the improved sequence. 
   d. If (improved sequence < S)  
      i. Accept improved sequence as S with probability. 
   e. Update T\_i. 
4. **End**

**Output:** solution from the SA algorithm

---

### Overall procedure: VNS algorithm

**Input:** orders, BOMs and machines information, VNS parameters  
**Output:** solution from the VNS algorithm

1. Create S\_i as a sequence of orders.  
2. **While** (not terminating condition of VNS) do  
   a. Perturbation S\_i generates neighborhood sequences from S\_i.  
   b. Evaluate neighborhood sequences.  
   c. Select the improved sequence. 
   d. If (improved sequence < S)  
      i. Accept improved sequence as S\_i. 
   e. **End**
3. **End**

**Output:** solution from the VNS algorithm

---

### Example of BOMs, machines and operations after VMRP explosion

<table>
<thead>
<tr>
<th>Order</th>
<th>Q</th>
<th>T</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_1</td>
<td>12</td>
<td>4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>O_2</td>
<td>10</td>
<td>5</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>O_3</td>
<td>14</td>
<td>5</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>O_4</td>
<td>8</td>
<td>2</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**All ratios of Parent : Child**

- **Order O\_1**
  - Q\_1 = 12 pcs
  - T\_1 = 4\$/pcs/day
  - E\_1 = 0.2\$/pcs/day
  - F\_1 = 0.1\$/pcs/day

- **Order O\_2**
  - Q\_2 = 10 pcs
  - T\_2 = 5\$/pcs/day
  - E\_2 = 0.2\$/pcs/day
  - F\_2 = 0.1\$/pcs/day

- **Order O\_3**
  - Q\_3 = 14 pcs
  - T\_3 = 2.5\$/pcs/day
  - E\_3 = 0.2\$/pcs/day
  - F\_3 = 0.05\$/pcs/day

---

**Fig. 3 SA algorithm**

<table>
<thead>
<tr>
<th>Order</th>
<th>Q</th>
<th>T</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_1</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>O_2</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>O_3</td>
<td>7</td>
<td>6</td>
<td>16</td>
<td>9</td>
</tr>
</tbody>
</table>

**Fig. 4 VNS algorithm**

<table>
<thead>
<tr>
<th>Order</th>
<th>Q</th>
<th>T</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_1</td>
<td>8</td>
<td>8</td>
<td>30</td>
<td>22</td>
</tr>
<tr>
<td>O_2</td>
<td>6</td>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>O_3</td>
<td>7</td>
<td>7</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

**Fig. 5 Example of BOMs, machines and operations after VMRP explosion**
Step 3: Optimize the Start Times of Operations

This step tries to optimize the start times of operations on the selected machines obtained from step 2 by the LP model. Before our formulation, the orders are renumbered, where the first order in the sequence has \( i = 1 \), and the second order in the sequence has \( i = 2 \), and so on. The problem is formulated by using indices, parameters, variables and decision variables 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 which is already specified from step 2

Parameters
\( p_{i,j,k} \) = processing time of order \( i \) operation \( j \) on machine \( k \)  
\( d_i \) = due time of order \( i \)  
\( T_i \) = tardiness cost per unit of order \( i \)  
\( E_i \) = earliness cost per unit of order \( i \)  
\( F_i \) = flow-time cost per unit of order \( i \)  
\( Q_i \) = quantity of order \( i \)

Variables
\( c_i \) = completion time of order \( i \)  
\( t_i \) = tardiness of order \( i \)  
\( e_i \) = earliness of order \( i \)  
\( f_i \) = flow-time of order \( i \)  
\( Z \) = total cost

Decision Variable
\( x_{i,j} \) = start time of order \( i \) operation \( j \)

Objective Function:
\[
\text{Minimize } Z = \sum_{i=1}^{n} T_i c_i + \sum_{i=1}^{n} E_i e_i + \sum_{i=1}^{n} F_i f_i 
\]

Subject to:
To facilitate how to construct the constraints for our formulation, the schedule shown in Fig. 6 is used.

a) Constraints to Maintain the Sequence of Operations of Each Machine

There are four operations on machine 3 (\( O_{1,3}, O_{2,3}, O_{2,2} \) and \( O_{4,2} \)), and three operations on machine 4 (\( O_{1,4}, O_{3,3} \) and \( O_{4,3} \)). These sequences can be constructed by (3)-(7). Note that the sequence of operations for our case studies can be further developed based on this idea.

For Machine 3:
\[
\begin{align*}
x_{4,2} &\geq x_{2,2} + p_{2,2,3} \\
x_{2,2} &\geq x_{3,2} + p_{3,2,3} \\
x_{3,2} &\geq x_{1,2} + p_{1,2,3}
\end{align*}
\]

For Machine 4:
\[
\begin{align*}
x_{4,3} &\geq x_{3,3} + p_{3,3,4} \\
x_{3,3} &\geq x_{1,3} + p_{1,3,4}
\end{align*}
\]

b) Constraint to Maintain the Precedence Relationships of Operations of an Order

\[
x_{j,k} + p_{j,k} \geq t_{j^*} \quad \forall i, \forall k; \quad j=1,2,3,...,m-1
\]

c) Constraints for Completion Time, Tardiness, Earliness and Flow-Time

The completion time of an order can be constructed by (9):
\[
c_i = x_{i,j^*} + p_{i,j^*,k} \quad \forall i; \forall j^*; \forall k
\]

The tardiness of an order can be constructed by (10):
\[
t_i = \begin{cases} 
0, & \text{if } c_i \leq d_i \\
(c_i - d_i), & \text{otherwise} 
\end{cases} \quad \forall i
\]

The earliness of an order can be constructed by (11):
\[
e_i = \begin{cases} 
0, & \text{if } c_i \geq d_i \\
(d_i - c_i), & \text{otherwise} 
\end{cases} \quad \forall i
\]

The flow-time of an order can be constructed based on bill of materials shown in Fig. 5. The flow-times of the four orders...
are shown in (12)-(18). Note that the flow-time for our case studies can be further developed based on this idea.

$$f_1 \geq c_1 - x_{1,j}$$

$$f_1 \geq c_1 - x_{1,3}$$

$$f_2 \geq c_2 - x_{2,1}$$

$$f_2 \geq c_2 - x_{2,2}$$

$$f_3 = c_3 - x_{3,1}$$

$$f_4 \geq c_4 - x_{4,1}$$

$$f_4 \geq c_4 - x_{4,2}$$

$d)$ The Non-Negativity of Decision Variable

$$x_{i,j} \geq 0$$

The result after applying the LP model is shown in Fig. 7. It is obviously seen that the total cost before applying the LP model is dramatically reduced to $388.20.

III. CASE STUDIES AND EXPERIMENTS

This section consists of two parts. The first part deals with the details of case studies. The second part deals with an experiment for parameters tuning of TS, SA and VNS. They all are explained as follows.

A. Details of Case Studies

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

a) The production shop is a multi-stage assembly flow shop with alternative machines.
b) The planning horizon is one month.
c) No preemptive schedule and overtime.

B. Experiments for Parameters Tuning of TS, SA and VNS

The required parameters for TS, SA and VNS are summarized in Table III. For the parameter tuning, 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 total cost. The stopping criterion for each run is 2 hours. Note that data in Table III are obtained by a set of screening experiments.

<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>Number of order quantities</td>
<td>1,512</td>
<td>1,735</td>
<td>1,935</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 work centres</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

IV. RESULTS AND DISCUSSION

Based on the parameter tuning experiment, the best setting for each case study is varied depending on the different characteristics of case studies. Therefore, a common setting for all case studies is more useful for the planner. To determine the best common setting, the relative percentage deviation (RPD) and its average (ARPD) are used. Both of them are calculated by (20) and (21), where $TC_{ALL}$ is the total cost obtained from each run, $TC_{BEST}$ 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 the minimum ARPD since it guarantees that the total cost obtained from this setting is very close to its best total cost.

$$RPD_v = \frac{TC_{ALL} - TC_{ALG}}{TC_{ALG}} \times 100\%$$

$$ARPD = \frac{1}{V} \sum_{v=1}^{V} (RPD_v)$$
The best common setting shown in Table IV is the setting at the minimum ARPD. By this setting, it guarantees that the total cost from this setting is very close to its best total cost (near best). Table V shows the best total cost, near best total cost and RPD at minimum ARPD. Based on ARPD, it can be seen that TS obtains the smallest deviations, whereas SA obtains the highest deviation. However, the deviation gap is less than 1%, which means that the best common setting concept is very efficient. It also observes that VNS outperforms TS and SA for both the best and near best total costs.

Fig. 8 shows the total cost development characteristics of the proposed algorithm. It can be seen that TS and SA reach their steady state very fast, while VNS reaches its steady state slower. However, the planner should wait a slightly longer to obtain a significantly better solution.

To further improve the total costs, the LP model is applied to all solutions from the best common settings. The improved total costs are shown in Table VI. It is obvious that the near best total costs shown in Table V are substantially reduced. This proves that the LP model is very efficient. To compare the improved total costs of TS, SA and VNS with the best total cost of the existing algorithm, the relative percentage improvement over the best total cost of the existing algorithm (RPI) is used. It is calculated from (22), where $MTC_{ALG}$ is the
improved total cost of each algorithm, TCEX, is the best total cost from the existing algorithm. Based on the RPI value, VNS significantly outperforms other algorithms. VNS improves the total cost of the existing algorithm about 29.18% on average, while TS and SA improve the total cost from of the existing algorithm about 22.95% and 20.03%, respectively. 

\[ \text{RPI} = \frac{\text{TCEX} - \text{TC}}{\text{TC}} \times 100\% \]  

(22)

V. CONCLUSIONS AND FUTURE RESEARCH

In this study, the non-population search algorithms called TS, SA and VNS are presented to solve the industrial scale capacitated MRP problem to minimize the total cost. There are two mechanisms of the proposed algorithm. Firstly, the total cost is improved by non-population search algorithms. Secondly, this total cost is further improved by applying the LP model. The performance of the presented algorithm is evaluated by many industrial scale instances in order to ensure that it can be implemented to various industrial situations. The result shows that VNS obtains the best total cost. It reduces the total cost obtained from the existing algorithm almost 30% on average.

There are some interesting research gaps for future investigations as: (1) Performance evaluation of population search algorithm, (2) effectiveness comparison between non-population and population search algorithms, (3) modifications of conventional non-population and population search algorithms to make them faster and better, (4) implementation of the proposed algorithm to other manufacturing shops such as job shop and open shop, (5) time extended decision and other lot-sizing methods.

ACKNOWLEDGMENT

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

REFERENCES


International Scholarly and Scientific Research & Innovation 11(3) 2017 653

ISSN:000000091950263


