Associate Professor Mohammad Mohammadi, PhD
E-mail: mohammadi@khu.ac.ir (corresponding author)
Department of Industrial Engineering, Faculty of Engineering
Kharazmi University, Tehran, Iran

THE TRADEOFF BETWEEN OUTSOURCING AND USING MORE FACTORIES IN A DISTRIBUTED FLOW SHOP SYSTEM

Abstract. *Simultaneous lot-sizing and scheduling in a distributed flow shop has been considered in this paper. Number of factories is a parameter and it is not necessary to use all factories. Outsourcing is the other realistic assumption of this paper. In other words, the tradeoff between outsourcing and number of utilized factories is discussed in this paper. An exact formulation of the problem has been provided as a mixed integer program. To solve the problem, a heuristic has been developed.*

Keywords*: Outsourcing; Distributed flow shop; Lot-sizing and scheduling; Assignment; Sequence-dependent.*

JEL Classification: C61, D24, O14

1. Introduction

Selection the appropriate lot-sizes and production schedules has an undeniable role in running the production organizations. Lot-sizing and scheduling problems have been an active area of research starting from the seminal paper of Wagner & Whithin (1958). Along with the setup issue, there are various other features to be dealt with when modeling lot-sizing and scheduling problems, among them 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 (Mohammadi et al., 2010; Fandle and Stammen-Hegene, 2006; Karimi et al., 2003; Merece and Fonton, 2003).

Because of the interrelationship between lot-sizing and scheduling, simultaneous lot-sizing and scheduling is essential when sequence-dependent setup costs and times occur. Involving sequence-dependent setups is an important consideration in many practical applications. Because of their prevalence in, and importance to, industry and because of the challenges they present to solution methodologies, lot-sizing and scheduling problems that involve a sequence-dependent setup have attracted attention from many researchers (Fandle and Stammen-Hegene, 2006).

Chung and Choi (2013) considered a two-machine ordered flow shop problem, where each job is processed through the in-house system or outsourced to a subcontractor. For in-house jobs, a schedule is constructed and its performance is measured by the makespan. Since this problem is NP-hard, they presented an approximation algorithm. Naderi and Ruiz (2010) presented a new generalization of the regular permutation flow shop scheduling problem (PFSP) that referred to as the distributed permutation flow shop scheduling problem or DPFSP. Also, a comprehensive computational and statistical analysis are conducted in order to analyze the performance of the proposed methods. Zandieh and Rashidi (2009) presented an effective hybrid genetic algorithm for hybrid flow shops with sequence dependent setup times and processor blocking. Also, Azab and Naderi (2014) presented greedy heuristics for distributed job shop problems. The idea of the proposed heuristics is to iteratively insert operations (one at each iteration) into a sequence to build up a complete permutation of operations. The performance of the model and the six heuristics are comprehensive evaluated by numerical experiments. In other research, Sukkerd and Wuttipornpun (2016) proposed a Hybrid genetic algorithm and tabu search for finite capacity material requirement planning system in flexible flow shop with assembly operations. The results of this paper show that HGATS requires a practical computational time when applied to real industrial cases.

Simultaneous lot-sizing and scheduling in capacitated flow shop with sequence-dependent setups has been considered by Mohammadi et al. (2010). They proposed a mathematical formulation and mixed integer programming based heuristics for the problem. Costa et al. (2015) presented a hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem. The proposed technique makes use of a matrix encoding able to simultaneously manage the sequence of jobs within each group and the sequence of groups to be processed along the flow-shop manufacturing system. In other study, to solve large-sized instances of the problems, Mohammadi et al. (2010) presented a new algorithmic approach and a genetic algorithm, respectively. Also, in other research a genetic algorithm-based heuristic for capacitated lot-sizing problem in flow shops with sequence-dependent setups have been presented (Mohammadi and Fatemi Ghomi, 2011). In single factory problems, the problem is to lot-size and schedule products on a set of serially arranged machines, while in the distributed problem another additional decision arises: the allocation of the products to factories (Naderi and Ruiz, 2010; Moon et al., 2002).

This paper is organized as follows: section 2 introduces a detailed description of the problem. Sections 3 & 4 present the heuristic and numerical experiments, respectively. The last section is devoted to the final remarks and recommendation for future studies.

2. Problem formulation

2.1. Assumptions

- There are a specific number of factories. It is not necessary to utilize all of the factories.
- Each factory consists of m serially-arranged machines. Each product can not be produced in more than one factory.

- External demands exist just for final products. Demand of a specific product could be satisfied by outsourcing or production on a factory.
- In each factory production permutation is not necessarily maintained the same in each period.
- When the machines are setup, sequence-dependent setup costs and times occur.
- The setting up of a machine must be completed in a period.
- Shortage is not permitted.
- A component can not be produced in a period until the production of its required component is finished. In other words, production at a level can only be started if a sufficient amount of the required items from the previous level are available; this is called vertical interaction.
- To guarantee the vertical interaction, idleness between each production and its respective setup is defined by introducing the shadow product (Mohammadi et al., 2010, Fandle and Stammen-Hegene, 2006).
- At the beginning of the planning horizon, all machines are in starting setup configuration.
- 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.

2.2. Mathematical formulation

The following notations are used in the model formulation:

Indices:

i, j, k: Product type

n, n', n'': Designation for a specific setup number

m: Level of production

f: A specific factory

t: Period

Parameters:

T : Planning horizon.

N : Number of different products.

M : Number of production levels/ number of machines.

$Big\ M$: A large real number.

F : Number of factories.

$C_{m,f,t}$: Available capacity of machine *m* at factory *f* in period *t*.

$d_{j,t}$: External demand for product *j* at the end of period *t*.

$h_{j,m}$: Storage costs unit rate for product *j* in level *m*.

$b_{j,m,f}$: Production time to produce one unit of product *j* on machine *m* of factory *f*.

$p_{j,m,f,t}$: Production cost to produce one unit of product *j* on machine *m* of factory *f* in period *t*.

$S_{i,j,m,f}$: Sequence-dependent setup time for the setup of the machine *m* in factory *f* from production of product *i* to production of product *j*; for $i \neq j, S_{i,j,m,f} \geq 0$ and for $i = j, S_{i,j,m,f} = 0$.

$W_{i,j,m,f}$: Sequence-dependent setup cost for the setup of the machine *m* in factory *f* from production of product *i* to production of product *j*; for $i \neq j, W_{i,j,m,f} \geq 0$ and for $i = j, W_{i,j,m,f} = 0$.

A_f : Fixed cost to stabilize factory *f*.

0 : The starting setup configuration on machines.

$O_{j,t}$: Cost of outsourcing one unit of product *j* in period *t*.

Variables:

$y^n_{i,j,m,f,t}$: Binary variable, which indicates whether the *nth* setup on machine *m* & factory *f* in period *t* is from product *i* to product *j* ($y^n_{i,j,m,f,t} = 1$) or not ($y^n_{i,j,m,f,t} = 0$).

$x^n_{j,m,f,t}$: Quantity of product *j* produced after *nth* setup on machine m & factory *f* in period *t*.

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

$q^n_{j,m,f,t}$: Shadow product: the gap (in quantity units) between nth setup of factory *f* (to product *j*) on machine *m* in period *t* and its related production in order to ensure that direct predecessor of this product (production of product *j* on machine *m-1* from factory *f* in period *t*) has been completed.

F_f : Binary variable, if factory *f* has been selected for operation ($F_f = 1$) in other words ($F_f = 0$).

$o_{j,t}$: Quantity of product *j* which are outsourced in period *t*.

$$Min\, Z = \sum_{n=1}^{N}\sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{t=1}^{T} W_{i,j,m,f} \cdot y^{n}_{i,j,m,f,t} + \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{t=1}^{T} p_{j,m,f,t} \cdot x^{n}_{j,m,f,t} + \sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} h_{j,m} \cdot I_{j,m,t} + \sum_{f=1}^{F} A_f \cdot F_f + \sum_{j=1}^{N}\sum_{t=1}^{T} o_{j,t} \cdot O_{j,t} \quad \textbf{(1)}$$

Subject to

$$d_{j,t} = o_{j,t} + \sum_{n=1}^{N}\sum_{f=1}^{F} x^{n}_{j,M,f,t} + I_{j,M,t-1} - I_{j,M,t} \qquad (j=1,\dots,N; t=1,\dots,T) \quad \textbf{(2)}$$

$$I_{j,m,t-1} + \sum_{n=1}^{N}\sum_{f=1}^{F} x^{n}_{j,m,f,t} = I_{j,m,t} + \sum_{n=1}^{N}\sum_{f=1}^{F} x^{n}_{j,m+1,f,t} \quad \textbf{(3)}$$
$$(j=1,..,N; m=1,..,M-1; t=1,\dots,T)$$

$$bigM.\Big(\sum_{i=0,(i\neq j)\&(i>0)\ for\, n'>1}^{N} y^{n'}_{i,j,m,f,t} - 1\Big) + \sum_{n=1}^{n'}\sum_{i=0}^{N}\sum_{k=1}^{N} y^{n}_{i,k,m,f,t} \cdot S_{i,k,m,f} + \sum_{n=1}^{n'}\sum_{k=1}^{N} b_{k,m,f} \cdot q^{n}_{k,m,f,t} + \sum_{n=1}^{n'}\sum_{k=1}^{N} b_{k,m,f} \cdot x^{n}_{k,m,f,t}$$
$$\le bigM.\Big(1 - \sum_{i=0,(i\neq j)\&(i>0)\ for\, n''>1}^{N} y^{n''}_{i,j,m+1,f,t}\Big) + \sum_{n=1}^{n''}\sum_{i=0}^{N}\sum_{k=1}^{N} y^{n}_{i,k,m+1,f,t} \cdot S_{i,k,m+1,f} \quad \textbf{(4)}$$
$$+ \sum_{n=1}^{n''}\sum_{k=1}^{N} b_{k,m+1,f} \cdot q^{n}_{k,m+1,f,t} + \sum_{n=1}^{n''-1}\sum_{k=1}^{N} b_{k,m+1,f} \cdot x^{n}_{k,m+1,f,t}$$
$$(j=1,\dots,N; n', n''=1,\dots,N; m=1,\dots,M-1; f=1,\dots,F; t=1,\dots,T)$$

$$\sum_{n=1}^{N}\sum_{i=0}^{N}\sum_{j=1}^{N} y^{n}_{i,j,m,f,t} \cdot S_{i,j,m,f} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{j,m,f} \cdot x^{n}_{j,m,f,t} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{j,m,f} \cdot q^{n}_{j,m,f,t} \le C_{m,f,t} \quad \textbf{(5)}$$
$$(m=1,\dots,M; f=1,\dots,F; t=1,\dots,T)$$

$$x^{n}_{j,m,f,t} \le (Big\ M). \sum_{i=0,(i\neq j)\&(i>0)(for\, n>1)}^{N} y^{n}_{i,j,m,f,t} \quad \textbf{(6)}$$
$$(n=1,\dots,N;\ j=1,\dots,N; m=1,\dots,M;\ f=1,\dots,F; t=1,\dots,T)$$

$$q^{n}_{j,m,f,t} \le (Big\ M). \sum_{i=1}^{N} y^{n}_{i,j,m,f,t} \quad \textbf{(7)}$$
$$(n=1,\dots,N;\ j=1,\dots,N; m=1,\dots,M; f=1,\dots,F; t=1,\dots,T)$$

$$y^{1}_{j,i,m,f,1} = 0 \quad (j \neq 0; i = 1,\dots,N; m = 1,\dots,M; f = 1,\dots,F) \qquad \textbf{(8)}$$

$$\sum_{i=0}^{N} y^{1}_{0,i,m,f,1} = 1 \quad (m = 1,\dots,M; \; f = 1,\dots,F) \qquad \textbf{(9)}$$

$$y^{n}_{j,0,m,f,t} = y^{n+1}_{0,0,m,f,t}$$
$$(n = 1,\dots,N-1; j = 0,\dots,N; m = 1,\dots,M; f = 1,\dots,F; t = 1,\dots,T) \qquad \textbf{(10)}$$

$$y^{N}_{0,0,m,f,t-1} = y^{1}_{0,0,m,f,t}$$
$$(m = 1,..,M; f = 1,\dots,F; t = 2,\dots,T) \qquad \textbf{(11)}$$

$$\sum_{j=0}^{N} y^{n}_{j,i,m,f,t} = \sum_{k=0}^{N} y^{n+1}_{i,k,m,f,t}$$
$$(n = 1,\dots,N-1; j = 0,\dots,N; m = 1,\dots,M; \; f = 1,\dots,F; t = 1,\dots,T) \qquad \textbf{(12)}$$

$$\sum_{j=0}^{N} y^{N}_{j,i,m,f,t-1} = \sum_{k=0}^{N} y^{1}_{i,k,m,f,t}$$
$$(n = 1,\dots,N-1; j = 0,\dots,N; m = 1,\dots,M; f = 1,\dots,F; t = 1,\dots,T) \qquad \textbf{(13)}$$

$$\sum_{n=1}^{N} \sum_{i=0}^{N} \sum_{f=1}^{F} y^{n}_{i,j,m,f,1} \leq 1$$
$$(j = 1,\dots,N; m = 1,\dots,M) \qquad \textbf{(14)}$$

$$\sum_{n=1}^{N} \sum_{i=0}^{N} \sum_{t=2}^{T} y^{n}_{i,j,m,f,t} \leq (Big\ M).\sum_{n=1}^{N} \sum_{i=0}^{N} y^{n}_{i,j,m,f,1}$$
$$(j = 1,\dots,N; m = 1,\dots,M; f = 1,\dots,F) \qquad \textbf{(15)}$$

$$F_{f} \leq \sum_{n=1}^{N} \sum_{i=0}^{N} \sum_{j=1}^{N} y^{n}_{i,j,1,f,1}$$
$$(f = 1,\dots,F) \qquad \textbf{(16)}$$

$$y^{n}_{i,j,m,f,t} \in \{0,1\}$$
$$(n = 1,\dots,N; \; i, j = 0,\dots,N; m = 1,\dots,M; \; f = 1,\dots,F; t = 1,\dots,T) \qquad \textbf{(17)}$$

$$I_{j,m,t}, x^{n}_{j,m,f,t}, q^{n}_{j,m,f,t}, o_{j,t} \geq 0$$
$$(n = 1,...,N;\ j = 0,...,N;\ f = 1,...,F; t = 1,...,T) \qquad \textbf{(18)}$$

$$I_{j,m,0} = 0 \qquad (j = 1,...,N; m = 1,...,M) \qquad \textbf{(19)}$$

Objective function which minimizes the sum of the sequence-dependent setup costs, the production costs, the storage costs, the costs of stabilizing factories and the costs of outsourcing the products are shown in equation (1). Note that when in equation (2), the demand, supply in each period are presented. Equation (3) ensures that in a network, total of in-flows to each node is equal to out-flows from that node. Equation (4) determine that each factory guarantees within one period each typical product *j* on machine *m* is produced before its direct successor (product *j* on machine *m+1*). 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) & (9) in each factory guarantee for each machine, the first setup at the beginning of the planning horizon is from the starting setup configuration. Equations (10) & (11) for each factory indicates that if destination of a setup is "*0*", successive setups are from "*0*" to "*0*". Equations (12) & (13) represent the relationship between successive setups. Equations (8) to (13) ensure that for each quadruplet *(n,m,f,t)* there is exactly one pair *(i,j)* which $y^{n}_{i,j,m,f,t} = 1$. Equations (14) & (15) ensure that each product can not be produced in more than one factory. Equation (16) indicates that a factory is used if at least one product produced on it. Equations (17) & (18) represent the type of variables. Equation (19) indicates that at the beginning of the planning horizon there is no on-hand inventory.

2.3. Development of lower bound

The formulation presented in the previous sub-section is not a practical approach to solve large instances of the problem. Solving the single-level multi-period CLSP with sequence-dependent setups is equivalent to solving multiple-dependent TSPs. Hence, like the TSP, the CLSP 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 (Mohammadi et al., 2010; Almada-Lobo et al., 2007). The introduction of new assumptions makes the problem even more complicated. Therefore, it is necessary to find appropriate solution for medium and large instances. Also it is important to develop a computable lower bound in order to test the accuracy of the heuristic.

Model M_1 has been assumed to be obtained from the initial model by relaxing all binary variables. After relaxing the binary variables, Equation (4) does not have important role on the problem because for non-integer values of $y^{n}_{i,j,m,f,t}$, the left side of Equation (4) would be negative and right side of Equation (4) would be a big number. It means by relaxing binary variables, Equation (4) could be ignored. Model M_2 is obtained by adding the following Equation to M_1 :

$$\sum_{n=1}^{N} \sum_{i=0,(i\neq j)\&(i>0)\ for\, n>1}^{N} y^{n}_{i,j,m,f,t} = a_{j,m,f,t} \qquad \textbf{(20)}$$

$$(j=1,...,N; m=1,...,M;\ \mathrm{f}=1,...,\mathrm{F}; \mathrm{t}=1,...,\mathrm{T})$$

$a_{j,m,f,t}$ is a binary variable. Equation (20) is similar to the right hand side of Equation (6). Equation (20) is valid to the initial model, by considering Equations (14) & (17), $\sum_{n=1}^{N} \sum_{i=0,(i\neq j)\&(i>0)\ for\, n>1}^{N} y^{n}_{i,j,m,f,t}$ cannot get other values except 0 & 1.

3. Development of a heuristic

The proposed heuristic has been composed of 4 sub-sections which are described as follows:

3.1. Decision about the products

The heuristic starts by dividing products *(j=1,…,N)* in two classes. Products in class 1 would be produced and the rest of products would be outsourced. Dividing procedure could be described as follows:

For *j=1 to N*, p_j has been calculated:

$$p_j = \frac{\sum_{f=1}^{F} A_f}{F} + T.\sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{f=1}^{F}\frac{W_{i,j,m,f}}{F} + \frac{\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{t=1}^{T}(p_{j,m,f,t}\cdot d_{j,t})}{F} - \sum_{t=1}^{T}(d_{j,t}\cdot O_{j,t})$$

p_j could be described as the difference between the average cost of producing product *j* and the cost of outsourcing the mentioned product. Products with negative values of $p_j\,(p_j < 0)$ would be selected to be produced. Products would be renamed. Products with negative values of p_j are named from 1 to *N'* and the rest of products are named from *N'+1* to *N*,

3.2. Selection some factories

The heuristic continues by dividing factories *(f=1,...,F)* in two classes. Factories in class 1 would be utilized ($F_f = 1$) and the rest of factories would not be utilized. This procedure is as follows:

For *f=1* to *F*, C_f has been calculated:

$$C_f = A_f + T.\sum_{i=0}^{N'}\sum_{j=1}^{N'}\sum_{m=1}^{M} W_{i,j,m,f} + \sum_{j=1}^{N'}\sum_{m=1}^{M}\sum_{t=1}^{T}(p_{j,m,f,t}.d_{j,t}) - \frac{\sum_{f=1}^{F} A_f}{F}$$
$$-\frac{T}{F}\sum_{i=0}^{N'}\sum_{j=1}^{N'}\sum_{m=1}^{M}\sum_{f=1}^{F} W_{i,j,m,f} - \frac{1}{F}\sum_{j=1}^{N'}\sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{t=1}^{T}(p_{j,m,f,t}.d_{j,t})$$

Factories with negative values of C_f ($C_f < 0$) would be selected to be utilized. C_f could be described as the difference between the average cost of utilizing factory *f* and the average cost of utilizing all factories when lot-for-lot policy has been considered. Factories would be renamed. Factories with negative values of C_f are named from *1* to *f'* and the rest of factories are named from *f'+1* to *F*.

3.3. Assignment the products to factories

For *(f=1 to f')* & for *(j=1 to N')* calculate $W_{j,f}$ as follows:

1. $W_{j,f} = \sum_{i=0}^{N'}\sum_{m=1}^{M} W_{i,j,m,f}$
2. $W_{j,f}$ are sorted in non-decreasing order
3. Consider the lowest $W_{j,f}$: assign product *j* to factory *f*
4. Delete all $W_{j,f}$ for assigned product *(j)*. If number of assigned products to a specific factory *(f'')* has become $[\frac{N'}{f'} + 1]$, delete all $W_{j,f}$ for *f=f''*.
5. Go to step 3 if any $W_{j,f}$ is remained.

3.4. Determining lot-sizing, sequencing and scheduling decisions

Rolling-horizon heuristics have been used to overcome computational infeasibility issues for large-sized MIP problems by substituting most of the binary variables and constraints with continuous ones. The initially adopted approach decomposes the main problem into a sequence of smaller MIPs, each of which with a more tractable number of binary variables. However, for the large instances of problem, the computational intractability has been remained (Mohammadi et al., 2010; Merece and Fonton, 2003; Araujo et al., 2007; Araujo et al. 2008; Beraldi et al., 2008; Clark, 2003; Clark, 2000).

To confront the mentioned intractability, Mohammadi et al. (2010) relaxed all binary variables of the problem and solved the resulting problem through an iterative procedure. In iteration *k*, relaxed binary variables of period *k* have been divided into the two different groups where the members of the first and second groups respectively get value 1 and 0. The iterative procedure has been used in this paper is similar to that of Mohammadi et al. (2010) and has been demonstrated in Figure 1.

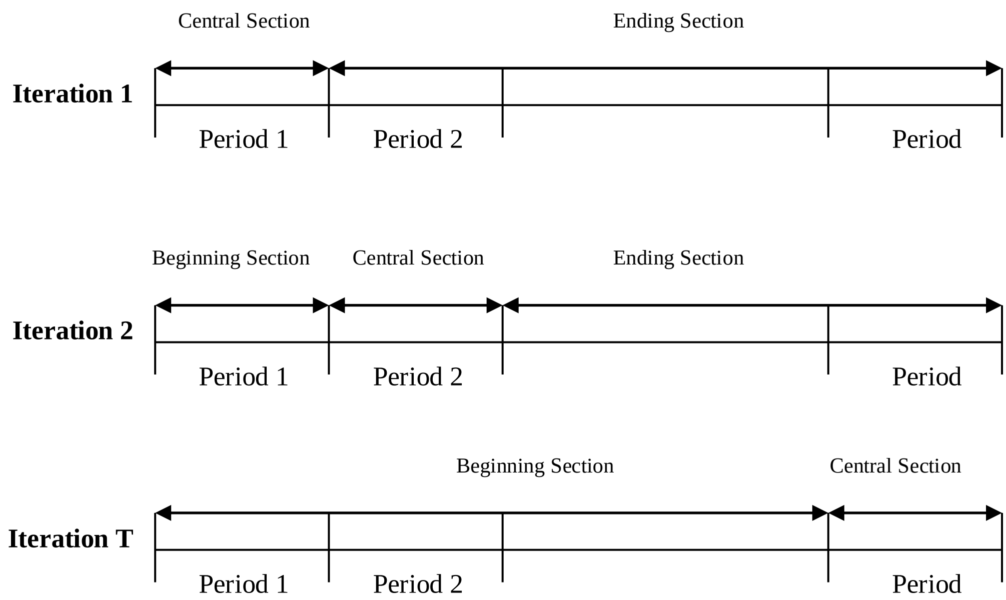

Figure 1. The iterative procedure

In a specific iteration *k*, the rolling-horizon approach includes the following sections:

1. The beginning section (first section) is composed of the *(k-1)* first periods. All variables and constraints of the beginning section are frozen.
2. The central section (second section) contains the *kth* period. The relaxed binary variables of this period are divided into the two groups where the members of first and second groups get value 1 and 0, respectively. The dividing heuristic is as follows:
   For factories *f'+1* to *F*, for *n=1* to *N*: $y^{n}_{0,0,m,f,k} = 1 \quad (n = 1,..., N; m = 1,..., M)$.

As mentioned in the former sub-section, each of the product *(1 to N')* has been assigned to a specific factory *f (f=1,…,f')*
For *f=1* to *f'*:

1. The total setup costs for the products are calculated as follows:

$$W_j = \sum_{m=1}^{M} \sum_{\substack{i=0,\, for\, i>0 \\ ,i\ is\ produced in\ f}}^{N} W_{i,j,m,f} \quad (j\, is\ produced\ in\ factory\ f)$$

2. The products are sorted in a non-decreasing order of W_j values.
3. Delete products in which $I_{j,1,k-1} > d_{j,k}$. Then, instead of each deleted product, the last remaining product is replaced. The last product is also repeated *(N-j')* times, where *j'* is the number of products which are produced in factory *f*.
4. Set $end_0 = 0$. For *k>1*, end_{k-1} is the last product in the sequence vector of factory *f* in period *k-1*.
5. Let *[i]* denote the *ith* product in the ordered sequence.
   For *[i] =1* to *N*:
   a. Consider the insertion of product *[i]* into the every position.
   b. Calculate the sum of setup costs for all products scheduled so far using the actual setup costs.
   c. Place product *i* in a position with lowest resulting sum of setup costs.
6. Let *[i]* indicate the *ith* product in the final ordered sequence. If *i=1*, then $y^{i}_{end_{k-1},[i],m,f,k} = 1, else, y^{i}_{[i-1],[i],m,f,k} = 1$.

3. The ending section (third section) consists the last periods (from period *k+1* to *T*). Therefore, the model is simplified similar to that of Mohammadi et al. (2010). Simplification strategy is described as follows:

Computational time would be saved more if the majority of variables from the ending section would be eliminated. $y^{n}_{i,j,m,f,t}(n>1), x^{n}_{j,m,f,t}(n>1)\, and\, q^{n}_{j,m,f,t}$ are eliminated from the ending section. Also, all constraints except equations (2), (3) & (5) are ignored, and all setup costs (and times) are assumed to be zero.

For ending section, $b_{j,m,f}$ and $p_{j,m,f,t}$ is modified in order to estimate the capacity consumption of future setups. Assume that A_1 and A_2 denote the objective value of the lower bound (M_2) and its respective sum of variable production costs, respectively. For ending section, $b_{j,m,f}$ and $p_{j,m,f,t}$ are replaced by $b'_{j,m,f}$ and $p'_{j,m,f,t}$ respectively as follows:

$$b'_{j,m,f} = (\frac{A_1}{A_2}).b_{j,m,f} \quad ; \quad p'_{j,m,f,t} = (\frac{A_1}{A_2}).p_{j,m,f,t}$$

The iteration *k* would be continued by solving a linear programming problem consisting of all 3 sub-sections. At the end of each iteration, the continuous variables of central section are modified as follows to reduce the related holding costs:

For *f=1* to *f'*, for *j=1* to *N*, for $x^n_{j,f,k} > 0$, a specific value of L_j is determined satisfying the following relation:

$$\sum_{l=1}^{L_j} d_{j,k+l} \leq x^n_{j,M,f,k} - d_{j,k} \leq \sum_{l=1}^{L_j+1} d_{j,k+l}$$

and for *m=1 to M*, the value of $x^n_{j,m,f,k}$ is changed to $\sum_{l=1}^{L_j} d_{j,k+l} + d_{j,k}$.

L_j indicates the last period that its respective demand of product j has been produced in period *k*. To ensure that the equations (2) & (3) hold true, $I_{j,m,k}$ would be modified as follows:

$$I_{j,M,k} = \sum_{n=1}^{N}\sum_{f=1}^{F} x^n_{j,f,M,k} + I_{j,M,k-1} - d_{j,k} \tag{21}$$

$$I_{j,m,k} = I_{j,m+1,k} \quad (m = 1,\ldots,M-1) \tag{22}$$

$$x^n_{j,m,f,k} = x^n_{j,m+1,f,k} \quad (m = 1,\ldots,M-1) \tag{23}$$

This implies that in central section, the production is either zero or equal to the some of consecutive demands for a number of periods into the future.

4. Numerical experiments

Some numerical tests have been performing to ascertain the accuracy of the lower bound. Tables 1 & 2 show the results of such tests in some instances of the problem with *(N=3, M=2, F=2, T=3)* and *(N=3, M=3, F=3, T=3).*

Table 1. Comparison of M_2 against M_1 and original model in problem size $N=3$, $M=2$, $F=2$ & $T=3$. The values inside the brackets are the computational time in seconds and the percentage values are the difference between the objective values of M_1 & M_2 against the original model.

NUMBER	ORIGINAL PROBLEM	M_1	M_2
	3157.70	2808.79	3031.23
1		11.05%	4.01%
	(243.37)	(0.054)	(1.012)
	3381.65	2929.53	3180.29
2		13.37%	5.95%
	(268.38)	(0.043)	(1.504)
	3253.91	2880.47	3098.53
3		11.48%	4.78%
	(294.14)	(0.061)	(0.941)
	3415.03	3019.30	3282.35
4		11.59%	3.89%
	(298.43)	(0.055)	(1.120)
	3277.33	2920.43	3090.21
5		10.89%	5.71%
	(266.41)	(0.072)	(0.949)

Table 2. Comparison of M_2 against M_1 and original in problem size $N=3$, $M=2$, $F=3$ & $T=3$. The values inside the brackets are the computational time in seconds and the percentage values are the difference between the objective values of M_1 & M_2 against the original model.

NUMBER	ORIGINAL PROBLEM	M_1	M_2
	4631.24	4157.93	4398.44
1		10.20%	5.03%
	(9691.51)	(0.134)	(11.57)
	5151.08	4646.22	4922.34
2		9.81%	4.44%
	(10091.96)	(0.144)	(15.11)
	5231.19	4621.09	4996.91
3		11.62%	4.48%
	(9091.23)	(0.191)	(9.57)
	4997.43	4500.12	4717.59
4		10.00%	5.60%
	(11233.44)	(0.119)	(12.19)
	5233.19	4668.17	4953.39
5		10.80%	5.35%
	(9903.42)	(0.149)	(16.69)

Comparison the results of the second columns of Tables 1 and 2 shows that computation time grows exponentially by increasing the dimension of the problem. According to Table 1, the average computational time for problems with (N=3, M=2, F=2, T=3) is 274.15s. According to Table 2, the average computational time for problems with (N=3, M=3, F=3, T=3) is 10002.31s. It means that by increasing M&F from 2 to 3, average computational time increases more than 36 times. Tables 1 and 2 also confirm the advantages of lower bound M_2.

To apply the exact model, M_1 and M_2, GAMS models are provided using GAMS IDE (ver 2.0.19.0) and solved using OSL 2. The heuristic is coded in Matlab programming language. All models are run on a personal computer with a Pentium 4 processor running at 3.4 GHZ. The required parameters for all numerical experiments are extracted from the following uniform distributions:

$$b_{j,m,f} \approx U(1.5,2),\ d_{j,t} \approx U(0,180),\ h_{j,M} \approx U(0.2,0.4),\ P_{j,m,f,t} \approx U(1.5,2),$$
$$W_{i,j,m,f} \approx U(35,70),\ S_{i,j,m,f} \approx U(35,70),\ C_{m,f,t} \approx U(a,b),\quad a = 200.N + 100.(m-1)$$
$$b = 200.N + 200.(m-1),\ A_f \approx U(150,250),\ O_{j,t} \approx U(3,4)$$

To evaluate and compare the performance of developed heuristic, 20 problems with different sizes are selected to test. For each problem size, five problem instances are randomly generated and the required parameters for those problems are extracted from the mentioned uniform distributions. Table 3 compares the solution times and objective values of heuristic and lower bound.

Table 3. Comparison of lower bound and heuristic. The values inside the brackets are the computational time in seconds.

PROBLEM SIZE (*N.M.F.T*)	LOWER BOUND (M_2)	HEURISTIC
3.3.3.3	(13.03)	(0.23)
		14.94%
5.3.5.3	(98.38)	(1.57)
		16.03%
3.5.3.3	(17.77)	(0.59)
		16.57%
3.3.3.5	(15.43)	(1.15)
		17.02%
5.5.5.5	(300.71)	(10.52)
		15.19%
7.5.7.5	(1419.09)	(115.54)
		15.93%
5.7.5.5	(730.43)	(81.27)

		17.05%
5.5.5.7	(593.03)	(19.37) 16.55%
7.7.7.7	>7200*	(514.36) 16.98%
10.5.10.5	(7047.33)	(519.84) 16.23%
5.10.5.5	(4353.23)	(87.09) 16.09%
5.5.5.10	(1113.95)	(59.39) 17.36%
10.7.10.7	>7200*	(2057.39) 16.49%
7.10.7.7	>7200*	(215.69) 17.41%
7.7.7.10	>7200*	(300.49) 17.79%
10.10.10.10	>7200*	(3011.15) 17.09%
15.10.15.10	>7200*	(5043.88) 17.55%
10.15.10.10	>7200*	(2909.09) 18.14%
10.10.10.15	>7200*	(3922.53) 18.69%
15.15.15.15	>7200*	(7190.51) 19.61%

*Means that finding the optimum value for the second lower bound requires more than 7200 seconds and the objective value at this time has been considered. The percentage values are the difference between the objective values of the heuristic against the lower bound.

5. Conclusion and future studies

The main contribution of this paper is to develop one exact formulation and a heuristic for a complex flow shop lot-sizing and scheduling problem. Considering the expanding role of meta-heuristic approaches to solve complicated lot-sizing problems (Jans and Degraeve, 2007; Defersha and Chen, 2008), the application of meta-heuristic approaches to face this complex problem could be mentioned as a future research.

REFERENCES

[1] **Wagner H.M., Within T.M. (1958),** ***Dynamic Version of the Economic Lot Size Model****;* Management Science. 5, pp. 89-96;
[2] **Mohammadi M., Torabi S.A., Fatemi Ghomi S.M.T., Karimi B. (2010),** ***A New Algorithmic Approach for Capacitated Lot-Sizing Problem in Flow Shops with Sequence-Dependent Setups*****;** International Journal Advantages Manufacturing Technology 49, pp. 201-211;
[3] **Fandle G., Stammen-Hegene C. (2006),** ***Simultaneous Lot-sizing and Scheduling for Multi-product Multi-level Production***; International Journal Production Economy 104 (2), pp. 308-316;
[4] **Karimi B., Fatemi Ghomi S.M.T., Wilson J.M. (2003),** ***The Capacitated Lot Sizing Problem: A Review of Models and Algorithms****;* Omega 31(5), pp. 365-378;
[5] **Merece C., Fonton G. (2003),** ***MIP-based Heuristics for Capacitated Lot-sizing Problems;*** International Journal Production Economy 85(1), pp. 97-111;
[6] **Zhu X., Wilelm W. E. (2006),** ***Scheduling and Lot-sizing with Sequence-dependent Setup: A Literature Review***; IIE Trans 38, pp. 987-1007;
[7] **Mohammadi M., Fatemi Ghomi S.M.T., Karimi B., Torabi S.A. (2010),** ***MIP-based Heuristics for Lot-sizing in Capacitated Pure Flow Shop with Sequence-dependent Setups***; International Journal Production Research 48(10), pp. 2957-2973;
[8] **Mohammadi M., Fatemi Ghomi S.M.T., Karimi B., Torabi S.A. (2010),** ***Rolling-horizon and Fix-and-relax Heuristics for the Multi-product Multi-level Capacitated Lot-sizing Problem with Sequence-dependent Setups****;* International Journal Manufacturing 21, pp. 501-510;
[9] **Mohammadi M., Fatemi Ghomi S.M.T. (2011),** ***Genetic Algorithm-based Heuristic for Capacitated Lot-sizing Problem in Flow Shops with Sequence-dependent Setups****;* Expert System Appl 38, pp. 7201-7207;
[10] **Naderi B., Ruiz R. (2010),** ***The Distributed Permutation Flow Shop Scheduling Problem****;* Computer Operation Research 37(4), pp. 754-768;
[11] **Moon C., Kim J., Hur S. (2002),** ***Integrated Process Planning and Scheduling with Minimizing Total Tardiness in Multi-plants Supply Chain***; Computer & Industrial Engineering 43 (1-2), pp. 331-349;
[12] **Almada-Lobo B., Klabjan D., Carravilla M.A., Oliviera J. (2007),** ***Single Machine Multi-product Capacitated Lot-sizing with Sequence-Dependent Setups****;* International Journal Production Research 45(20), pp. 4873-4894;
[13] **Araujo S.A., Arenales M.N., Clark A.R. (2007),** ***Joint Rolling-horizon Scheduling of Materials Processing and Lot-sizing with Sequence-dependent Setups****;* Journal Heuristics 13(4), pp. 337-358;
[14] **Araujo S.A., Arenales M.N., Clark A.R. (2008),** ***Lot-sizing and Furnace Scheduling in Small Foundries***; Computer Operation Research 35(3), pp. 916-932;

[15] **Beraldi P., Ghiani G., Grieco A., Guerriero E. (2008),** ***Rolling-horizon and Fix-and-relax Heuristics for the Parallel Machine Lot-sizing and Scheduling Problem with Sequence-dependent Setup Costs***; Computer Operation Research 35(11), pp. 3644-3656;
[16] **Clark A.R. (2003),** ***Optimization Approximations for Capacity Constrained Material Requirements Planning***; International Journal Production Economy 84(2), pp. 115-131;
[17] **Clark A.R., Clark S.J. (2000),** ***Rolling-horizon Lot-sizing when Setup Times are Sequence-dependent;*** International Journal Production Research 38(10), pp. 2287-2308;
[18] **Chung. D. and Choi. B. (2013),** ***Outsourcing and Scheduling for two-Machine Ordered Flow Shop Scheduling Problems***; European Journal of Operational Research 226, pp. 46–52;
[19] **Sukkerd. W., Wuttipornpun. T. (2016),** ***Hybrid Genetic Algorithm and Tabu Search for Finite Capacity Material Requirement Planning System in Flexible Flow Shop with Assembly Operations***; Computers & Industrial Engineering 97, pp. 157–169;
[20] **Zandieh M, Rashidi. E (2009),** ***An Effective Hybrid Genetic Algorithm for Hybrid Flow Shops with Sequence Dependent Setup Times and Processor Blocking;*** Journal of Industrial Engineering 4, pp. 51- 58;
[21] **Azab. A., Naderi. B. (2014),** ***Greedy Heuristics for Distributed Job Shop Problems;*** 2nd ICRM 2014 International Conference on Ramp-Up Management, DOI:10.1016/j.procir.2014.05.025;
[22] **Costa. A., Cappadonna. F. A., Fichera. S. (2015),** ***A Hybrid Genetic Algorithm for Minimizing Makespan in a Flow-Shop Sequence-dependent Group Scheduling Problem;*** Journal of Intelligent Manufacturing, pp. 1–15, DOI:10.1007/s10845-015-1049-1;
[23] **Jans R., Degraeve Z. (2007),** ***Meta-heuristics for Dynamic Lot-sizing: A Review and Comparison of Solution Approaches***; European Journal Operation Research 177(3), pp. 1851-1875;
[24] **Defersha F.M., Chen M. (2008),** ***A Linear Programming Embedded Genetic Algorithm for an Integrated Cell Formation and Lot-Sizing Considering Product Quality;*** European Journal Operation Research 187(1), pp. 46-69.