Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 9 (2): 296-303, 2009
ISSN 1812-5654

Development of Heuristics for Multi-Product Multi-Level Capacitated Lotsizing Problem with Sequence-Dependent Setups

[1]M. Mohammadi, [1]S.M.T. Fatemi Ghomi, [1]B. Karimi and [2]S.A.Torabi
[1]Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran
[2]Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Abstract: This study considers the problem of multi-product multi-level capacitated lotsizing and sequencing problem with sequence-dependent setups. A Mixed Integer Programming (MIP) formulation of the problem is proposed which is impractical to solve in reasonable computing time for non-small instances. Reducing the dimensionality of the problem and allowing to solve larger instances, a modified mathematical model is developed which ignores majority of combinations. The ability to quickly find integer-feasible solutions for non-small instances is another aspect of this paper. Hybrid methods that mixes rolling-horizon approach and heuristic are developed. Heuristic is used to determine binary variables of current period. To test the accuracy of hybrid methods, a procedure for obtaining a lower bound on the optimal solution is developed. The trade-offs between objective values and computing times are also provided.

Key words: Sequencing and scheduling, pure flow shop, setup carry over, rolling-horizon, setup state selection rule

INTRODUCTION

Lotsizing and scheduling problems have been an area of active research starting from the seminal study of Wagner and Whithin (1958). Since then there has been a considerable amount of investigation in order to incorporate other important features.

Among the characteristic features of the models for lotsizing and scheduling are the segmentation of the planning horizon, the time dependence of the model parameters, the information degree of the model parameters, the number of products and production stages, the production structure and the capacity restrictions (Fandel and Stammen-Hegene, 2006; Merece and Fonton, 2003; Karimi *et al*., 2003).

Models of lotsizing and scheduling are divided in the literature into small bucket and big bucket problems (Eppen and Martin, 1987). Small bucket problems break the planning horizon in small time periods such that at most one product can be manufactured in a single period. Consequently, if a setup is performed, the entire time interval must be devoted to the setup. That is, setups and production runs comprise of an integer number of time intervals. Small bucket problems for multi-level multi-product production are the Multi-Level Discrete Lotsizing and Scheduling Problem (MLDLSP) and the Multi-Level Proportional Lotsizing and Scheduling Problem (MLPLSP) (Kimms, 1996). Both models enable simultaneous lotsizing and scheduling, but limit the number of products to be manufactured in a period.

The Multi-Level Capacitated Lotsizing Problem (MLCLSP), a big bucket problem, does not have this disadvantage, but it can not fix lotsizes and schedules simultaneously. To attempt to unite the advantages of the MLPLSP and MLCLSP, Fandel and Stammen-Hegene (2006) have made a model formulation based on two-level time structure (Fleischmann and Meyr, 1997) which enables simultaneous lotsizing and scheduling for multi-product multi-level job shop production.

This study deals with the deterministic dynamic models with a finite planning horizon, where the production of several different products on serially-arranged capacitated machines is concerned. The challenging problem of efficient lotsizing on a flow shop with sequence-dependent setups is modeled using a new Mixed Integer Programming (MIP) formulation. Simultaneous lotsizing and scheduling is essential if sequence-dependent setup costs and setup times occur during production.

Solving the single-level multi-period CLSP with sequence-dependent setups is equivalent to solving multiple dependent TSPs (Gupta and Magnusson, 2005).

Corresponding Author: S.M.T. Fatemi Ghomi, Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran Tel: (+98 21) 66413034 Fax: (+98 21) 66413025

Hence, like the TSP, the CLSP with sequence-dependent setups also belongs to a set of problems that are called NP-hard. That means it is very difficult to optimally solve large instances of the problem. In fact, the solution time rises exponentially as either the number of variables and constraints increase. The introduction of multi-level production makes the problem even more complicated. Therefore, it is necessary to find reasonable heuristic solutions for medium and large instances. Also it is important to develop a computable lower bound in order to test the accuracy of the heuristics.

In this study, mathematical model and solution approaches for multi-product multi-level capacitated lotsizing problem with sequence-dependent setups have been proposed.

PROBLEM DEFINITION AND MATHEMATICAL MODEL

Assumptions: Following assumptions are made for the problem of multi-product multi-level capacitated lotsizing with sequence-dependent setups:

- Several products are produced on serially-arranged machines in flow shop structure. There are M stages of processing which occur in a linear fashion from 1,…,M
- Each machine is constrained in capacity
- When the machines are setup, sequence-dependent setup costs and times accrue
- The setting-up of a machine must be completed in a period
- There must be precisely N (number of products) setups in each period on each machine, even if a setup is just from a product to itself. Since a setup time (and cost) from a product to itself is zero, note that the model does not force a machine to have exactly N positive-time (and cost) setups but rather up to N such setups. The remaining zero-time (cost) setups are modeling phantoms and do not exist in reality (Clark and Clark, 2000; Clark, 2003). This feature makes possible for a lotsize, or production run, to continue over consecutive time periods without incurring real setup for latter period (setup carry over)
- The required resources and parts must be ready for production
- External demand exists for final products and is satisfied at the end of each period
- There are no lead times between the different production levels for transportation or cooling the products
- Shortages are not permitted
- A component can not be produced earlier in a period than the production of its required component is finished. In other words, production on a production level can only start if a sufficient amount of the product from the previous production level is available; this is called vertical interaction
- To guarantee the vertical interaction, idleness before each production is defined with the help of shadow product (Fandel and Stammen-Hegene, 2006)
- There are no demand and no storage costs for shadow products
- At the beginning of the planning horizon each machine is setup for a defined product. The starting setup configuration on all machines are the same
- The triangle inequality holds, i.e., it is never faster to change over from one product to another by means of a third product. In other words, a direct changeover is at least as capacity efficient as going via another product
- Setup cost has the form $w_{ijm} = f_w.S_{ijm}$ where, f_w is opportunity cost per unit of setup time
- Infinite buffers exist between stages and before the first stage and after the last stage

Mathematical model

Indices

i, j, k	:	Product type
n, n', n''	:	Designation for a specific setup No.
m	:	Level of production
t	:	Period

Parameters

T	:	Planning horizon
N	:	No. of different products
M	:	No. of production levels
bigM	:	A large real No.
$C_{m,t}$	:	Available capacity of machine m in period t (in time units)
$d_{j,t}$	:	External demand for product j at the end of period t (in units of quantity)
$h_{j,m}$	:	Storage costs unit rate for product j in level m
b_{jm}	:	Capacity of machine m required to produce a unit of product (or shadow product) j (in time units per quantity units)
$p_{j,m,t}$	:	Production costs to produce one unit of product j on machine m in period t (in money unit per quantity unit)
S_{ijm}	:	Sequence-dependent setup time for the setup of the machine m from production of product i to production of product j (in time units); for $i \neq j$, $S_{ijm} \geq 0$ and for $i = j$, $S_{ijm} = 0$

W_{ijm} : Sequence-dependent setup cost for the setup of the machine m from production of product i to production of product j (in money units); for $i \neq j$, $W_{ijm} \geq 0$ and for $i = j$, $W_{ijm} = 0$

j_{0m} : The starting setup configuration on machine m

Decision variables

$I_{j,m,t}$: Stock of product j at level m at the end of period t

y^{n}_{ijmt} : Binary variable, which indicates whether the nth setup on machine m in period t is from product i to product j $(y^{n}_{ijmt} = 1)$ or not $(y^{n}_{ijmt} = 0)$

x^{n}_{jmt} : Quantity of product j produced between nth and (n+1)th setups on machine m in period t

q^{n}_{jmt} : Shadow product: idle time (in quantity units) before production of product j on machine m in period t in order to ensure that direct predecessor of this product (production of product j on machine m-1 in period t) has been completed. In other words, the gap between nth setup and its production (in quantity units), in order to guarantee vertical interaction

$$\text{Min} \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} w_{ijm} \cdot y^{n}_{ijmt} + \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} p_{jmt} \cdot x^{n}_{jmt} + \sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} h_{jm} \cdot I_{j,m,t-1} \tag{1}$$

Subject to:

$$d_{j,t} = I_{j,M,t-1} + \sum_{n=1}^{N} x^{n}_{jMt} - I_{j,M,t}$$

$$j = 1,\ldots,N,\ t = 1,\ldots,T \tag{2}$$

$$I_{j,m,t-1} + \sum_{n=1}^{N} x^{n}_{jmt} = I_{jmt} + \sum_{n=1}^{N} x^{n}_{j,m+1,t}$$

$$j = 1,\ldots,N,\ m = 1,\ldots,M\text{-}1,\ t = 1,\ldots,T \tag{3}$$

$$\text{bigM}.\left(\sum_{i=1, i\neq j (\text{for } n'>1)}^{N} y^{n'}_{ijmt} - 1\right) + \sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{k=1}^{N} y^{n}_{ikmt} \cdot S_{ikm} + \sum_{n=1}^{n'}\sum_{k=1}^{N} b_{km} \cdot q^{n}_{kmt}$$

$$+ \sum_{n=1}^{n'}\sum_{k=1}^{N} b_{km} \cdot x^{n}_{kmt} \leq \text{bigM}.\left(1 - \sum_{i=1, i\neq j (\text{for } n''>1)}^{N} y^{n''}_{i,j,m+1,t}\right)$$

$$+ \sum_{n=1}^{n''}\sum_{i=1}^{N}\sum_{k=1}^{N} y^{n}_{i,k,m+1,t} \cdot S_{i,k,m+1} + \sum_{n=1}^{n''}\sum_{k=1}^{N} b_{k,m+1} \cdot q^{n}_{k,m+1,t} + \sum_{n=1}^{n''-1}\sum_{k=1}^{N} b_{k,m+1} \cdot x^{n}_{k,m+1,t}$$

$$j = 1,..,N,\ n' = 1,\ldots,N,\ n'' = 1,\ldots,N,\ m = 1,\ldots,M\text{-}1,\ t = 1,\ldots,T \tag{4}$$

$$\sum_{n=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y^{n}_{ijmt} S_{ijm} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{jm} \cdot x^{n}_{jmt} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{jm} \cdot q^{n}_{jmt} \leq C_{m,t}$$

$$m = 1,\ldots,M,\ t = 1,\ldots,T \tag{5}$$

$$x^{n}_{jmt} \leq (C_{m,t}/b_{j,m}) \cdot \sum_{i=1, i\neq j (\text{for } n>1)}^{N} y^{n}_{ijmt}$$

$$n = 1,\ldots,N,\ j = 1,\ldots,N,\ m = 1,\ldots,M,\ t = 1,\ldots,T \tag{6}$$

$$q^{n}_{jmt} \leq (C_{m,t}/b_{j,m}) \cdot \sum_{i=1}^{N} y^{n}_{ijmt}$$

$$n = 1,\ldots,N,\ j = 1,\ldots,N,\ m = 1,\ldots,M,\ t = 1,\ldots,T \tag{7}$$

$$y^{1}_{jiml} = 0$$

$$j \neq j_{0m},\ i = 1,\ldots,N,\ m = 1,\ldots,M \tag{8}$$

$$\sum_{i=1}^{N} y^{1}_{j0m,i,m,1} = 1$$

$$m = 1,\ldots,M \tag{9}$$

$$\sum_{j=1}^{N} y^{n}_{ijmt} = \sum_{k=1}^{N} y^{n+1}_{ikmt}$$

$$n = 1,\ldots,N\text{-}1,\ i = 1,\ldots,N,\ m = 1,\ldots,M,\ t = 1,\ldots,T \tag{10}$$

$$\sum_{j=1}^{N} y^{N}_{j,i,m,t-1} = \sum_{k=1}^{N} y^{1}_{i,k,m,t}$$

$$i = 1,\ldots,N,\ m = 1,\ldots,M,\ t = 2,\ldots,T \tag{11}$$

$$y^{n}_{ijmt} = 0 \text{ or } 1 \tag{12}$$

$$I_{j,m,t}, x^{n}_{jmt}, q^{n}_{jmt} \geq 0 \tag{13}$$

$$j = 1,\ldots,N,\ m = 1,\ldots,M \tag{14}$$

In this model, Eq. 1 represents the objective function which minimizes the sum of the sequence-dependent setup costs, the storage costs and the production costs. Equation 2 ensures the demand supply in each period. Equation 3 shows that in a network, total of in-flows to each node (j, m, t) is equal to out-flows from that node. In other words, the sum of stock on hand of product j on machine m at the beginning of period t and production volume of product j on machine m during period t is equal to the sum of the stock on hand of product j on machine m at the end of period t and production volume of product j on machine m+1 during period t.

Equation 4 guarantees within one period that product j on machine m is produced before its direct successor (product j on machine m+1).

Left side of Eq. 4 is equal to the time between the beginning of period t and the end of production of product j on machine m if production of product j on machine m can take place between n'th and (n'+1)th setups in period t, else it is a negative number. Right side of Eq. 4 is equal to the time between the beginning of period t and the beginning of production of product j on machine m+1 if production of product j on machine m+1 can take place between n''th and (n''+1)th setups in period t, else it is a big number.

Equation 5 represents the capacity constraints of machines during periods.

Equation 6 indicates that setup is considered in production process.

Equation 7 indicates the relationship between shadow products and setups.

Equations 8 and 9 guarantee that for each machine, the first setup at the beginning of the planning horizon is from a defined product.

Equation 10 and 11 represent the relationship between successive setups.

Equation 12 and 13 represent the type of variables.

Equation 14 indicates that at the beginning of planning horizon there is no on-hand inventory.

DEVELOPMENT OF LOWER BOUNDS

So far, we have successfully formulated the problem. However, the formulation presented in the earlier section is not a practical approach to solve large instances of the problem. Present experiments show that computation time grows exponentially with the number of products, the number of machines and the number of periods in the planning horizon. Therefore, it is necessary to develop a computable lower bound in order to test the accuracy of the heuristics. Note that the heuristic solution is by definition an upper bound on the optimal solution.

Here, we obtain two lower bounds on the problem. First lower bound is achieved by relaxing all binary variables and relaxing Eq. 4. The latter is made by adding the following equation to the first lower bound.

$$\sum_{i=1}^{N} y^{1}_{ijmt} + \sum_{i=1, i\neq j}^{N} \sum_{n=2}^{N} y^{n}_{ijmt} = a_{jmt} \qquad (15)$$

$$a_{jmt} = 0 \text{ or } 1$$

Table 1: Comparison of lower bounds and exact optimal solutions.The values inside the brackets are the computational time in seconds and the percentage values are the difference between the objective values of the lower bound and the original model

No.	Original problem	First lower bound	Second lower bound
	3055.12	2731.28	2946.14
1		10.6%	3.57%
	(157.24)	(0.019)	(0.441)
	3424.44	2825.16	3295.62
2		17.5%	3.76%
	(180.71)	(0.029)	(0.539)
	3105.71	2708.18	3008.53
3		12.8%	3.13%
	(202.27)	(0.031)	(0.37)
	3287.53	2743.30	3156.03
4		16.55%	4%
	(140.23)	(0.027)	(0.491)
	3279.95	2755.21	3129.95
5		16%	4.57%
	(163.94)	(0.029)	(0.397)

Equation 15 is similar to the right hand side of Eq. 6, $\sum_{i=1, i\neq j (for\, n>1)}^{N} y^{n}_{ijmt}$. In Eq. 15, we aggregate $\sum_{i=1, i\neq j (for\, n>1)}^{N} y^{n}_{ijmt}$ by summing over all n. Therefore second relaxation is a lower bound on the original problem.

In order to ascertain the accuracy of these lower bounds, we performed many numerical tests. Table 1 shows the results of such tests in some instances of the problem with N = 3, M = 2 and T = 3.

To apply the Optimal Enumeration Method (OEM) on original problem and lower bounds, GAMS models are provided using GAMS IDE (Ver. 2.0.19.0) and solved using OSL 2. GAMS models are run on a personal computer with a Pentium 4 processor running at 3.4 GHz. The required parameters for these problems are extracted from the following uniform distributions:

$b_{j,m} \approx U(1.5,2)$, $d_{j,t} \approx U(0,180)$, $h_{j,m} \approx U(0.2,0.4)$, $p_{j,m,t} \approx U(1.5,2)$, $W_{ijm=} = S_{ijm} \approx U(35,70)$, $C_{m,t} \approx U(200,300)$.

Table 1 confirms the advantages of the second lower bound, it is therefore used.

SOLUTION METHOD

Solving the original model

Solution procedure: Rolling-horizon heuristics have been used to overcome computational infeasibility for large MIP problems by substituting most of the binary variables and constraints with continuous variables and constraints (Beraldi *et al*., 2008; Araujo *et al*., 2007; Araujo *et al*., 2008; Merce and Fontan, 2003; Clark, 2003; Clark and Clark, 2000). The approach initially adopted decomposes the model into a succession of smaller MIPs, each with a more tractable number of binary variables.

Relaxing all binary variables and determining the setup pattern of current period by a heuristic instead of solving a MIP is our approach to solve this hard-to-solve MIP problem.

Present rolling-horizon approach decomposes the planning horizon into three sections. For a given iteration k:

- The first section (beginning section) is composed of the (k-1) first periods. Within this section, setup pattern have been frozen by the previous iterations
- The second section (central section) includes the kth period. In this section the whole relaxed problem is considered and setup pattern of this period is determined by heuristic
- The third section (ending section) includes the last periods (from period k+1 to period T). There, the model is simplified according to a selected simplification strategy.

Each iteration consists of solving a linear programming problem. At the end of iteration k, all sections roll one period and a new iteration can then be performed. The procedure stops when there is no longer any ending section. The last iteration defines decision variables over the entire horizon.

Setup state selection rule for current period: According to this method, all binary variables of current period would be determined. Note that according to Eq. 8-11 for each triple (n, m, t) there is exactly one pair (i, j) for which $y^{n}_{ijmt} = 1$ and for $(i', j') \neq (i, j)$, $y^{n}_{i'j'mt} = 0$. In other words, for each triple (n, m, t) of current period, this heuristic specifies a pair (i, j) which $y^{n}_{ijmt} = 1$.

According to our method, if there is sufficient amount of inventory to satisfy demand of current period ($I_{j,m,t-1} > d_{j,t}$), product j in stage m would not be produced in current period.

The part of heuristic that is used to determine ordering of products in current period is similar to (g/2,g/2) Johnson's rull-based heuristic has been used by Kurz and Askin (2003, 2004) to schedule flexible flow lines with sequence-dependent setups.

To order products in stages of current period, it is necessary to define job duration. Duration of a job in each stage is, by definition, the minimum time to produce demand of current period. These job durations only are used to determine sequences of products in current period and would not be used as lotsizes.

Let [i] indicates the ith job in an ordered sequence in the following. In the following heuristic, a job duration is used. For product j in stage m and period t it is specified by D (j, m, t) and is defined as:

$$D(j, m, t) = D(j, m-1, t) + d(j, t).b(j, m) + \min_i \{S_{ijm}\}$$

Simplification strategy: More computatinal time is economized by eliminating the majority of variables. y^{n}_{ijmt} (n > 1), x^{n}_{jmt} (n > 1) and q^{n}_{jmt} are eliminated. Except Eq. 2, 3 and 5, other constraints are ignored for ending section. All setup costs (and times) for ending section are assumed to be 0.

$b_{j,m}$ and $p_{j,m,t}$ should be modified to estimate the capacity consumption of future setups. We assume A_1 is the objective value of the second lower bound and A_2 is the objective value of the original problem by relaxing all binary variables, ignoring Eq. 4 and all setup costs and times equal to 0. we would replace $b_{j,m}$ and $p_{j,m,t}$ with $b'_{j,m}$ and $p'_{j,m,t}$ as follows:

$$b'_{j,m} = (A_1 / A_2) \cdot b_{j,m}$$
$$p'_{j,m,t} = (A_1 / A_2) \cdot p_{j,m,t}$$

A simplified representation for latter periods in the rolling horizon is less difficult to solve and hence permits the solution of larger problems.

The whole algorithm (H1)

All binary variables are relaxed

Begin:

$end^{m}_{0} = j_{0m}$

$t \leftarrow 1$

while $t \leq T$ loop

for tt>t

- relax Eq. 4, 10 and 11
- y^{n}_{ijmt} for (n>1), x^{n}_{jmt} for (n>1) and q^{n}_{jmt} are equal to 0
- $p'_{j,m,tt}$ and $b'_{j,m}$ are used instead of $p_{j,m,tt}$ and $b_{j,m}$ (for tt>t)

In current period (t)

Create job durations D(j, m, t) as follows:

If $I_{j,m,t-1} > d_{j,t}$; D(j, m, t) = 0

else if $j \neq end^{m}_{t-1}$, $D(j, m, t) = D(j, m-1, t) + d_{j,t}.b_{j,m} + \min_{i(i \neq j)} \{S_{ijm}\}$;

else $D(j, m, t) = D(j, m-1, t) + d_{j,t}.b_{j,m}$;

Create D^{jt}_{1} and D^{jt}_{M} as follows: $D^{jt}_{1} = \sum_{m=1}^{[M/2]} D(j,m,t)$, $D^{jt}_{M} = \sum_{m=[M/2]+1}^{M} D(j,m,t)$;

Let $U = j \mid D^{jt}_{1} < D^{jt}_{M}$ and $V = j \mid D^{jt}_{1} \geq D^{jt}_{M}$. The set U is the set of jobs that moves through stages 1 to [M/2] faster than they move through stages [M/2]+1 to M. The set V is the set of jobs that moves through stages [M/2]+1 to M faster than they move through stages 1 to [M/2].

Arrange jobs in U in non-decreasing order of D_1^{jt} and arrange jobs in V in non-increasing order of D_M^{jt}. Append the ordered list V to the end of U, creating an ordered list to use in the next step. Delete jobs which $I_{j,l,t-1}>d_{j,t}$. Repeat the last member of the list unless the list contains N members.

If i = 1, $y^i_{end^1_{t-1},[i],1,t}=1$, else, $y^i_{[i-1],[i],1,t}=1$;

m←2
while m≤M loop

Order the jobs in non-decreasing order (SPT) of D(j, m, t). Delete jobs which $I_{j,l,t-1}\geq d_{j,t}$. Instead of deleted jobs, we replace the last true job.

if i = 1, $y^i_{end^m_{t-1},[i],m,t}=1$, else, $y^i_{[i-1],[i],m,t}=1$
m ←m+1
end loop
Solve LP and fix y^n_{ijmt} for current period (t).
For t >1, i end^m_t s the last sequenced job in stage m and period t.
t← t+1
end loop.

Solving the modified model

The modified model: Reducing the dimensionality of the problem and allowing to solve larger instances, a modified mathematical model is developed which ignores the majority of combinations. In this model, in each period, products have the same sequence and size in all stages.

Following proposes the modified model:

$$\text{Min}\sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T}w_{ijm}\cdot y^n_{ijt}+\sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T}p_{jmt}\cdot x^n_{jt}+\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T}h_{jm}\cdot I_{j,t-1} \quad (16)$$

Subject to:

$$d_{j,t}=I_{j,t-1}+\sum_{n=1}^{N}x^n_{jt}-I_{j,t}$$

$$j = 1,\dots,N,\ t = 1,\dots,T \quad (17)$$

$$\text{bigM}.(\sum_{i=1,i\neq j(\text{for } n'>1)}^{N}y^{n'}_{ijt}-1)+\sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N}y^n_{ijt}S_{ijm}+\sum_{n=1}^{n'}\sum_{j=1}^{N}b_{jm}\cdot q^n_{jmt}+\sum_{n=1}^{n'}\sum_{j=1}^{N}b_{jm}\cdot x^n_{jt}\leq\sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N}y^n_{ijt}S_{i,j,m+1}+\sum_{n=1}^{n'}\sum_{j=1}^{N}b_{j,m+1}\cdot q^n_{j,m+1,t}+\sum_{n=1}^{n'-1}\sum_{j=1}^{N}b_{j,m+1}\cdot x^n_{j,t}$$

$$n' = 1,\dots,N,\ m = 1,\dots,M-1,\ t = 1,\dots,T \quad (18)$$

$$\sum_{n=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N}y^n_{ijt}S_{ijm}+\sum_{n=1}^{N}\sum_{j=1}^{N}b_{jm}\cdot x^n_{jt}+\sum_{n=1}^{N}\sum_{j=1}^{N}b_{jm}\cdot q^n_{jmt}\leq C_{m,t}$$

$$m = 1,\dots,M,\ t = 1,\dots,T \quad (19)$$

$$x^n_{jt}\leq(\text{bigM}).\sum_{i=1,i\neq j(\text{for } n>1)}^{N}y^n_{ijt}$$

$$n = 1,\dots,N,\ j = 1,\dots,N,\ t = 1,\dots,T \quad (20)$$

$$q^n_{jmt}\leq(C_{m,t}/b_{j,m})\cdot\sum_{i=1}^{N}y^n_{ijt}$$

$$n = 1,\dots,N,\ j = 1,\dots,N,\ m = 1,\dots,M,\ t = 1,\dots,T \quad (21)$$

$$y^1_{jil}=0$$

$$j\neq j_0,\ i = 1,\dots,N \quad (22)$$

$$\sum_{i=1}^{N}y^1_{j0,i,1}=1 \quad (23)$$

$$\sum_{j=1}^{N}y^n_{ijt}=\sum_{k=1}^{N}y^{n+1}_{ikt}$$

$$n = 1,\dots,N-1,\ i = 1,\dots,N,\ t = 1,\dots,T \quad (24)$$

$$\sum_{j=1}^{N}y^N_{j,i,t-1}=\sum_{k=1}^{N}y^1_{i,k,t}$$

$$i = 1,\dots,N,\ t = 2,\dots,T \quad (25)$$

$$y^n_{ijt}=0 \text{ or } 1 \quad (26)$$

$$I_{j,t},x^n_{jt},q^n_{jmt}\geq 0 \quad (27)$$

$$I_{j,0}=0$$

$$j = 1,\dots,N \quad (28)$$

The whole algorithm (H2)

All binary variables are relaxed
Begin:

$end_0 = j_0$
t←1
while t≤T loop
for tt >t

- relax Eq. 18, 24 and 25
- y^{n}_{ijt} for (n>1), x^{n}_{jt} for (n>1) and q^{n}_{jmt} are equal to 0
- $p'_{j,m,tt}$ and $b'_{j,m}$ are used instead of $P_{j,m,tt}$ and $b_{j,m}$ (for tt>t)

In current period (t)
Create job durations D (j, m, t) as follows:
If $I_{j,t-1}>d_{j,t}$, D (j, m, t) = 0
else if $j \neq end_{t-1}$ $D(j,m,t) = D(j, m-1, t)+d_{j,t}.+b_{j,m}+\min_{i(i\neq j)}\{S_{ijm}\}$
Else $D(j, m, t) = D(j, m-1, t)+d_{j,t}.b_{j,m}$;
Create D^{jt}_{1} and D^{jt}_{M} as follows:

$$D^{jt}_{1} = \sum_{m=1}^{[M/2]} D(j,m,t),\ D^{jt}_{M} = \sum_{m=[M/2]+1}^{M} D(j,m,t)$$

Let U = j | $D^{jt}_{1} < D^{jt}_{M}$ and V = j | $D^{jt}_{1} \geq D^{jt}_{M}$. The set U is the set of jobs that moves through stages 1 to [M/2] faster than they move through stages [M/2]+1 to M. The set V is the set of jobs that moves through stages [M/2]+1 to M faster than they move through stages 1 to [M/2].

Arrange jobs in U in non-decreasing order of D^{jt}_{1} and arrange jobs in V in non-increasing order of D^{jt}_{M}. Append the ordered list V to the end of U, creating an ordered list to use in the next step. Delete jobs which $I_{j,t-1}>d_{j,t}$. Repeat the last member of the list unless the list contains N members.

If i = 1, $y^{i}_{end_{t-1}[i],t} = 1$, else, $y^{i}_{[i-1],[i],t} = 1$
Solve LP and fix y^{n}_{ijt} for current period (t).
For t>1, end_{t-1} is the last sequenced job in period t.
t←t+1
end loop.

NUMERICAL EXPERIMENTS

To evaluate and compare the performance of developed heuristics, 20 problems with different sizes are selected to test. For each problem size, 5 problem instances are randomly generated and the required parameters for these problems are extracted from the following Uniform distributions:

$b_{j,m}\approx U(1.5,2)$, $d_{j,t}\approx U(0,180)$, $h_{j,m}\approx U(0.2,0.4)$, $p_{j,m,t}\approx U(1.5,2)$, $W_{i,j,m}= S_{i,j,m}\approx U(35,70)$

$C_{m,t}$ is calculated in accordance to satisfy demand of each period on a just-in-time basis with average setups.

The heuristics are coded in Matlab programming language and are run on a personal computer with a Pentium IV, with a 3.4 GHz processor and a 4 GB of RAM.

Table 2: Comparison between objective functions of the second lower bound and heuristics

Problem size (NMT)	No. of problem solved	The second lower bound	H1	H2
3×3×3	5	4677.76	5587.17	5757.65
5×3×3	5	7796.25	9335.91	9728.32
3×5×3	5	7725.37	9282.13	9547.97
3×3×5	5	7902.56	9473.04	9833.63
5×5×5	5	21956.29	26799.36	27905.79
7×5×5	5	29318.73	35542.56	36354.41
5×7×5	5	28167.51	34387.49	35743.21
5×5×7	5	29370.11	35121.41	36892.78
7×7×7	5	57964.97	---	71471.95
10×5×5	5	42312.54	---	52961.98
5×10×5	5	41025.09	49776.03	52390.31
5×5×10	5	42493.76	52033.48	53999.85
10×7×7	5	90294.74	---	112773.44
7×10×7	5	88313.43	---	111876.61
7×7×10	5	88207.71	---	113353.46
10×10×10	5	181125.19	---	---
15×10×10	5	285862.42	---	---
10×15×10	5	276412.35	---	---
10×10×15	5	269325.31	---	---
15×15×15	5	664453.13	---	---

---: Indicates that there is not enough memory to solve this instance

Table 3: Comparison between CPU times of the second lower bound and heuristics. The values inside the brackets are the computational times in seconds

Problem size (NMT)	No. of problem solved	The second lower bound	H1	H2
3×3×3	5	2.33	0.21	0.08
5×3×3	5	46.36	5.35	0.89
3×5×3	5	18.22	0.87	0.24
3×3×5	5	8.11	1.09	0.43
5×5×5	5	117.73	37.11	5.61
7×5×5	5	481.77	479.17	51.79
5×7×5	5	218.12	72.98	13.42
5×5×7	5	165.12	74.39	8.35
7×7×7	5	3413.14	---	375.67
10×5×5	5	2190.52	---	811.27
5×10×5	5	1616.31	211.46	23.91
5×5×10	5	365.11	188.47	81.37
10×7×7	5	>7200*	---	5886.47
7×10×7	5	>7200*	---	501.86
7×7×10	5	>7200*	---	1538.37
10×10×10	5	>7200*	---	---
15×10×10	5	>7200*	---	---
10×15×10	5	>7200*	---	---
10×10×15	5	>7200*	---	---
15×15×15	5	>7200*	---	---

*: Indicates that finding the optimum value for the second lower bound requires more than 7200 seconds and the objective function at this time has been considered, ---: Indicates that there is not enough memory to solve this instance

Table 2 and 3 compare the objective functions and cpu times of heuristics and the second lower bound.

In accordance to the experiments there is not enough memory to solve instances with more than 5 products through the first algorithm (H1). The second algorithm (H2) is also able to solve instances with 10 products.

CONCLUSIONS AND RECOMMENDATION FOR FUTURE STUDIES

The contribution of the study has been to derive and test two models and two heuristics for multi-product multi-level capacitated lotsizing problem with sequence-dependent setups.

Heuristic H1 is based on the original model and is able to solve only small size problems. Heuristic H2 is based on the modified model that assumes similar sequences and sizes of products in all machines. H2 is able to solve medium size problems.

According to the experiments H1 is superior for small size problems. For larger problems that H1 is not able to solve, H2 is used.

Because of the expanding role of meta-heuristic approaches to solve complicated lotsizing problem (Jans and Degraves, 2007; Defersha and Chen, 2008), the application of meta-heuristic approaches to face this hard to solve problem is recommended as an area for future research.

REFERENCES

Araujo, S.A., M.N. Arenales and A.R. Clark, 2007. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. J. Heuristics, 13: 337-358.

Araujo, S.A., M.N. Arenales and A.R. Clark, 2008. Lot sizing and furnace scheduling in small foundries. Comput. Operat. Res., 35: 916-932.

Beraldi, P., G. Ghiani, A. Grieco and E. Guerriero, 2008. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot sizing and scheduling problem with sequence-dependent set-up costs. Comput. Operat. Res., 35: 3644-3656.

Clark, A.R. and S.J. Clark, 2000. Rolling-horizon lot sizing when setup times are sequence-dependent. Int. J. Prod. Res., 38: 2287-2307.

Clark, A.R., 2003. Optimization approximations for capacity constrained material requirements planning. Int. J. Prod. Econ., 84: 115-131.

Defersha, F.M. and M. Chen, 2008. A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality. Eur. J. Operat. Res., 187: 46-69.

Eppen, G.D. and R.K. Martin, 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. Operat. Res., 35: 832-848.

Fandel, G. and C. Stammen-Hegene, 2006. Simultaneous lot sizing and scheduling for multi-product multi-level production. Int. J. Prod. Econ., 104: 308-316.

Fleischmann, B. and H. Meyr, 1997. The general lotsizing and scheduling problem. OR Spekt., 19: 11-21.

Gupta, D. and T. Magnusson, 2005. The capacitated lot sizing and scheduling problem with sequence-dependent setup costs and setup times. Comput. Operat. Res., 32: 727-747.

Jans, R. and Z. Degraeve, 2007. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. Eur. J. Operat. Res., 177: 1851-1875.

Karimi, B., S.M.T. Fatemi Ghomi and J.M. Wilson, 2003. The capacitated lot sizing problem: A review of models and algorithms. Omega, 31: 365-378.

Kimms, A., 1996. Multi-level,single-machine lotsizing and scheduling (with initial inventory). Eur. J. Operat. Res., 89: 86-99.

Kurz, M.E. and R.G. Askin, 2003. Comparing scheduling rules for flexible flow lines. Int. J. Prod. Econ., 85: 371-388.

Kurz, M.E. and R.G. Askin, 2004. Scheduling flexible flow lines with sequence-dependent setup times. Eur. J. Operat. Res., 159: 66-82.

Merece, C. and G. Fonton, 2003. MIP-based heuristics for capacitated lotsizing problems. Int. J. Prod. Econ., 85: 97-111.

Wagner, H.M. and T.M. Whithin, 1958. Dynamic version of the economic lot size model. Manage. Sci., 5: 89-96.