Transaction E: Industrial Engineering
Vol. 17, No. 2, pp. 85–94

SCIENTIA IRANICA

A New Rolling-Horizon Technique for Lotsizing in a Capacitated Pure Flow Shop with Sequence-Dependent Setups

M. Mohammadi[1] and S.M.T. Fatemi Ghomi[2,*]

Abstract. *A new rolling-horizon approach is presented in this paper to solve the problem of lotsizing in a capacitated pure flow shop with sequence dependent setups. Two solution algorithms are provided, based on a simplified version of the problem, combining the rolling-horizon approach with a heuristic. To evaluate the effectiveness of the proposed algorithms, a comparison is made between the results obtained by the proposed algorithms and those obtained by existing algorithms. The comparison indicates the superiority of the proposed algorithm for large scale problems.*

Keywords: *Lotsizing; Scheduling; Pure flow shop; Sequence-dependent; Rolling-horizon; Multi-level.*

INTRODUCTION

Modeling and solving lotsizing problems have been an area of active research starting from the seminal paper of Wagner and Whithin [1]. Since then, there has been a considerable amount of investigation 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 accuracy of the model parameters, the number of products and production stages, the production structure and the capacity restrictions [2-5].

Models of lotsizing and scheduling are divided in the literature into small bucket and big bucket problems [6]. Readers interested in small bucket problems can refer to [7]. Small bucket problems for multi-level multi-product production include the Multi-Level Discrete Lotsizing and Scheduling Problem (MLDLSP) [8] and the Multi-Level Proportional Lotsizing and Scheduling Problem (MLPLSP) [9-10]. 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) is a big bucket problem, and does not have this disadvantage, but it cannot determine lotsizes and schedules simultaneously.

To attempt to unite the advantages of the MLPLSP and MLCLSP, Fandel and Stammen-Hegene [3] presented a model formulation based on a two-level time structure [11], which enables simultaneous lotsizing and scheduling for multi-product multi-level job shop production. The non-linear nature of the model and existing three types of binary variable make the model too complicated, therefore, no solution approach has yet been presented.

Recently, Mohammadi et al. [12,13] proposed a mathematical formulation for lotsizing in capacitated pure flow shops with sequence-dependent setups and developed novel heuristics, all based on solving a sequence of smaller Mixed Integer Programs (MIPs). To solve larger instances of the problem, Mohammadi et al. [14] proposed an algorithmic approach. Heuristics provided in Mohammadi et al. [12] are superior to those provided in Mohammadi et al. [13,14]. The current paper proposes more efficient heuristics to solve the problem.

This paper has the following structure. The next section introduces a detailed description of the problem

1. *Department of Industrial Engineering, Faculty of Engineering, Tarbiat Moallem University, Karaj, Iran.*
2. *Department of Industrial Engineering, Amirkabir University of Technology, Tehran, P.O. Box 15916-34311, Iran.*
*. *Corresponding author. E-mail: fatemi@aut.ac.ir*

Received 27 August 2009; received in revised form 7 September 2010; accepted 6 November 2010

and its underlying assumptions. Further sections provide the solution approaches and report the numerical experiments, respectively. Finally, the last section is devoted to concluding remarks and recommendations for future studies.

PROBLEM DEFINITION AND MATHEMATICAL MODEL

Assumptions

Our assumptions and mathematical model are similar to those of Mohammadi et al. [12-14] and are presented as follows:

- Several products are produced on serially-arranged machines in a pure flow shop structure.
- Each machine is constrained in capacity.
- When the machines are setup, sequence-dependent setup costs and times are incurred.
- The setting-up of a machine must be completed within a period.
- There must be precise N (number of products) setups at 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 [15-16]. This feature makes it possible for a lotsize, or production run, to continue over consecutive time periods without incurring setup costs or times in the later periods (setup carry over).
- The required resources and parts must be ready for production.
- External demand exists for final products and must be satisfied at the end of each period.
- There are no lead times between the different production levels.
- Shortages are not permitted.
- A component cannot be produced earlier in a period than the finish of the production of its required component. In other words, production on a production level can only be started 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 between each setup and its production is defined with the help of a shadow product [3].
- There are no demands and no storage costs for shadow products.
- At the beginning of the planning horizon, all machines are setup for a defined product.
- 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.

MATEMATICAL MODEL

Indices

i, j, k	product type
n, n'	designation for a specific setup number
m	level of production
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
$C_{m,t}$	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
$b_{j,m}$	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)
$S_{i,j,m}$	sequence-dependent setup time for the changeover of machine m from production of product i to production of product j (in time units), for $i = j$, $S_{i,j,m} = 0$
$W_{i,j,m}$	is the sequence-dependent setup cost for the changeover of machine m from production of product i to production of product j (in money units), for $i = j$, $W_{i,j,m} = 0$, setup costs have the form $W_{i,j,m} = f_w.S_{i,j,m}$ where f_w is opportunity cost per unit of setup time.
j_0	the starting setup configuration on machines. This is the same on all machines.

Decision Variables

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

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

$x^n_{j,m,t}$ quantity of product j produced after nth setup on machine m in period t

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

Readers interested in the original model can refer to [12]. As mentioned briefly in the introduction, to solve larger instances of the problem, Mohammadi et al. [12] provided three (of four) heuristics, based on a simplified model, which assumed that products follow a similar sequence on all machines. Therefore, they used $y^n_{i,j,t}$ instead of $y^n_{i,j,m,t}$. In this paper, to solve the problem through our algorithmic approach, we focus on a more simplified model. In this model, in addition to the sequence of products, the size of products is also independent of the machines. Therefore, $y^n_{i,j,t}$, $x^n_{j,t}$ and $I_{j,t}$ are used instead of $y^n_{i,j,m,t}$, $x^n_{j,m,t}$ and $I_{j,m,t}$ in this model. Other parameters and variables are similar to those mentioned in the former model. Our proposed simplified model follows:

$$\min \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} w_{i,j,m}.y^n_{i,j,t} + \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} p_{j,m,t}.x^n_{j,t} + \sum_{j=1}^{N}\sum_{t=1}^{T} h_{j,M}.I_{j,t}. \tag{1}$$

Subject to:

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

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

$$\sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N} y^n_{i,j,t}.S_{i,j,m} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{j,m}.q^n_{j,m,t} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{j,m}.x^n_{j,t} \le \sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N} y^n_{i,j,t}.S_{i,j,m+1} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{j,m+1}.q^n_{j,m+1,t} + \sum_{n=1}^{n'-1}\sum_{j=1}^{N} b_{j,m+1}.x^n_{j,t},$$

$$n' = 1,\cdots,N,$$

$$m = 1,\cdots,M-1,$$

$$t = 1,\cdots,T, \tag{3}$$

$$\sum_{n=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y^n_{i,j,t}.S_{i,j,m} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{j,m}.x^n_{j,t} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{j,m}.q^n_{j,m,t} \le C_{m,t},$$

$$m = 1,\cdots,M,$$

$$t = 1,\cdots,T, \tag{4}$$

$$x^n_{j,t} \le (\text{big}M). \sum_{i=1, i\neq j (\text{forn}>1)}^{N} y^n_{i,j,t},$$

$$n = 1,\cdots,N,$$

$$j = 1,\cdots,N,$$

$$t = 1,\cdots,T, \tag{5}$$

$$q^n_{j,m,t} \le (C_{m,t}/b_{j,m}). \sum_{i=1}^{N} y^n_{i,j,t},$$

$$n = 1,\cdots,N,$$

$$j = 1,\cdots,N,$$

$$m = 1,\cdots,M,$$

$$t = 1,\cdots,T, \tag{6}$$

$$y^1_{j,i,1} = 0,$$

$$j \neq j_0,$$

$$i = 1,\cdots,N, \tag{7}$$

$$\sum_{i=1}^{N} y^{1}_{j_0,i,1} = 1, \tag{8}$$

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

$$n = 1, \cdots, N-1,$$

$$i = 1, \cdots, N,$$

$$t = 1, \cdots, T, \tag{9}$$

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

$$i = 1, \cdots, N,$$

$$t = 2, \cdots, T, \tag{10}$$

$$y^{n}_{i,j,t} = 0 \quad \text{or} \quad 1, \tag{11}$$

$$I_{j,t}, x^{n}_{j,t}, q^{n}_{j,m,t} \geq 0, \tag{12}$$

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

$$j = 1, \cdots, N. \tag{13}$$

In this model, Equation 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 guarantees that within one period, each typical product j on machine m is produced before its direct successor (product j on machine $m+1$).

The left side of Equation 3 is equal to the time between the beginning of period t and the end of production of product j on machine m. The right side of Equation 3 is equal to the time between the beginning of period t and the beginning of production of product j on machine $m+1$.

Equation 4 represents the capacity constraints of machines during periods. Equation 5 indicates that setup is considered in the production process. Equation 6 indicates the relationship between shadow products and setups. Equations 7 and 8 guarantee that the first setup at the beginning of the planning horizon is from a defined product. Equations 9 and 10 represent the relationship between successive setups. Equations 11 and 12 represent the type of variables. Equation 13 indicates that at the beginning of the planning horizon there is no on-hand inventory.

SOLUTION METHOD

Idea

Rolling-horizon heuristics are usually used in dynamic lotsizing and scheduling problems, where demands are gradually revealed during the planning horizon and part types have to be allocated to machines in an ongoing fashion as new orders arrive. On the other hand, a rolling-horizon approach is still suitable when all parameters are perfectly known [4,15-19]. In this case, 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. The approach initially adopted decomposes the model into a succession of smaller MIPs, each with a more tractable number of binary variables. For larger instances of problem, the resulting huge number of binary variables in each MIP causes great computational intractability.

To face this intractability, instead of solving a succession of smaller MIPs, we would relax all binary variables of the problem. The resulting problem is solved through a T-iteration based algorithm. In a specific iteration, k, the relaxed binary variables of period k are divided into two groups, where members of the first group have value 1 and members of the second group have value 0. In other words, our rolling-horizon approach decomposes the planning horizon into three sections as follows. For a given iteration, k, all sections are described as follows:

- The first section (beginning section) is composed of the first $(k-1)$ periods. Within this section, the decision variables have been partially or completely frozen by the previous iterations, according to a selected freezing strategy.
- The second section (central section) is the kth period. The relaxed binary variables of this period are divided into two groups, where members of the first group have value 1 and members of the second group have value 0.
- The third section (ending section) is composed of the last periods (from period $k+1$ to period T). There, the model is simplified according to a selected simplification strategy.

Each iteration is completed by solving a linear programming problem that consists of all three sections. At the end of iteration k, all sections roll forward by one period and a new iteration is then performed. The procedure stops when there is no longer an ending section. The last iteration (T) defines all decision variables over the entire horizon. Figure 1 demonstrates the iterative procedure.

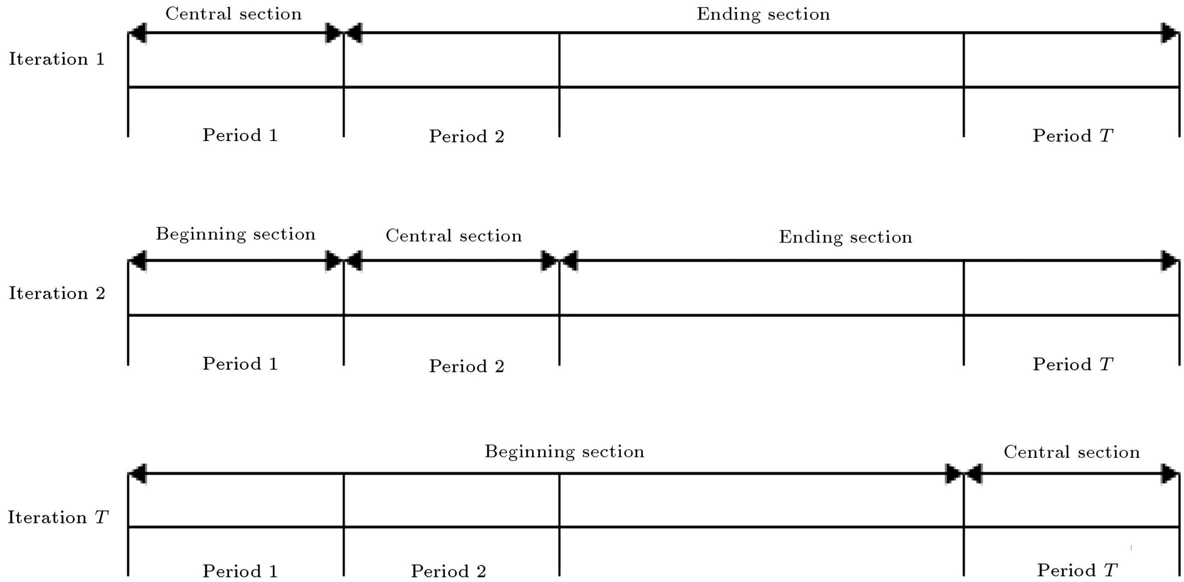

Figure 1. Demonstration of the iterative procedure.

Procedure to Divide Relaxed Binary Variables into Two Groups

As mentioned before, in each iteration of the rolling-horizon approach, all relaxed binary variables of the central section are divided into two groups. Members of the first group would get value 1 and members of the second group would get value 0. Note that according to Equations 7 to 10, for each pair (n,t), there is exactly one pair (i,j) for which $y^n_{i,j,t}=1$ and for $(i',j')\neq(i,j)$, $y^n_{i',j',t}=0$. In other words, for each pair (n,t) of the central section, a pair (i,j), in which $y^n_{i,j,t}=1$ would be specified through a heuristic, which will be explained in the next subsection.

Description

A detailed description of the rolling-horizon approach is now given.

The First Rolling-Horizon Approach (R1)

For a given iteration, k, this rolling-horizon approach contains the following sections:

- The first section (beginning section) is composed of the $(k-1)$ first periods. Within this section, $x^n_{j,t}$ and $y^n_{i,j,t}$ have been frozen by the previous iterations.
- The second section (central section) includes the kth period. The relaxed binary variables of this period would be divided into two groups, where members of the first group would get value 1 and members of the second group would get value 0.

This procedure would be described as follows.

In this heuristic, relaxed binary variables of the central section would get value 0 or 1 by an adaptation of the Flow Time Multiple Insertion Heuristic (FTMIH). This class of heuristics is based on the insertion heuristic for the Traveling Salesman Problem (TSP), and can be considered an adaptation of a heuristic by Nawaz et al. [20]. Kurz and Askin [21-22] used a modified version of this heuristic in scheduling flexible flow lines with sequence dependent setups. The insertion heuristic for the TSP begins with a set of cities to be visited in order, and a set of cities that have not yet been visited. The unvisited cities are sorted according to some criteria, and then added to the current set of cities to be visited. The location of the new city in the sequence is determined by comparing the increase in time to visit all cities.

Let $[i]$ indicate the ith product in an ordered sequence in the following heuristic. For $m=1,\cdots,M(\forall i,j)$; $S'_m=\min\{S_{i,j,m}\}$. The following steps are performed in the central section of the current iteration (k):

1. Create the product durations for all products as follows:

$$D(j,k)=d_{j,k}.\sum_{m=1}^{M}(b_{j,m}+S'_{i,j,m}).$$

2. $\text{end}_0=j_0$, where for $k>1$, end_{k-1} is the last sequenced product in period k.
3. Arrange the products in non-increasing order (LPT) of $D(j,k)$ for use in step 5.
4. Delete products for which $I_{j,k-1}>d_{j,k}$. According to Equation 13, for $k=1$, $I_{j,k-1}=0$ and $k>1$, $I_{j,k-1}$ has been determined at the end of iteration

$k-1$. Then, instead of each deleted product, the last remaining product is replaced.

5. For $[i]=1$ to N:
   (a) Consider inserting product $[i]$ into every position.
   (b) Calculate the sum of flow times for all products scheduled so far, using the actual setup times.
   (c) Place product i in the position with the lowest resultant sum of flow times.
6. Let $[i]$ indicate the ith product in the ordered sequence:

$$\text{If} \quad i=1, \quad y^{i}_{\text{end}_{k-1}[i],k}=1,$$

$$\text{else} \quad y^{i}_{[i-1],[i],k}=1.$$

As mentioned before, the rest of the relaxed binary variables in the central section take value 0.

- 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, as follows.

More computational time is economized by eliminating the majority of the variables from the ending section. $y^n_{i,j,t}(n>1)$, $x^n_{jt}(n>1)$ and q^n_{jmt} are eliminated from the ending section. Except for Equations 2 and 4, the other constraints are ignored in the ending section. All setup costs (and times) for the ending section are assumed to be 0.

For the ending section, $b_{j,m}$ and $p_{j,m,t}$ should be modified to estimate the capacity consumption of future setups. We assume that A_1 is the objective value of the lower bound provided by Mohammadi et al. [12], and that A_2 is the sum of variable production costs of the aforementioned lower bound. For the ending section, we would replace $b_{j,m}$ and $p_{j,m,t}$ with $b'_{j,m}$ and $p'_{j,m,t}$ as follows:

$$b'_{j,m}=(A_1/A_2).b_{j,m},$$

$$p'_{j,m,t}=(A_1/A_2).p_{j,m,t}.$$

A simplified representation for the ending section in the rolling-horizon is less difficult to solve, and hence permits the solution of larger problems.

This iteration is completed by solving a linear programming problem that contains all three sections.

The Second Rolling-Horizon Approach (R2)

In order to solve larger instances of the problem on the one hand and improve the quality of solutions on the other, some modifications have been made to the former rolling-horizon approach, as follows:

1. In each iteration, all variables and constraints of the beginning section are frozen (instead of freezing only $y^n_{i,j,t}$ and $x^n_{j,t}$), thus making it possible to solve larger instances of the problem.
2. At the end of each iteration (after solving the linear programming problem), the continuous variables of the central section are modified to reduce the holding costs of the central section.

   Variables of the central section are modified as follows:

$$\text{For} \quad j=1 \quad \text{to} \quad N,$$

$$\text{For} \quad x^n_{j,k}>0.$$

The specific value of L_j, which satisfies the following relation, is determined.

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

The value of $x^n_{j,k}$ is changed to $\sum_{l=1}^{L_j} d_{j,k+l}+d_{j,k}$.

L_j indicates the last period in which its respective demand of product j has been produced in period k. To ensure that Equation 4 holds true, $I_{j,k}$ would be modified as follows:

$$I_{j,k}=\sum_{n=1}^{N} x^n_{j,k}+I_{j,k-1}-d_{j,k}. \tag{14}$$

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

NUMERICAL EXPERIMENTS

For a first evaluation of the proposed algorithms, we compare their results for a set of twenty problems, as introduced in [12]. Twenty different problem sizes in the range of $(N,M,T)=(3,3,3)$ to $(N,M,T)=(15,15,15)$ have been considered. For each problem size, 5 problem instances are randomly generated and the average values are considered. The required parameters are similar to those of Mohammadi et al. [12], and are as follows:

$$b_{j,m}\approx U(1.5,2),$$

$$d_{j,t}\approx U(0,180),$$

$$h_{j,m}\approx U(0.2,0.4),$$

$$p_{j,m,t}\approx U(1.5,2),$$

$W_{ijm} = S_{ijm} \approx U(35, 70)$,

$C_{m,t} = U(a, b)$,

$a = 200N + 100(m - 1)$,

$b = 200N + 200(m - 1)$.

Our proposed algorithms are coded in the Matlab programming language and are run on a personal computer with a Pentium 4 processor running at 3.4 GHZ. Table 1 shows the comparison between the objective functions and CPU times of the proposed algorithms. The lower bound provided by Mohammadi et al. [12] is used to compare our proposed algorithms against the heuristics provided in [12]. Because heuristics provided in [12] are superior to those provided in [13-14], they are used for comparison with our proposed algorithms. Tables 2 and 3 report such comparisons for small and non-small instances, respectively.

Table 2 shows that the average difference between the heuristics provided by Mohammadi et al. [12] (H1 to H4) and our proposed algorithms (R1 and R2) against the mentioned lower bound for small instances are, respectively, 11.74%, 13.44%, 22.83%, 18.57%, 23.61% and 16.85%. This shows the superiority of the proposed heuristics for small instances of the problem.

Table 3 shows that the average difference between the heuristics provided by Mohammadi et al. [12] (H3 to H4) and our proposed algorithms (R1 and R2) against the aforementioned lower bound for large instances are, respectively, 26.90%, 22.59%, 22.95% and 16.92%. This shows the superiority of the proposed heuristics for large instances of the problem. Table 3 shows the advantages of R2 for large instances of the problem.

CONCLUDING REMARKS AND RECOMMENDATION FOR FUTURE STUDIES

The main contribution of this paper is the presentation of a new rolling-horizon technique to solve the problem of lotsizing in a pure flow shop with sequence-dependent setups. Two heuristics, based on the mentioned rolling-horizon approach, have been proposed.

According to the numerical experiments, our second proposed algorithm (R2) is superior, especially for large problem sizes.

Because of the expanding role of metaheuristic approaches to solve complicated lotsizing problems [23-25], the application of metaheuristic approaches to solve hard problems is recommended as an area for future research.

Table 1. Comparison between the objective function and computational time of the proposed algorithms. The values inside the parentheses are the computational time in seconds. "-" means that a feasible solution has not been found.

Problem Size (N.M.T)	R1	R2
3.3.3	(0.08)	(0.05)
	5,749.91	5,432.92
5.3.3	(0.91)	(0.56)
	9,661.23	9,062.10
3.5.3	(0.25)	(0.17)
	9,604.19	9,033.63
3.3.5	(0.44)	(0.28)
	9,710.98	9,101.23
5.5.5	(5.64)	(2.31)
	27,351.54	25,852.76
7.5.5	(51.85)	(12.64)
	36,215.41	34,001.05
5.7.5	(13.45)	(4.63)
	34,436.45	32,940.46
5.5.7	(8.39)	(2.96)
	36,277.19	34,205.39
7.7.7	(376.11)	(87.02)
	71,240.44	68,082.26
5.10.5	(23.97)	(4.59)
	50,862.92	47,632.13
5.5.10	(81.48)	(5.75)
	52,379.12	50,150.03
7.10.7	(503.34)	(33.24)
	109,220.27	103,668.31
7.7.10	(1,543.10)	(41.98)
	110,479.94	104,516.72
10.5.5	(814.88)	(135.23)
	52,149.02	49,353.29
10.7.7	(5,888.78)	(539.81)
	110,759.81	104,679.93
10.10.10	-	(1,222.21)
	—	210,803.47
15.10.10	—	(2,858.53)
	—	337,524.75
10.15.10	—	(1,396.16)
	—	321,241.12
10.10.15	—	(1,937.09)
	—	317,337.40
15.15.15	—	(4,997.12)
	—	779,964.12

Table 2. Comparison between our proposed algorithms (R1 and R2) and those provided by Mohammadi et al. [12] (H1 to H4) for small and medium instances. The values inside the parentheses are the computational time in seconds. The percentage values are the difference between the objective values against lower bound provided by Mohammadi et al. [12]. "-" means that a feasible solution has not been found.

Problem Size (N.M.T)	H1	H2	H3	H4	R1	R2
3.3.3	(2.31)	(0.036)	(0.13)	(0.70)	(0.08)	(0.05)
	10.02%	12.77%	18.11%	13.89%	22.92%	16.14%
5.3.3	(711.11)	(9.44)	(0.44)	(2.98)	(0.91)	(0.56)
	9.60%	12.98%	18.90%	15.49%	23.92%	16.24%
3.5.3	(98.34)	(1.38)	(0.16)	(1.42)	(0.25)	(0.17)
	10.40%	14.25%	19.90%	14.56%	24.32%	16.93%
3.3.5	(26.26)	(0.21)	(0.23)	(0.91)	(0.44)	(0.28)
	10.60%	13.26%	19.59%	16.85%	22.88%	15.17%
5.5.5	(7,200)	(146.67)	(2.19)	(24.51)	(5.64)	(2.31)
	11.18%	14.74%	20.33%	16.17%	24.57%	17.74%
7.5.5	—	(2,147.43)	(15.74)	(208.22)	(51.85)	(12.64)
	—	12.82%	23.88%	20.29%	23.52%	15.97%
5.7.5	—	(735.58)	(5.21)	(58.31)	(13.45)	(4.63)
	—	14.24%	25.23%	19.48%	22.26%	16.94%
5.5.7	(7,200)	(318.33)	(5.28)	(38.81)	(8.39)	(2.96)
	18.66%	13.19%	25.08%	22.45%	23.51%	16.46%
7.7.7	—	(4,038.80)	(77.61)	(1,437.27)	(376.11)	(87.02)
	—	12.97%	25.48%	19.35%	22.90%	17.54%
5.10.5	—	(1,846.93)	(11.43)	(127.51)	(23.97)	(4.59)
	—	12.78%	24.38%	21.60%	23.98%	16.10%
5.5.10	—	(734.66)	(8.52)	(96.31)	(81.48)	(5.75)
	—	14.34%	26.48%	21.55%	23.26%	18.02%
7.10.7	—	(7,200)	(58.72)	(1,411.13)	(503.34)	(33.24)
	—	13.76%	24.26%	19.42%	23.67%	17.38%
7.7.10	—	(7,200)	(79.60)	(1,983.31)	(1,543.10)	(41.98)
	—	12.62%	25.22%	20.34%	25.25%	18.48%

Table 3. Comparison between our proposed algorithms (R1 and R2) and those provided by Mohammadi et al. [12] (H1 to H4) for large instances. The values inside the parentheses are the computational time in seconds. The percentage values are the difference between the objective values against lower bound provided by Mohammadi et al. [12]. "-" means that a feasible solution has not been found.

Problem Size (N.M.T)	**H1**	**H2**	**H3**	**H4**	**R1**	**R2**
10.5.5	—	—	(132.55)	(2,095.89)	(814.88)	(135.23)
	—	—	22.94%	19.14%	23.24%	16.63
10.7.7	—	—	(298.96)	(3,807.97)	(5,888.78)	(539.81)
	—	—	25.43%	22.48%	22.66%	15.93%
10.10.10	—	—	(823.33)	(5,711.96)	—	(1,222.21)
	—	—	26.98%	22.11%	—	16.38%
15.10.10	—	—	(1,831.27)	—	—	(2,858.53)
	—	—	26.66%	—	—	18.07%
10.15.10	—	—	(1,138.51)	(6,854.35)	—	(1,396.16)
	—	—	29.38%	26.65%	—	16.21%
10.10.15	—	—	(1,383.41)	—	—	(1,937.09)
	—	—	31.60%	—	—	17.82%
15.15.15	—	—	(3,231.59)	—	—	(4,997.12)
	—	—	25.29%	—	—	17.38%

REFERENCES

1. Wagner, H.M. and Whithin, T.M. "Dynamic version of the economic lot size model", *Management Sci.*, **5**, pp. 89-96 (1958).
2. Kianfar, F. and Mokhtari, G. "Lot sizing and lead time quotations in assembly systems", *Scientia Iranica, Transaction E, Industrial Eng.*, **16**(2), pp. 100-113 (2009).
3. Fandel, G. and Stammen-Hegene, C. "Simultaneous lot sizing and scheduling for multi-product multi-level production", *Int. J. Prod. Eco.*, **104**(2), pp. 308-316 (2006).
4. Merece, C. and Fonton, G. "MIP-based heuristics for capacitated lotsizing problems", *Int. J. Prod. Eco.*, **85**(1), pp. 97-111 (2003).
5. Karimi, B., Fatemi Ghomi, S.M.T. and Wilson, J.M. "The capacitated lot sizing problem: a review of models and algorithms", *Omega*, **31**(5), pp. 365-378 (2003).
6. Eppen, G.D. and Martin, R.K. "Solving multi-item capacitated lot-sizing problems using variable redefinition", *Operat. Res.*, **35**(6), pp. 832-848 (1987).
7. Tempelmeier, H. and Buschkuhl, L. "Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource", *Int. J. Prod. Eco.*, **13**(1), pp. 401-412 (2008).
8. Kimms, A. "Multi-level, single-machine lotsizing and scheduling (with initial inventory)", *Eur. J. Oper. Res.*, **89**(1), pp. 86-99 (1996).
9. Kimms, A. "Multi-level, single-machine lot sizing and scheduling (with initial inventory)", Manuskripte aus den Instituten für Betriebswirtschaftslehre der universität Kiel, No. 329, Kiel (1993).
10. Kimms, A. and Drexl, A. "Some insights into proportional lotsizing and scheduling", Manuskripte aus den Instituten der universität Kiel, No. 406, Kiel (1996).
11. Fleischmann, B. and Meyr, H. "The general lotsizing and scheduling problem", *OR Spekt*, **19**(2), pp. 11-21 (1997).
12. Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B. and Torabi, S.A. "Rolling-horizon and fix-and-relax

heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups", *J. Intell. Manuf.*, **21**(4), pp. 501-510 (2010a).

13. Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B. and Torabi, S.A. "MIP-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups", *International Journal of Production Research*, **48**(10), pp. 2957-2973 (2010b).

14. Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B. and Torabi, S.A. "A new algorithmic approach for capacitated lot-sizing problem in flow shops with sequence-dependent setups", *International Journal of Advanced Manufactury Technology*, **49**, pp. 201-211 (2010c).

15. Clark, A.R. and Clark, S.J. "Rolling-horizon lot-sizing when setup times are sequence-dependent", *Int. J. Prod. Res.*, **38**(10), pp. 2287-2308 (2000).

16. Clark, A.R. "Optimization approximations for capacity constrained material requirements planning", *Int. J. Prod. Eco.*, **84**(2), pp. 115-131(2003).

17. Beraldi, P., Ghiani, G., Grieco, A. and Guerriero, E. "Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent setup costs", *Comput. Operat. Res.*, **35**(11), pp. 3644-3656 (2008).

18. Araujo, S.A., Arenales, M.N. and Clark, A.R. "Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups", *J. Heuristics.*, **13**(4), pp. 337-358 (2007).

19. Araujo, S.A., Arenales, M.N. and Clark, A.R. "Lot sizing and furnace scheduling in small foundries", *Comput. Operat. Res.*, **35**(3), pp. 916-932 (2008).

20. Nawaz, M., Enscore, E.E. and Ham, I. "A heuristic algorithm for the m-machine, n-job flowshop sequencing problem", *OMEGA*, **11**(1), pp. 91-95 (1983).

21. Kurz, M.E. and Askin, R.G. "Comparing scheduling rules for flexible flow lines", *Int. J. Prod. Eco.*, **85**(3), pp. 371-388 (2003).

22. Kurz, M.E. and Askin, R.G. "Scheduling flexible flow lines with sequence-dependent setup times", *Eur. J. Oper. Res.*, **159**(1), pp. 66-82 (2004).

23. Goren, H.G., Tunali, S. and Jans, R. "A review of applications of genetic algorithms in lot sizing", *J. Intell. Manuf.*, **21**(4), pp. 575-590 (2010).

24. Jans, R. and Degraeve, Z. "Meta-heuristics for dynamic lotsizing : A review and comparison of solution approaches", *Eur. J. Oper. Res.*, **177**(3), pp. 1851-1875 (2007).

25. Defersha, F.M. and Chen, M. "A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality", *Eur. J. Oper. Res.*, **187**(1), pp. 46-69 (2008).

BIOGRAPHIES

Mohammad Mohammadi was born in Tehran, Iran in 1977. He received his B.S. degree in Industrial Engineering from Iran University of Science and Technology, Tehran, in 2000, and his M.S. and Ph.D. degrees in Industrial Engineering from Amirkabir University of Technology, Tehran, in 2002 and 2009, respectively. He currently gives a lecture in the Department of Industrial Engineering at Tarbiat Moallem University, Karaj. His research and teaching interests include: Sequencing and Scheduling, Production Planning, Time Series, Metaheuristics and Supply Chains.

Seyyed Mohammad Taghi Fatemi Ghomi was born in Ghom, Iran, in 1952. He received his B.S. degree in Industrial Engineering from Sharif University, Tehran, in 1973, and a Ph.D. degree in Industrial Engineering from Bradford University, England in 1980. He is currently Professor in the Department of Industrial Engineering at Amirkabir University of Technology in Tehran. His research and teaching interests include: Stochastic Activity Networks, Production Planning and Control, Scheduling, Queuing Systems, and Statistical Quality Control.