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.

---

overall procedure: the proposed algorithm
input: orders, BOMs and machines information
output: solution from the proposed algorithm
begin
//**Step 1**: Construct initial sequence of orders
          generate an initial sequence by a dispatching rule;

//**Step 2**: Improve the initial sequence by non-population search algorithms
          apply TS, SA, VNS

//**Step 3**: Optimize the start times of operations
          apply LP to the schedule from step 3;

output: solution from the existing algorithm
end;

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.

Watcharapan Sukkerd and Teeradej Wuttipornpun are with the Department of Industrial Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand (email: s.watcharapan@gmail.com, teeradejw@kmutnb.ac.th).

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

TABLE I
LITERATURE REVIEW

Approach	References	Characteristics of problem	Objectives with minimizing
		Population search	
GA	Zandieh and Karimi, 2011 [7]	Hybrid flexible flow shop	Total weighted tardiness and maximum completion time
	Chang et al., 2013 [8]	Permutation flow shop	Makespan
	Pang, 2013 [9]	Flow shop	Maximum lateness
PSO	Zhang et al., 2008 [10]	Flow shop	Makespan
	Tang and Wang, 2010 [11]	Hybrid flow shop	Total weighted completion time
	Eddaly et al., 2016 [12]	Blocking flow shop	Makespan
ACO	Gajpal and Rajendran, 2006 [13]	Flow shop	Completion time
	Yagmahan and Yenisey, 2010 [14]	Flow shop	Makespan and total flowtime
	Zhang and Jing, 2012 [15]	Permutation flow shop	Makespan
CS	Marichelvam et al., 2014 [16]	Hybrid flow shop	Makespan
	Dasgupta and Das, 2015 [17]	Flow shop	Makespan and mean flow time
	Wang et al., 2016 [18]	Flow shop scheduling	Makespan
		Non-population search	
TS	Ekşioğlu et al., 2008 [19]	Flow shop scheduling	Makespan
	Wang and Tang, 2009 [20]	Hybrid flow shop	Sum of weighted completion time
	Chen, Pan and Wu, 2009 [21]	Permutation flow shop	Makespan
	Liao and Huang, 2011 [22]	Two machine flow shop	Makespan
ILS	Dong, Chen and Huang, 2011 [23]	Permutation flow shop	Total flow time
	Dong et al., 2013 [24]	Permutation flow shop	Total flow time
	Wang et al., 2014 [25]	Permutation flow shop	Makespan
	Ribas et al., 2013 [26]	Blocking flow shop	Total tardiness
VNS	Li, Pan and Wang, 2014 [27]	Hybrid flow shop	Maximum completion time
	Moslehi and Khorasanian, 2014 [28]	Permutation flow shop	Makespan
	M'Hallah, 2014 [29]	Permutation flow shop	Earliness and tardiness
	Lei, 2015 [30]	Flow shop	Makespan of the first and the total tardiness of the second agent
SA	Jungwattanakit et al., 2009 [31]	Flexible flow shop	Sum of makespan and the number of tardy jobs
	Zhang and Wu, 2011 [32]	Job shop	Total weighted tardiness
	Jarosław et al., 2013 [33]	Flow shop	Makespan and the sum of tardiness
	Nikzad et al., 2015 [34]	Flexible flow shop	Maximum completion time

```
overall procedure: TS algorithm
input: orders, BOMs and machines information, TS parameters
output: solution from the TS algorithm
begin
        create S; //S: a sequence of orders
        while (not terminating condition of TS) do
            generate neighbourhood sequences from S and check with TL;
            evaluate the neighbourhood sequences;
            select the improved sequence and update TL;
            S←improved sequence;
        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_i, 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.

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

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 $O_1 \Rightarrow O_3 \Rightarrow O_2 \Rightarrow O_4$. All operations of order O_1 are scheduled to less tardiness machines first, and the operations of order O_3 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.

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

overall procedure: SA algorithm
input: orders, BOMs and machines information, SA parameters
output: solution from the SA algorithm
begin

    create S; //S: a sequence of orders

    set T; // T: initial temperature

    while (**not** terminating condition of SA) **do**
        generate neighbourhood sequences from S;
        evaluate the neighbourhood sequences;
        select the improved sequence;
        if (improved sequence < S)
            $S \leftarrow$ improved sequence;
        else
            accept improved sequence as S with probability;
        end;
        update T;
    end;

output: solution from the SA algorithm
end;

Fig. 3 SA algorithm

overall procedure: VNS algorithm
input: orders, BOMs and machines information, VNS parameters
output: solution from the VNS algorithm
begin

    create S; //S: a sequence of orders

    while (**not** terminating condition of VNS) **do**
    $NS = 1$; //NS: neighbourhood structures
        while (**not** terminating condition of NS) **do**
        $S' \leftarrow$ perturbation S;
        generate neighbourhood sequences from S';
        evaluate neighbourhood sequences;
        select the improved sequence;
        if (improved sequence < S)
            $S \leftarrow$ improved sequence;
            $NS = 1$;
        else
            $NS = NS + 1$;
        end;
        end;
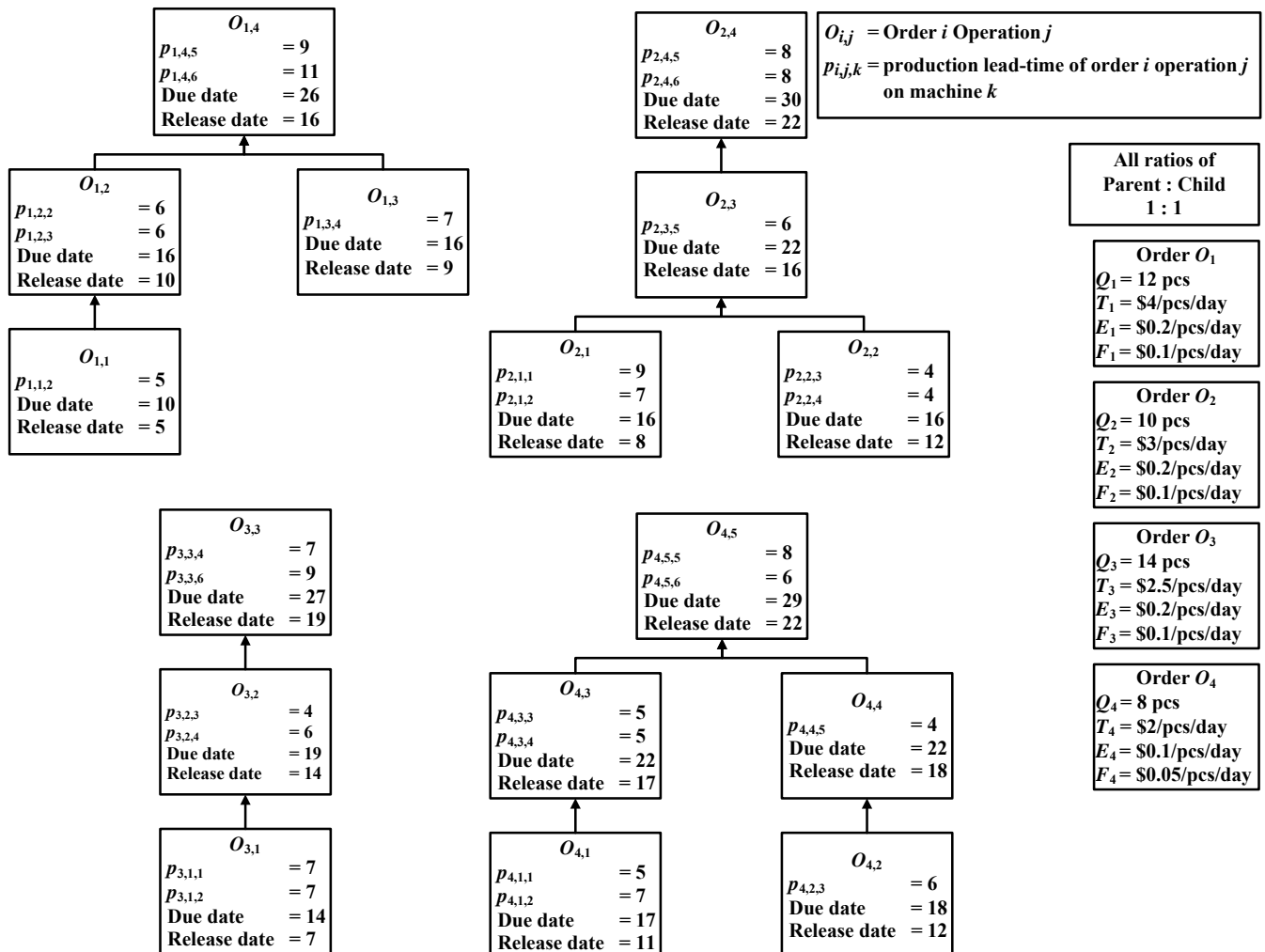   **end;**

output: solution from the VNS algorithm
end;

Fig. 4 VNS algorithm

$O_{1,4}$
$p_{1,4,5}$ = 9
$p_{1,4,6}$ = 11
Due date = 26
Release date = 16

$O_{1,2}$
$p_{1,2,2}$ = 6
$p_{1,2,3}$ = 6
Due date = 16
Release date = 10

$O_{1,3}$
$p_{1,3,4}$ = 7
Due date = 16
Release date = 9

$O_{1,1}$
$p_{1,1,2}$ = 5
Due date = 10
Release date = 5

$O_{3,3}$
$p_{3,3,4}$ = 7
$p_{3,3,6}$ = 9
Due date = 27
Release date = 19

$O_{3,2}$
$p_{3,2,3}$ = 4
$p_{3,2,4}$ = 6
Due date = 19
Release date = 14

$O_{3,1}$
$p_{3,1,1}$ = 7
$p_{3,1,2}$ = 7
Due date = 14
Release date = 7

$O_{2,4}$
$p_{2,4,5}$ = 8
$p_{2,4,6}$ = 8
Due date = 30
Release date = 22

$O_{2,3}$
$p_{2,3,5}$ = 6
Due date = 22
Release date = 16

$O_{2,1}$
$p_{2,1,1}$ = 9
$p_{2,1,2}$ = 7
Due date = 16
Release date = 8

$O_{2,2}$
$p_{2,2,3}$ = 4
$p_{2,2,4}$ = 4
Due date = 16
Release date = 12

$O_{4,5}$
$p_{4,5,5}$ = 8
$p_{4,5,6}$ = 6
Due date = 29
Release date = 22

$O_{4,3}$
$p_{4,3,3}$ = 5
$p_{4,3,4}$ = 5
Due date = 22
Release date = 17

$O_{4,4}$
$p_{4,4,5}$ = 4
Due date = 22
Release date = 18

$O_{4,1}$
$p_{4,1,1}$ = 5
$p_{4,1,2}$ = 7
Due date = 17
Release date = 11

$O_{4,2}$
$p_{4,2,3}$ = 6
Due date = 18
Release date = 12

$O_{i,j}$ = Order i Operation j
$p_{i,j,k}$ = **production lead-time of order i operation j on machine k**

All ratios of Parent : Child
1 : 1

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 = \$3/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 = \$0.1/pcs/day

Order O_4
Q_4 = 8 pcs
T_4 = \$2/pcs/day
E_4 = \$0.1/pcs/day
F_4 = \$0.05/pcs/day

Fig. 5 Example of BOMs, machines and operations after VMRP explosion

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
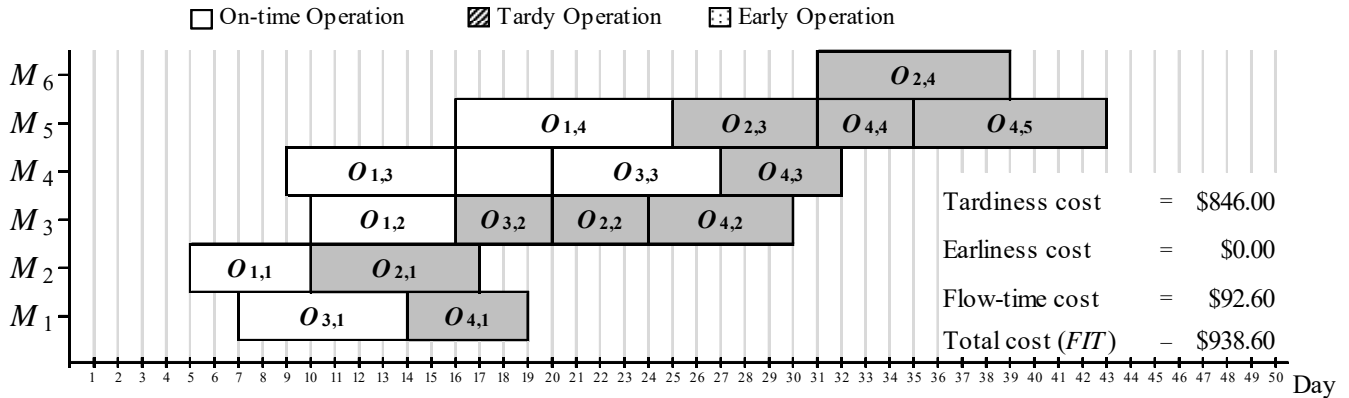Vol:11, No:3, 2017

Fig. 6 Results of the proposed algorithm before LP

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:

$$\text{Minimize } Z = \sum_{i=1}^{n} T_i Q_i t_i + \sum_{i=1}^{n} E_i Q_i e_i + \sum_{i=1}^{n} F_i Q_i f_i \quad (2)$$

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 ($O_{1,2}$, $O_{3,2}$, $O_{2,2}$ and $O_{4,2}$), and three operations on machine 4 ($O_{1,3}$, $O_{3,3}$ 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:

$$x_{4,2} \geq x_{2,2} + p_{2,2,3} \quad (3)$$

$$x_{2,2} \geq x_{3,2} + p_{3,2,3} \quad (4)$$

$$x_{3,2} \geq x_{1,2} + p_{1,2,3} \quad (5)$$

For Machine 4:

$$x_{4,3} \geq x_{3,3} + p_{3,3,4} \quad (6)$$

$$x_{3,3} \geq x_{1,3} + p_{1,3,4} \quad (7)$$

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

$$x_{i,j+1} \geq x_{i,j} + p_{i,j,k} \quad \forall i; \forall k; \ j=1, 2, 3, .., m{-}1 \quad (8)$$

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

The completion time of an order can be constructed by (9):

$$c_i = x_{i,j^*} + p_{i,j^*,k} \quad \forall i; \ \forall j^*; \ \forall k; \quad (9)$$

The tardiness of an order can be constructed by (10):

$$t_i = \begin{cases} 0, & \text{if } c_i \leq d_i \\ c_i - d_i, & \text{otherwise} \end{cases} \quad \forall i \quad (10)$$

The earliness of an order can be constructed by (11):

$$e_i = \begin{cases} 0, & \text{if } c_i \geq d_i \\ d_i - c_i, & \text{otherwise} \end{cases} \quad \forall i \quad (11)$$

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

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

are shown in (12)-(18). Note that the flow-time for our case studies can be further developed based on this idea.
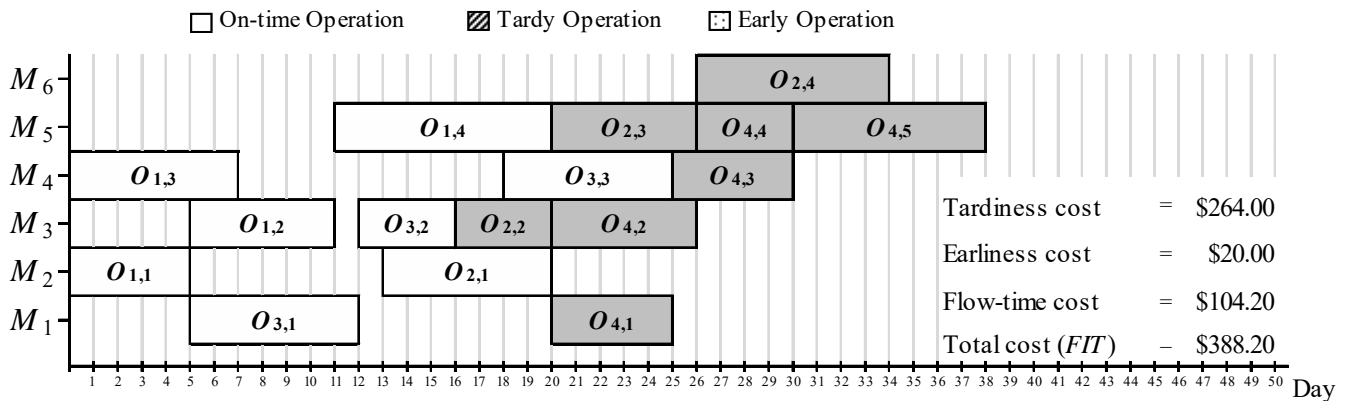

Fig. 7 Results of the proposed algorithm after applying the LP model

$$f_1 \geq c_1 - x_{1,1} \tag{12}$$

$$f_1 \geq c_1 - x_{1,3} \tag{13}$$

$$f_2 \geq c_2 - x_{2,1} \tag{14}$$

$$f_2 \geq c_2 - x_{2,2} \tag{15}$$

$$f_3 = c_3 - x_{3,1} \tag{16}$$

$$f_4 \geq c_4 - x_{4,1} \tag{17}$$

$$f_4 \geq c_4 - x_{4,2} \tag{18}$$

d) The Non-Negativity of Decision Variable

$$x_{i,j} \geq 0 \tag{19}$$

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 II
DIFFERENT CHARACTERISTICS OF CASE STUDIES

Characteristics	Case 1	Case 2	Case 3
Number of finished products	16	19	18
Number of order quantities	1,512	1,735	1,986
Min/Max levels in bill of materials	2/5	3/5	4/6
Number of operations	520	610	690
Number of work centres	18	20	20

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_{ALG} 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.

$$RPD_v = \frac{TC_{ALG(v)} - TC_{BEST(v)}}{TC_{BEST(v)}} \times 100\% \tag{20}$$

$$ARPD = \frac{1}{V} \sum_{v=1}^{V} (RPD_{TC})_v \tag{21}$$

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

TABLE III
PARAMETERS FOR TS, SA AND VNS

Algorithm	No. of NBH sequences	NBH operators	TL size	No. of iterations for each temperature	Initial temperature	Alpha	Shaking operators
TS	10, 15	SWAP, INSERT	30, 40, 50	-	-	-	-
SA	5, 10	SWAP, INSERT	-	1, 10	80, 100	0.8, 0.9	-
VNS	70, 90	SWAP and INSERT	-	-	-	-	SWAP, INSERT

TABLE IV
BEST COMMON SETTINGS FOR ALL CASE STUDIES AT MINIMUM *ARPD*

Algorithm	No. of NBH sequences	NBH operators	TL size	No. of iterations of each temperature	Initial temperature	Alpha	Shaking operators
TS	10	SWAP	30	-	-	-	-
SA	5	INSERT	-	1	100	0.9	-
VNS	70	SWAP and INSERT	-	-	-	-	SWAP

Fig. 8 Solution development characteristics

TABLE V
VALUES OF BEST TOTAL COSTS, NEAR BEST TOTAL COSTS AND *RPD* AT MINIMUM *ARPD*

Algorithms	Best total costs ($)/Near best total costs ($)/*RPD*			Min. *ARPD*
	Case 1	Case 2	Case 3	
TS	8,247 / 8,277 / 0.36%	13,470 / 13,534 / 0.47%	17,971 / 18,027 / 0.31%	0.38%
SA	10,082 / 10,082 / 0.00%	15,162 / 15,589 / 2.82%	19,965 / 19,965 / 0.00%	0.94%
VNS	7,824 / 7,883 / 0.75%	12,688 / 12,825 / 1.08%	16,996 / 16,996 / 0.00%	0.61%

TABLE VI
IMPROVED TOTAL COSTS AND *RPI* AFTER APPLYING LP

Algorithms	Improved total costs ($)/*RPI*			
	Case 1	Case 2	Case 3	Average
Existing FCMRP (2014)	9,424 (Worst)	16,016 (Worst)	19,280 (Worst)	14,906.58 (Worst)
TS	7,540 / 19.99% (2)	12,480 / 22.07% (2)	14,116 / 26.79% (2)	11,378.91 / 22.95% (2)
SA	8,305 / 11.87% (3)	12,770 / 20.27% (3)	13,894 / 27.94% (3)	11656.33 / 20.03% (3)
VNS	6,629 / 29.66% (1)	11,476 / 28.35% (1)	13,583 / 29.55% (1)	10562.67 / 29.18% (1)

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 *IMTC_ALG* is the

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

improved total cost of each algorithm, TC_{EX} 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.

$$RPI = \frac{|IMTC_{ALG} - TC_{EX}|}{TC_{EX}} \text{ x } 100\% \qquad (22)$$

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.

REFERENCES

[1] P. B. Nagendra, and S. K. Das, "Finite capacity scheduling method for MRP with lot size restrictions," *International Journal of Production Research*, vol. 39, pp. 1603-1623, 2001.
[2] A. M. Örnek, and O. Cengiz, "Capacitated lot sizing with alternative routings and overtime decisions," *International Journal of Production Research*, vol. 44, no. 24, pp. 5363–5389, 2006.
[3] C. Öztürk, and A. M. Örnek, "A MIP based heuristic for capacitated MRP systems," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 926–942, 2012.
[4] N. A. Bakke, and R. Hellberg, "The challenges of capacity planning," *International Journal of Production Economics*, vol. 30-31, no.1, pp. 243-264, 1993.
[5] S-H. Lee, S. Trimi, D. Choi, and J. S. Rha, "A comparative study of proprietary ERP and open source ERP modules on the value chain," *International Journal of Information and Decision Sciences*, vol. 3, no. 1, pp. 26-38, 2011.
[6] T. Wuttipornpun, and P. Yenradee, "Finite capacity material requirement planning system for assembly flow shop with alternative work centres," *International Journal of Industrial & Systems Engineering*, vol. 18, no. 1, pp. 95-124, 2014.
[7] M. Zandieh, and N. Karimi, "An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times," *Journal of Intelligent Manufacturing*, vol. 22 no. 6, pp. 979-989, 2011.
[8] P-C. Chang, W-H. Huang, J-L. Wu, and T. C. E. Cheng, "A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem," *International Journal of Production Economics*, vol. 141 no. 1, pp. 45-55, 2013.
[9] K-W. Pang, "A genetic algorithm based heuristic for two machine no-wait flowshop scheduling problems with class setup times that minimizes maximum lateness," *International Journal of Production Economics*, vol. 141 no. 1, pp. 127-136, 2013.
[10] C. Zhang, J. Sun, X. Zhu, and Q. Yang, "An improved particle swarm optimization algorithm for flowshop scheduling problem," *Information Processing Letters*, vol. 108 no. 4, pp. 204-209, 2008.
[11] L. Tang, and X. Wang, "An Improved Particle Swarm Optimization Algorithm for the Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time in Process Industry," *Transactions on Control Systems Technology*, vol. 18 no. 6, pp. 1303-1313, 2010.
[12] M. Eddaly, B. Jarboui, and P. Siarry, "Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem," *Journal of Computational Design and Engineering*, vol. 3 no. 4, pp. 295-311, 2016
[13] Y. Gajpal, and C. Rajendran, "An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshop," International Journal of Production Economics, vol. 101 no. 2, pp. 259-272, 2006.
[14] B. Yagmahan, and M. M. Yenisey, "A multi-objective ant colony system algorithm for flow shop scheduling problem," *Expert Systems with Applications*, vol. 37 no. 2, pp. 1361-1368, 2010.
[15] Z. Zhang, and Z. Jing, "An improved ant colony optimization algorithm for permutation flow shop scheduling to minimize makespan," *13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 605-609, 2012.
[16] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Applied Soft Computing*, vol. 19, pp. 93-101, 2014.
[17] P. Dasgupta, and S. Das, "A Discrete Inter-Species Cuckoo Search for flowshop scheduling problems," *Computers & Operations Research*, vol. 60, pp. 111-120, 2015.
[18] H. Wang, W. Wang, H. Sun, Z. Cui, S. Rahnamayan, and S. Zeng, "A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems," *Soft Computing, Springer-Verlag Berlin Heidelberg*, pp. 1-11, 2016.
[19] B. Ekşioğlu, S. D. Ekşioğlu, and P. Jain, "A tabu search algorithm for the flowshop scheduling problem with changing neighborhoods," *Computers & Industrial Engineering*, vol. 54, no. 1, pp. 1-11, 2008.
[20] X. Wang, and L. Tang, "A tabu search heuristic for the hybrid flow shop scheduling with finite intermediat buffers," *Computers & Operations Research*, vol. 36, no. 3, pp. 907-918, 2009.
[21] J-S. Chen, J. C-H. Pan, and C-K. Wu, "Hybrid tabu search for re-entrant permutation flow-shop scheduling problem," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1924-1930, 2008.
[22] L-M. Liao, and C-J. Huang, "Tabu search heuristic for two-machine flowshop with batch processing machines," *Computers & Industrial Engineering*, vol. 60, no. 3, pp. 426-432, 2011.
[23] X. Dong, P. Chen, and H. Huang, "An Improved Iterated Local Search Algorithm for the Permutation Flowshop Problem with Total Flowtime," *Advance in Automation and Robotics*, Springer Berlin Heidelberg, vol. 1, pp. 41-48, 2011.
[24] X. Dong, P. Chen, H. Huang, and M. Nowak, "A multi-restart iterated local search algorithm for the permutation flow shop problem minimizing total flow time," *Computers & Operations Research*, vol. 40, no. 2, pp. 627-632, 2013.
[25] Y. Wang, X. Dong, P. Chen, and Y. Lin, "Iterated Local Search Algorithms for the Sequence-Dependent Setup Times Flow Shop Scheduling Problem Minimizing Makespan," *Foundation of Intelligent Systems*, vol. 277, pp. 329-338, 2014.
[26] I. Ribas, R. Companys, and X. Tort-Martorell, "An efficient iterated local search algorithm for the total tardiness blocking flow shop problem," *International Journal of Production Research*, vol.51, no. 17, pp. 5238-5252, 2013.
[27] J-Q. Li, Q-K. Pan, and F-T. Wang, "A hybrid variable neighborhood search for solving the hybrid flowshop scheduling problem," *Applied Soft Computing*, vol. 24, pp. 63-77, 2014.
[28] G. Moslehi, and D. Khorasanian, "A hybrid variable neighborhood search algorithm for solving the limited-buffer permutation flow shop

World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:3, 2017

scheduling problem with the makespan criterion," *Computers & Operations Research*, vol. 52, pp. 260-268, 2014.

[29] R. M'Hallah, "Minimizing total earliness and tardiness on a permutation flow shop using VNS and MIP" *Computers & Industrial Engineering*, vol. 75, pp. 142-156, 2014.

[30] D. Lei, "Variable neighborhood search for two-agent flow shop scheduling problem," *Computers & Industrial Engineering*, vol. 80, pp. 125-131, 2015.

[31] J. Jungwattanakit, M. Reodecha, P. Chaovalitwongse, and F. Werner, "A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria," *Computers & Operations Research*, vol. 36, pp. 358-378, 2009.

[32] R. Zhang, and C. Wu, "A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective," *Computers & Operations Research*, vol. 38, pp. 854-867, 2011.

[33] P. Jarosław, S. Czeslaw, and Z. Domonik, "Optimizing bicriteria flow shop scheduling problem by simulated annealing algorithm," *Procedia Computer Science*, vol. 18, pp. 936-945, 2013.

[34] F. Nikzad, J. Rezaeian, I. Mahdavi, and I. Rastgar, "Scheduling of multi-component products in a two-stage flexible flow shop," *Applied Soft Computing*, vol. 32, pp. 132-143, 2015.