International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 3350-3360

Solving Dynamic Multi-Product Multi-Level Capacitated Lot-Sizing Problems with Modified Part Period Balancing Heuristics Method

Songwut Prakaiwichien, Vichai Rungreunganun

Department of Industrial Engineering, Faculty of Engineering,
King Mongkut's University of Technology North Bangkok 10800, Thailand.

Abstract

The objective of this research is to solve the dynamic multi-product multi-level capacitated lot sizing problem on condition that shortage is not allowed by developing Heuristic method using the Part Period Balancing (PPB) method. This method start from determine many patterns of production plans of finished product item in accordance with number of periods planed. After that, consider production plans of finished product item one by one. Then determine lot size of remain items in each period from the first period until last period. In each period prioritize lot sizing of item according to setup cost and inventory holding cost ratios **(**RAH) from the highest to the lowest. Next, calculate total costs of production plans of all items. If total costs of new production plan is lower than prior plan, then determine lot size according to the new plan. The experiment reveals that when comparing to the solution calculated by Lingo 12 software, the percentage of difference of the solutions does not exceed than 9 %. While this method consumes processing time much lesser than Lingo 12 software even though they are the large scale problems.

INTRODUCTION

Determining Lot sizing is a tool to solve the problem of lot sizing is material requirement planning system which is widely used industries that have multi level product structures. This method is to find the number of materials, parts and subassemblies according to demand of finish product in both quantity and time from production schedule. This procedure is to prevent the problem of inventory shortage during production and reduce cost of inventory management in terms of raw materials and work in process. Two important issues should be taken into account are amount of each resource used in the production must not exceed the capacity of limited resources and the cost of production which are holding cost , setup cost and overtime cost.

The objective of this research is to solve the problem of lot sizing of the all items in multi level product structure that assembly product system under limited production resources with minimal total costs (or close to). Cost of production to be considered are holding cost, setup cost and overtime cost on condition that shortage is not allowed while knowing the number of final product requirements that is dynamics demand which are deterministic data. The solution of this problem can be as follows:

1. Exact methods

This method uses commercial programs in order to find the optimal solution. However, this method is complicate and use a lot of time in processing especially large-size problems. Therefore, this method is appropriate for small-size problems only and can not be used in real production plan process.

2. Heuristic methods

Since, in practical the problems are always large and complicated, therefore, in order to find the solution that close to the best solution, Heuristic methods, that is an approximate finding solution and consume acceptable time, are more practical to used in solving the said problems.

Therefore, this research propose heuristic method that is developed for finding the good solution with acceptable processing times which can be more practical to use in finding the solutions with both good quality of solution and processing time.

REVIEW LITERATURE

To lot sizing problems in assembly production systems by using dynamic programming algorithms and a branch and bound algorithms to obtain the optimal solution, Crowston and Wagner (1973). Tempelmerier and Derstroff (1996) develop a Lagrangean heuristic. They also started with a Wagner-Within solution and then used a smoothing procedure to try to find a feasible solution. Ozdamar and Barbarosoglu (2000) presented another heuristic using Lagrangean Relaxation and Simulated Annealing. Dellaert, Jeunet and Jonard (2000) propose a genetic algorithms to solve the general multi-level lot-sizing problem with time-varying costs by develop a binary encoding genetic algorithm and design five specific genetic operators to ensure that exploration takes place within the set of feasible solutions. Dellaert and Jeunet (2003) proposed a randomized multi-level lot sizing heuristic for general product structures. Pitakaso et al. (2006) solve the multi-level lot-sizing problems by using ant-based algorithm

Han et al. (2009) solve the uncapacitated multilevel lot-sizing problem with assembly structure by using a particle swarm optimization (PSO) algorithm and compare with a genetic algorithm (GA).

Helber and Sahling (2010) propose to solve dynamic multi-level capacitated lot sizing which has positive lead times by using the Optimization-based algorithm. They found that the results are more effective than the approach of Tempelmeier / Derstroff and Stadtler.

James and Almada-Lobo (2011) propose new iterative MIP-based neighborhood search heuristics in solving single- and parallel-machine capacitated lot sizing and scheduling problem with sequence-dependent setup times and costs. Then, compare this method with other methods and commercial solver.

Mohammadi and Ghomi (2011) propose to solve capacitated lot sizing problem in flow shops with sequence-dependent setups by using genetic algorithm-based heuristic that combines genetic algorithm with rolling horizon approach. The experiment shows that the large-size problem instances solved by this algorithm are better than former heuristics method.

Toledo, et al. (2011) proposed a hybrid heuristic approach for solving multi-level capacitated lot sizing problems. The results of this comparison with the AMH and SH method found that this approach was better.

Wu et al. (2011) presented two new mixed integer programming models for capacitated multi-level lot-sizing problems with backlogging using linear programming relaxations which give the lower bound of the optimal response.

Xiao et al. (2011) study to solve the uncapacitated multi level lot-sizing (MLLS) problems with component commonality and multiple end-items by using an approach based on the variable neighborhood search (VNS) algorithm. They test with benchmark instances comparing to other algorithm and found that the VNS algorithm can solve MLLS problems efficiently with good solution.

Xiao et al. (2012) study to solve uncapacitated multilevel lot-sizing problems by using iterated neighborhood search (INS) algorithm. They test with different size of benchmark instances and found this algorithm perform good results.

Seeanner et al. (2013) study to solve general multi-level lot-sizing and scheduling problems by propose to combine the principles of variable neighborhood decomposition search and the fix & optimize heuristic (VNDS). Then compare the results with other methods including a standard MIP-solver.

Stadtler and Sahling (2013) propose a new model formulation for lot-sizing and scheduling of multi-stage flow lines which allows a continuous material flow with zero lead-time offset and can use standard MIP software to find the solution. Then, compare the results and processing time.

Toledo et al. (2013) proposes to solve the multi-level capacitated lot sizing problem with backlogging by using a new hybrid multi-population genetic algorithm (HMPGA) approach which combines a multi-population based metaheuristic using fix-and-optimize heuristic and mathematical programming techniques. After testing with Multi-Item Lot-Sizing with Backlogging library for 4 test sets, HMPGA's performance is better comparing the other two methods published.

Chen (2015) study to solve two dynamic multi-level capacitated lot sizing problems (MLCLSP) both with and without setup carryover. This research proposes a new fix-and-optimize (FO) approach by developing a variable neighborhood search approach. Then, Compare the results with the fix-and-optimize approach proposed by Helber and Sahling. The Experiments show that the result from this approach is a better.

Fiorotto et al. (2015) propose approach to solve capacitated lot sizing problem with multiple items, setup time and unrelated parallel machines. In order to solve the master problem, they apply methods that combine Lagrangian relaxation and Dantzig–Wolfe decomposition in a hybrid form. The results are shown that this methods produce good quality of lower bounds and competitive upper bounds comparing with other methods.

Duda (2017) propose to solve a multi-item capacitated lot-sizing multi-family problem with set up times by using a genetic algorithm (GA) hybridized with variable neighborhood search (VNS). It is found that, for large instances problems, the results from this method are better than that of a dedicated genetic algorithm and CLPEX Solver-based rolling horizon methods.

METHODS

MODEL FORMULATION

The dynamic multi level capacitated lot sizing problem is aimed at minimizing variable production costs over a finite planning interval. The variable production costs which are considered comprise of holding costs and setup costs. The planning interval is divided into several periods and limited by the planning horizon T.

For each period in the planning interval, the end item demand is assumed to be known and has to be fulfilled without backlogging. Inventory holding costs are calculated based on the end of period inventory. Setup costs and setup times accrue for an item in each period of production. The basic model for this problem (Stadtler, 1996) is as follows.

Minimize $\sum_{i=1}^{I}\sum_{t=1}^{T}(sc_i Y_i + h_i I_{it} + oc_{mt} O_{mt})$ (1)

Subject to

$I_{i,t-1} + X_{it} = D_{it} + \sum_{k \in S_i} r_{ik}^{d}.X_{kt} + I_{it}$ $\forall$ i = 1,…,I ; t = 1,…,T (2)

$\sum_{i=1}^{I} a_{mi}.X_{it} \leq C_{mt} + O_{mt}$ $\forall$ m = 1,…,M ; t = 1,…,T (3)

$X_{it} \leq B_{it} Y_{it}$ $\forall$ i = 1,…,I ; t = 1,…,T (4)

I_{it} , O_{it} ,X_{it} $\geq$ 0 $\quad\forall$ i = 1,…,I ; t = 1,…,T (5)

Y_{it} $\in$ {0,1} $\quad\forall$ i = 1,…,I ; t = 1,…,T (6)

Indices and index sets:

i	Items or operations	, i =1, ..., I
m	Resources	, m=1, ...,M
t	Periods	, t=1, ..., T
S_i	Set of immediate successors of item i in the bill of material	

where the known parameters are:

a_{mi}	Capacity needed on a resource m for one unit of item i
B_{it}	Large number, not limiting feasible lot-sizes of item i in period t
C_{mt}	Available capacity of resource m in period t
h_i	Holding cost for one unit of item i in a period
oc_{mt}	Overtime cost for one unit of resource m in period t
D_{it}	External demand for item i in period t
r_{ik}^{d}	Number of units of item i required to produce one unit of the immediate successor item k
sc_i	Setup cost for a lot of item i

and the decision variables are:

I_{it}	Inventory of item i at the end of period t
O_{mt}	Amount of overtime of resource m used in period t
X_{it}	Amount of item i produced in period t
Y_{it}	Binary variable indicating where production is allowed for item i in period t (Y_{it} =1, if item i is produced in period t and Y_{it} = 0, otherwise)

The objective (1) is to minimize the total cost which is the sum of holding, setup and overtime costs. Equation (2) is the inventory balance to make sure that no backlogging will occur. For multi-level production, a lot-size of item k will result in a dependent demand for its immediate predecessor items i. Equation (3) presents the capacities required for lot-size production which must not exceed available normal capacities (possibly extended by overtime. Capacity requirements result from both production time per item times the amounts produced as well as setup times incurred with each lot. Constraints setup in (4) enforce binary variables Y_{jt} to be 1, in the case that a lot of item j is produced in a period t. All variables are restricted to non-negative and binary values as shown in (5) and (6), respectively.

PROPOSED METHOD FOR SOLVING THE PROBLEM

As stated before, the objective of this paper is to propose a heuristic method to find the solution which is to minimize total cost for dynamic multi product multi level capacitated lot sizing problems under assembly production systems while backlogging is not allowed. The proposed method is explained as follows.

The algorithm of this method starts from determining the lot sizing of each product at all periods using the Lot for Lot method. Then, determine the lot sizing of finished product using the Part Period Balancing (PPB) method which is a basic heuristic method for solving a lot sizing problem. Factors needed to be considered during determining the lot sizing in all items and at all peirods are the relationship between products in accordance with Bill of Material (BOM) and resources constrain in producing that item. After that, determine the lot sizing for other items in BOM according to setup cost and inventory holding cost ratio (RAH) starting from the highest to the lowest.

Steps of solving the problem

1. Determine the general data

1.1 Bill of material(BOM) which are all items involved and relationship between these items , resources used for production in each item that can be represented by $a_{m,i}$, $B_{i,t}$, $C_{m,t}$, $r_{i,k}^{d}$

1.2 Cost involved which are holding cost per unit (h_i), setup cost per unit (sc_i), overtime cost per unit (oc_{mt})

External demand ($D_{i,t}$) of finished product which is highest level in BOM (Level 0)

2. Determine initial production plan which is specified as production plan No. 1 by using Lot for Lot method (LFL) in all items and all periods as presented in Equation (7)

$X_{i,t} = D_{i,t}$, for all period t (7)

3. Calculate Cumulative setup cost (CS) as follows:

3.1 Calculate CS of all items in the lowest level. Assign $CS_{i,t} = sc_i$, i = items in lowest level

3.2 Calculate CS of item in the next higher level. Calculate CS of any item i in this level as presented in Equation (8)

$$CS_{i,t} = SetupcostU(i) + \sum_{k\in\eta_i} \frac{CS_{j,t}}{N_j} \quad (8)$$

η_i = Set of immediate successors of item i in the BOM and N_j = Number of immediate successor item of j

4. Prioritize order of item in order to determine lot sizing by sorting setup cost and inventory holding cost ratios (RAH) from the highest to the lowest value. as presented in Equation (9)

$RAH_i = CS_{i,t} / h_{i,t}$ (9)

5. Calculate Modified resouce usage (MR) of all items as presented in Equation (10)

$$MR_{m,i} = a_{m,i} + \sum_{j\in\eta_i} \frac{P_{j,i} MR_{m,j}}{N_j} \quad (10)$$

$a_{m,i}$ Capacity needed on a resource m for one unit of item i

$P_{j,i}$ Number of units of item j required to produce one unit of the immediate successor item i

$MR_{m,j}$ will be calculatate only item j that uses common resources m. if item j uses other resource , it will not be calculated.

Steps to calculate MR

5.1 Calculate $MR_{m,i}$ of all items in the lowest level.

$MR_{m,i} = a_{m,i}$, i = items in lowest level

5.2 Calculate $MR_{m,i}$ of Item in the next higher level.

Calculate MR of any item i in this level as presented by equation No.10

6. Determine production plan for finished product in the highest level (Level no. 0) which is specified as item No. 1. Determine lot sizing in each period from the first period until last period (T). The steps of calculation are as follows:

6.1 Let p = 1

6.2 Let t = p

6.3 Consider $x_{i,t}$ which can be devided into 2 cases.

6.3.1 If $x_{i,t} = 0$, t = t+1

6.3.2 If $x_{i,t} > 0$ then, calculate remain capacity of resource *m* in period *t* ($RC_{m,t}$).

$RC_{m,t} = C_{m,t} - \sum_{i=1}^{I} a_{m,i} . x_{i,t}$

After that, calculate cummulative $RC_{m,t}$ from period t =1 to t (RemainCapA) .

RemainCapA = $\sum_{t=1}^{t} RC_{1,t} / MR_{m=1,i=1}$

RemainCapA can be devided into 2 cases

1) If RemainCapA ≥ D_{t+1}

$x_{1,t} = D_t + D_{t+1}$

$x_{1,t+1} = 0$

t = t +2

2) If RemainCapA < D_{t+1} , t = t+1

6.4 Go back to step 6.3 and repeat until last period (T). Then, We will get lot size for item 1 ($x_{i=1}$) at all periods.

6.5 Update lot size of other item (j) as $x_{j,t} = x_{i=1,t}$ in all periods. As a result, We will get Production plan No. p which update lot sizing of item 1 in all peroids.

6.6 Let p = p+1

6.7 Consider p compare to T

1) If p < T go back to step 6.2
2) If p = T, finish. As a result , we will get the amount of T-1 production plans of item 1 in all periods

7. Determine production plan for remain items which use common resources with item 1 . Determine lot sizing in each period from the first period until last period . In each period prioritize lot sizing of item according to RAH from the highest to the lowest. If lot sizing for all item are determine, go to next period and do the same process until period T-1.

7.1 Let p=1

7.2 Let information for initial production plan as production plan No. p for item 1 ($x_{i=1}$) in all period from step 6

7.3 Consider inventory at the end period of item i in period t-1($Inv_{i,t-1}$) as presented in Equation (11)

$$I_{it} = I_{i(t-1)} + X_{it} - D_{it} - \sum_{k \in S_i} r_{ik}^d . X_{kt} \qquad \forall\ i = 1,\ldots,I\ ;\ t = 1,\ldots,T \qquad (11)$$

If $Inv_{i,t-1} < 0$
$x_{i,t} = x_{i,t} - Inv_{i,t}$

recalculate $Inv_{i,t}$ and Inventory of immediate predecessor item of item i
recalculate $RC_{m,t}$

7.4 Consider $RC_{m,t}$ which can be devided to 2 cases

7.4.1 if $RC_{m,t} \geq 0$

To determine $x_{i,t}$, take the following steps

7.4.1.1 Consider $x_{i,t}$ which can be devided to 2 cases

1) If $x_{i,t}$ =0 skip to next item and go back to step 7.3

2) If $x_{i,t}$ > 0 go to step 7.4.12

7.4.1.2 Calculate CapVsQ for item i which can be devided to 2 cases

1) If item i has no immediate predecessor item that produce before and use common resources

CapVsQ = $\sum_{t=1}^{t} RC_{m,t}$ (calculate only in period t which has $x_{i,t}$ >0)

2) If item i is not the last item and has immediate predecessor item that produce before and use common resources

CapVsQ = $RC_{m,t}$

7.4.1.3 Find index1 which is equal to the amount of period that can be produced in advance in this period. The total amount of the item i that can be produced in advance must not exceed CapVsQ derived from step 7.4.1.2

7.4.1.4 Find index 2 which is equal to the amount of period that can be produced in advance in this period. The total holding cost which occured from the total amount of the item i that can be produced in advance must not exceed setup cost per unit of item i

7.4.1.5 Find Choice

Choice = mininum value of index1and index2

7.4.1.6 Consider Choice

1) if Choice = 0 skip to the item and go back to step 7.3

2) if Choice > 0 go to step 7.4.1.7

7.4.1.7 Consider whether item i has immediate predecessor item that must be produced in advance or not

1) if Yes, Let as follows:

$x_{i,t} = x_{i,t} + \sum_{t+1}^{t+Choice} x_{i,t}$

$x_{i,tt} = 0$, tt = t+1,…,t +Choice

$x_{j,t} = x_{i,t} - Inv_{i,t-1}$, Item j is all immediate predecessor item of item i

$x_{j,ttt} = x_{i,ttt}$, ttt = t+1,…,T

Then, update inventory at the end period t of Item i and all immediate predecessor items. Next, go to next item and go back to step 7.3

2) if No, Let as follows:

Plus $= \sum_{t=t+1}^{t+Choice} x_{i,t}$

$x_{i,tt} = 0$, tt = t+1,…, t +Choice

The result of $RC_{m,t}/MR_{m,t}$ compares with Plus can be devided into 2 cases

- If $RC_{m,t}/MR_{m,t}$ < Plus Let as follows

  $x_{i,t} = x_{i,t} + RC_{m,t}/MR_{m,i}$

Then, Add more number of item i for prior period in accordance with remain capacity in each period. The total amount of item i which added for prior period must be equal to Plus – $RC_{m,t}/MR_{m,i}$ from t-1 to 1 until meet the Plus – $RC_{m,t}/MR_{m,i}$

- If $RC_{m,t}/MR_{m,t} \geq$ Plus,
  then let $x_{i,t} = x_{i,t}$ + Plus

7.4.2 if $RC_{m,t} < 0$ take the following steps

7.4.2.1 Consider whether item i is in the lowest level that use resource m or not

1) If no , skip to the next item and go back to step 7.3
2) If yes, go to step 7.4.2.2

7.4.2.2 Let

Lack = $RC_{m,t}$
SumRemainCap= $\sum_{t=1}^{t-1} RC_{m,t}$

7.4.2.3 Consider whether SumRemainCap < Lack or not

1) If Yes, Let as follows:
   $x_{i,tt} = x_{i,tt} + RC_{m,tt}$, tt = 1,..,t-1
   $x_{i,t} = \max(x_{i,t} - SumRemainCap , 0)$
   $NoOT_{m,t}$ = Lack – SumRemainCap , $NoOT_{m,t}$ is Number of production in overtime period using resource m in period t
2) If No , Let as follows:
   $Add\_Av_{tt} = Y_{i,tt} . RC_{m,tt}$, $Y_{i,tt}$ =1 if $x_{i,tt} > 0$, 0 otherwise ; tt = 1,..,t-1
   SumAdd_Av $= \sum_{tt=1}^{t-1} Add\_Av_{tt}$

The result of SumAdd_Av compares with Lack can be devided into 2 cases

- If SumAdd_Av ≥ Lack , Let as follows:
  $x_{i,t} = x_{i,t}$ – Lack

Then, Add more number of item i for prior period which $x_{i,t} > 0$ in accordance with remain capacity in each period. The total amount of item i which added for prior period must be equal to Lack from t-1 to 1 until meet the Lack

- If SumAdd_Av < Lack , Let as follows:
  $x_{i,t}$ = $x_{i,t}$ – Lack
  $x_{i,tt}$ = $x_{i,tt}$ + Add_Av , tt = 1,..,t-1 and calculate only in period tt which has $x_{i,tt}$ >0

  Then, Add more number of item i for prior period which $x_{i,t}$ = 0 in accordance with remain capacity in each period. The total amount of item i which added for prior period must be equal to Lack – sum(Add_Av) from t-1 to 1 until meet the Lack– sum(Add_Av)

7.5 We will get production plan for remain items which use common resources with item i

8. Determine production plan for remain items. Determine lot sizing in each period from the first period until last period . In each period prioritize lot sizing of item in accordance with resource. We will choose resource that produce the item at the highest level first and produce all item which use this resource. Then, will choose resource next lower level. In each resource, prioritize lot sizing of item according to RAH from the highest to the lowest. Then, determine lot sizing of items in each period by follow step 7 If lot sizing for all item are determined, go to next period and do the same process until period T-1.

9. Calculate total cost as presented in Equation (12).

$$Total \sum_{i=1}^{I}\sum_{t=1}^{T}(sc_iY_i + h_iI_{it} + oc_iO_{it}) \qquad (12)$$

10. Consider whether total cost of production plan that we get from this method is lower than total cost of prior production plan or not. if yes, The solution will be production plan that we get from this method.

11. Let p =p+1.

12. Consider whether p = T or not.

12.1 if no, go back to step 7-12.

12.2 if yes , the solution will be total cost and production plan

The performance of the Heuristics Method described in the previous section was evaluated on a set of testing problems. The algorithm was programmed with Matlab R2013 software. The test instance data for testing is assembly product structures shown in Fig.1 (10 items) constrained by three resources (A , B , C). The test instance data of each test set is shown in Table 1 and 2. (Stadtler and Surie , 2000)

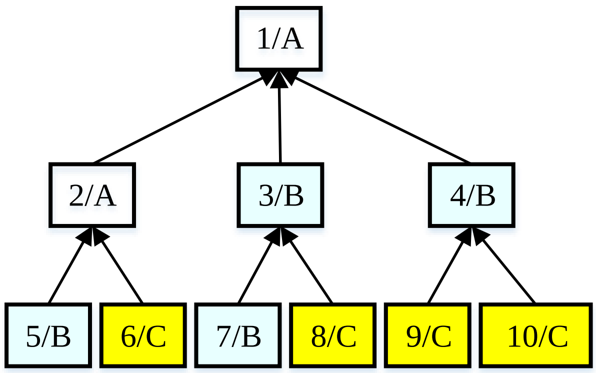

Figure 1. The Product structure for testing

Table 1. The test instance data for all Test set

	Item									
	1	2	3	4	5	6	7	8	9	10
Holding cost/unit/period	10	3	3	3	1	1	1	1	1	1
Setup cost for item	807									
Overtime cost/unit/period	10,000									
Capacity A (all period)	224									
B (all period)	448									
C (all period)	448									

Table 2. The External Demand for each test set

Test set No.	External Demand for item 1
1	69,77,72,75
2	69,77,72,75,94
3	69,77,72,75,94,92
4	69,77,72,75,94,92,111
5	69,77,72,75,94,92,111,96
6	69,77,72,75,94,92,111,96,117
7	69,77,72,75,94,92,111,96,117,123
8	69,77,72,75,94,92,111,96,117,123,123
9	69,77,72,75,94,92,111,96,117,123,123,126
10	69,77,72,75,94,92,111,96,117,123,123,126,141
11	69,77,72,75,94,92,111,96,117,123,123,126,141,124
12	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128
13	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100
14	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118
15	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121
16	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121,119,101
17	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121,119,101,81
18	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121,119,101,81,76
19	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121,119,101,81,76,74
20	69,77,72,75,94,92,111,96,117,123,123,126,141,124,128,100,118,121,119,101,81,76,74,64

RESULT AND DISCUSSION

Table 3. Results of the Numerical Experiments

Test set No.	Best Solution found by		Difference of Solution	Times (second)		Type of Lingo solution
	Lingo v.12	Heuristic		Heuristic	Lingo v.12	
1	20,069	21,023	4.76%	<1.0	2	Optimal
2	25,481	26,739	4.94%	<1.0	70	Optimal
3	29,621	31,756	7.21%	<1.0	508	Optimal
4	34,904	37,768	8.21%	<1.0	8,545	Optimal
5	40,262	42,906	6.57%	<1.0	50,090	Optimal
6	45,606	49,388	8.29%	1	102,256	feasible
7	51,416	55,298	7.55%	1	20,821	feasible
8	57,612	60,733	5.42%	1.1	22,490	feasible
9	64,742	69,309	7.05%	1.2	25,710	feasible
10	72,285	76,993	6.51%	1.3	23,505	feasible
11	79,111	84,096	6.30%	1.4	24,231	feasible
12	90,552	91,436	0.98%	1.5	24,061	feasible
13	92,258	97,429	5.61%	1.7	23,544	feasible
14	100,586	103,803	3.20%	1.7	23,905	feasible
15	104,426	110,813	6.12%	1.8	24,393	feasible
16	115,841	123,287	6.43%	2.1	16,794	feasible
17	123,799	128,529	3.82%	2.5	18,136	feasible
18	128,276	133,267	3.89%	2.8	23,873	feasible
19	135,850	135,101	-0.55%	3.8	25,115	feasible
20	136,228	139,107	2.11%	6	86,400	feasible

In table 3, it is shown the comparison of the total cost and processing times using the Lingo version v.12 software and the heuristic method which are as follows.

Test set No. 1-5 with period of 4-8, Lingo v.12 software can find the optimal solution in 2 , 70 , 508 , 8545 and 50090 seconds, respectively. However, the solutions using the heuristic method, are not exceed 9 % more than the solutions calculated by Lingo 12 software, while consuming processing time less than 1 second.

Test set No. 6-20 with period of 9-24, Lingo v.12 software cannot find the optimal solution, therefore, we terminate Lingo v.12 software process when the solution is not changed in at least 3 hours. The solution during period of 9-24 consume processing more than 6 hours. While the solutions using the heuristic method are not exceed 9 % more than the solutions calculated by Lingo v.12 software and consume processing time for 1 -6 seconds.

Considering the difference of the solutions between Lingo v.12 software and the heuristic method, we found that the increase in the percentage of difference of the solutions does not depend on the sizes of the problems. For example, Test set No. 17-20, which are large problems, the differences of the solutions is 3.82% 3.89% -0.55% and 2.11% respectively. While Test set No. 7 - 10 which are smaller problems, the differences of the solutions are 8.21%, 6.57%, 8.29%, and 7.55%, respectively. Therefore, the differences of the solutions depend on the solution from Lingo v.12 software.

The relationship between processing time and difference of solution from Heuristic Method and Lingo v.12 in various size problems can be shown in case that Lingo 12 software can find optimal solution only. The optimal solution occurred in test set No. 1-5 with period of 4-8. Since larger size problems which are period 9 – 24, Lingo v.12 cannot find the optimal solution within 24 hours. The more period, the processing time in solving problems will increase exponentially.

The relationship between processing time and difference of solution from Heuristic Method and Lingo v.12 in various size problems can be presented in chart No. 1

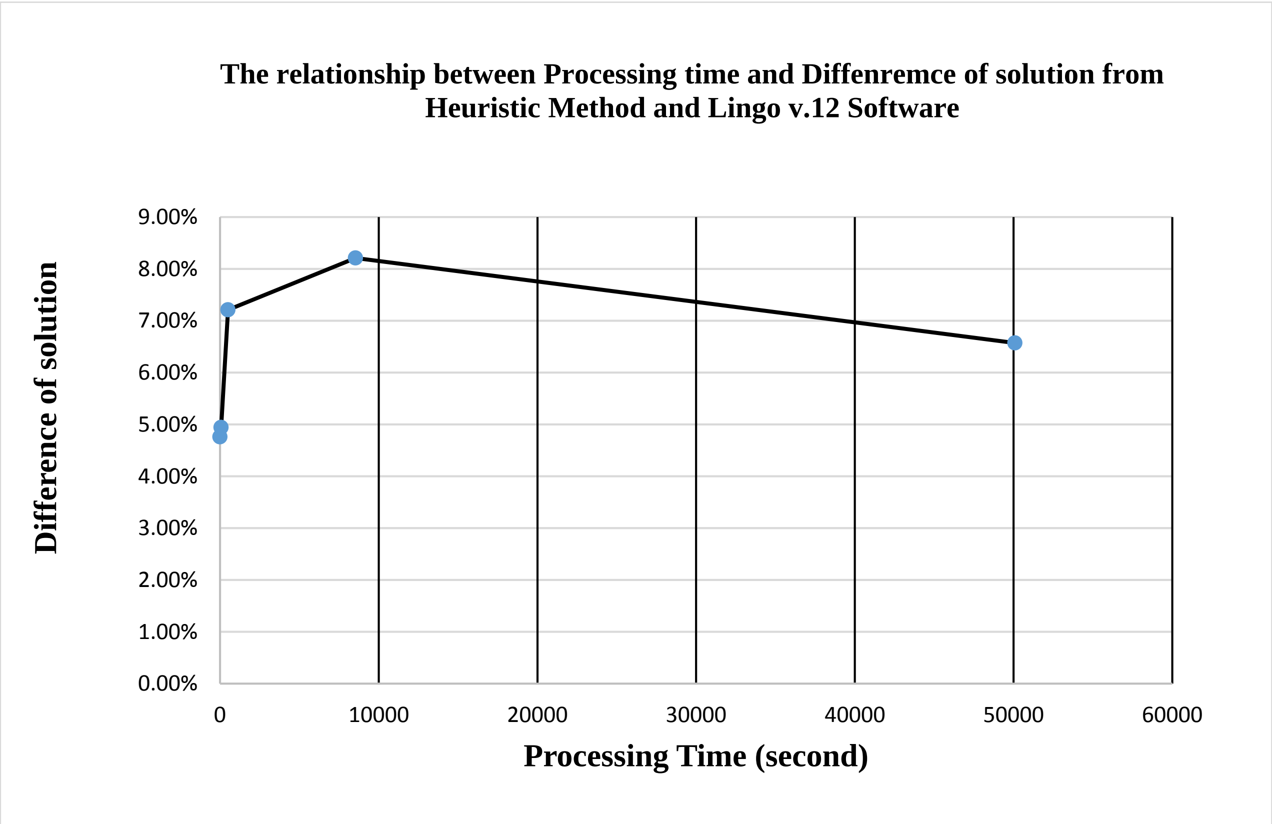

Chart No. 1 represent the relationship between processing time and difference of solution from Heuristic Method and Lingo v.12 in various size problems

The relationship between size of problems and difference of solution from Heuristic method and Lingo v.12 are presented in chart No. 2

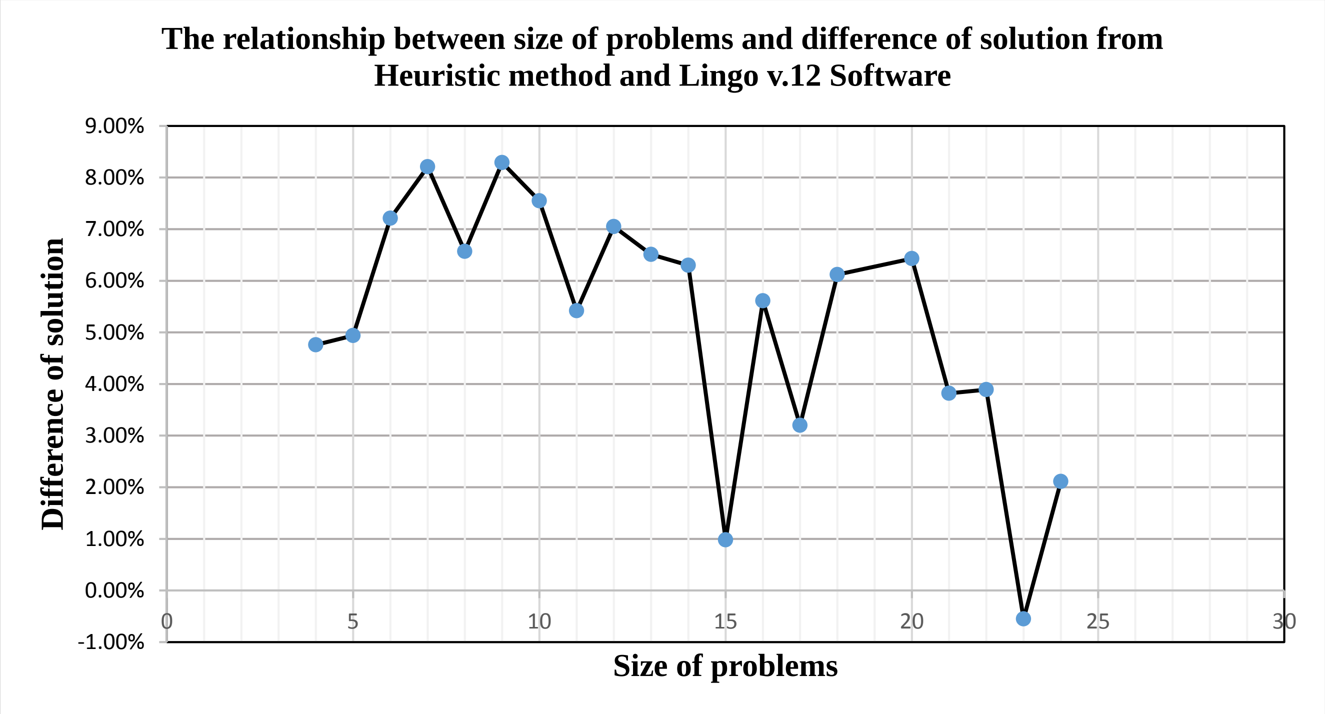

Chart No. 2 represent the relationship between size of problems and difference of solution from Heuristic method and Lingo v.12 in various size problems

The experiments revealed that Heuristics Method could obtain good approximation solution of Dynamic Multi-Product Multi-Level Capacitated Lot-Sizing Problems with processing time less than 6 second even though it is the large scale problems. While Lingo v.12 software can find optimal solutions only 8 periods. In the periods 9 -24, the solutions are just feasible solutions which are terminated when solutions is not changed for more than 3 hours.

CONCLUSION

This research develops the heuristic method by using the Part Period Balancing (PPB) method in order to solve Dynamic Multi-Product Multi-Level Capacitated Lot-Sizing Problems. Moreover, limited resources and total cost have to be taken into account. This method start from determine production plan of item, which is finished product, in many patterns at all periods. Then determine lot size of remain items in each period from the first period until last period. In each period prioritize lot sizing of item according to setup cost and inventory holding cost ratio (RAH) from the highest to the lowest. If lot sizing for all item are determine, go to next period and do the same process until last period. In determining lot size in each period, we must consider how many period that remain resources capacity can be produced in advance. In addition, we consider whether that alternative can save the total cost or not, if not, production plan must be the same. The experiment reveals that when comparing to the solution calculated by Lingo v.12 software, the percentage of difference of the solutions does not exceed than 9 %. While this method consumes processing time less than 6 seconds even though they are the large scale problems.

For further research, it should be studied to find more alternatives to create production plan of finished product in order to expand the capacity in finding solutions in the wider area. This alternative will give more opportunity for better solutions.

REFERENCES

[1] Chen, H. (October 2015). "Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems." Omega. Vol. 56 : 25-36.

[2] Crowston, W.B., Wegner M.W.(1973). "Dynamic Lot Size Models for Multi stage Assembly Systems." Management Science. Vol. 20 : 14-21.

[3] Dellaert, N., Jeunet, J. and Jonard, N. (2000). "A genetic algorithms to solve the general multi-level lot-sizing problem with time-varying costs." International Journal of production economics. Vol. 68 : 241-257.

[4] Dellaert, N.P. and Jeunet, J. (2003). "Randomized multi-level lot sizing heuristic for general product structures." European Journal of Operational Research. Vol.148 : 211-228.

[5] Duda, J. (April 2017). "A hybrid genetic algorithm and variable neighborhood search for multi-family capacitated lot-sizing problem." Electronic Notes in Discrete Mathematics. Vol. 58 : 103-110.

[6] Fiorotto, D.J., Araujo, S.A. and Jans, R. (November 2015). "Hybrid methods for lot sizing on parallel

machines." Computers & Operations Research. Vol. 63 : 136-148.

[7] Han, Y. et al. (June 2009). "Solving uncapacitated multilevel lot-sizing problems using a particle swarm optimization with flexible inertial weight." Computers and Mathematics with Applications. Vol. 57 : 1748-1755.

[8] James, R.J.W. and Almada-Lobo, B. (December 2011). "Single and parallel machine capacitated lot sizing and scheduling: New iterative MIP-based neighborhood search heuristics." Computers & Operations Research. Vol. 38 : 1816-1825.

[9] Mohammadi, M. and Fatemi Ghomi, S.M.T. (June 2011). "Genetic algorithm-based heuristic for capacitated lot sizing problem in flow shops with sequence-dependent setups." Expert Systems with Applications. Vol. 38 : 7201-7207.

[10] Ozdamar, L. and Barbarosoglu, G. (2000). "An integrated lagrangean relaxation simulated annealing approach to multi-level multi-item capacitated lot sizing problem." International Journal of Production Economics. Vol. 68 : 319-331.

[11] Pitakaso, R., et al. (2006). "Combining population-based and exact methods for multi-level capacitated lot-sizing problems." International Journal of Production Research. Vol. 44: 4755-4771.

[12] Seeanner, F., Almada-Lobo, B., Meyr, H. (January 2013). "Combining the principles of variable neighborhood decomposition search and the fix & optimize heuristic to solve multi-level lot-sizing and scheduling problems." Computers & Operations Research. Vol. 40 : 303-317.

[13] Stadtler, H. (1996). "Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lot sizing." European Journal of Operational Research. Vol. 94:561-581

[14] Stadtler, H. and Sahling, F. (March 2013). "A lot-sizing and scheduling model for multi-stage flow lines with zero lead times." European Journal of Operational Research. Vol. 225: 404-419.

[15] Stadtler, H. and Sürie, C. (2000). [Online]. "Description of MLCLSP Test Instances," Technische Universität Darmstadt, Tech. Rep., Available from:http://www.vwl.tu-darmstadt.de/bwl1/forschung/ti_mlclsp/

[16] Tempelmeier, H. and Derstroff, M. (1996). "A Lagrangean-based heuristic for dynamic multi-level multi-item constrained lot sizing with setup times." Management Science. Vol. 42 : 738-757.

[17] Toledo, C.F.M., Oliveira, R.R.R. and Franca, P.M. (2011). "A Hybrid Heuristic Approach to Solve the Multi-Level Capacitated Lot Sizing Problem." 2011 IEEE Congress on Evolutionary Computation (CEC). ,1194-1201.

[18] Toledo, C.F.M., Oliveira, R.R.R. and França, P.M. (April 2013). "A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging." Computers & Operations Research. Vol. 40 :910-919.

[19] Xiao, Y., et al. (March 2011). "A variable neighborhood search based approach for uncapacitated multilevel lot-sizing problems." Computers & Industrial Engineering. Vol. 60 : 218-227.

[20] Xiao, Y., et al. (March 2012). "Neighborhood search techniques for solving uncapacitated multilevel lot-sizing problems." Computers & Operations Research. Vol. 39 : 647-658.

[21] Wu, T., et al. (October 2011). "An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging." European Journal of Operational Research. Vol. 214 :428-441.