Journal of Industrial and Systems Engineering
Vol. 5, No. 1, pp 1-19
Spring 2011

A Flow shop Production Planning Problem with basic period policy and Sequence Dependent set up times

Mohammad Aliabadi[1], Fariborz Jolai[2*], Esmaeil Mehdizadeh[3], Masoud Jenabi[4]

[1,3]Department of Industrial Engineering, Islamic Azad University of Qazvin, Qazvin, Iran
[1]Md.aliabadi@gmail.com, [3]emqiau@yahoo.com

[2] Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
fjolai@ut.ac.ir

[4]Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
m.jenabi@aut.ac.ir

ABSTRACT

Many authors have examined lot sizing, scheduling and sequence of multi-product flow shops, but most of them have assumed that set up times are independent of sequence. Whereas dependence of set up times to sequence is more common in practice. Hence, in this paper, we examine the discussed problem with hypothesis of dependence of set up times to sequence and cyclic schedule policy in basic period form. To do so, a mixed integer non-linear programming (NLP) model is developed for this problem. To solve the model these techniques are applied: Heuristic G-group for determining the frequency of item production and assigning product to periods and three meta heuristic methods including hybrid Particle swarm optimization , hybrid Vibration damping optimization hybrid genetic algorithm are used to determine the sequence and economic lot sizes of each item. In addition, to compare these methods, some random problems are produced and computation of them shows the substantial superiority of hybrid Particle swarm optimization.

Keywords: Flow shop, Determining lot sizes, Finite planning horizon, Hybrid particle swarm optimization, Basic period.

1. INTRODUCTION

Smooth and cost-efficient running of a production line often depends on selecting appropriate lot-sizes and production schedules. One of the most difficult lot-sizing problems which is known to be NP-hard is Economic lot scheduling problem (ELSP) (Hsu, 1983). It is related to determining lot sizes and multi-product scheduling of production in a single stage production facility so that the demands are met with no shortage and also average inventory holding and set up costs are minimized. This problem is applicable in many production systems like metal forming, plastic injection, clothes industries and assembly lines. Most of the related contributions reported in the

---

[*] Corresponding Author

ISSN: 1735-8272, Copyright © 2011 JISE . All rights reserved.

literature usually have focused on cyclic schedules (i.e., schedules that are repeated periodically) with three famous scheduling policies:

- Common cycle (CC): In this form, the cycle between two runs of production is the same for all production items. The chief advantage of this method is the accessibility of a feasible schedule due to removing the time overlap of production considering a common cycle between two production runs, for all products if one exists (Hannsman, 1962). But because of inequality of demand rates, production and set up costs, this approach does not consider optimal solutions (Wagner and Davis, 2001) and delivers an upper bound of costs.
- Basic period: In this form, various products have different cycle times but cycle time of each product is an integer multiplier of a basic period (BP) that is long enough to fulfill the demand of all items. To clarify this approach we can say that BP is the time length devoted to set up and production of all or a subset of products. A solution for this approach consists of a set of integer multiplier $\{k_i | i = 1,...,n\}$ and the length of basic period. Using this solution, the time cycle of replenishment for each product can be computed as $T_i = k_i \times F$. In addition, due to the deference of production frequency of products ($k_i s$),planning horizon should be considered as cycles with length of $H \times F$ that are called global cycle time where H is the least common multiplier of values $k_i s$ and F is time length BP. In general, it gives better solutions than previous type but problem is an NP-hard one (Ouenniche and Boctor, 2001c).
- Time varying lot size: It allows multiple runs for each product at each cycle with different lot sizes over a cyclic schedule, and always gives a feasible schedule if one exists. Modeling a problem with this approach is more complicated than the other approaches, but usually gives better solutions (Moon et al 2002).

Reviewing the literature, we can see that some authors have investigated ELSP with dependence of set up times and costs in single facility production environment. Some of them are as follows:

Maxwell (1964) has investigated ELSP considering sequence dependent set up times and optimized a quadratic criteria related to a series of linear constraints. He has proposed an approximate solution for this problem. Irani and Gunaseenal (1988) have discussed the set up time structure as a family for ELSP. Dobson and Yano (1994) applied time-varying policy to solve the set up time dependent sequence ELSP. They used a lagrangian relaxation of the formulation which leads to a partial separation of the embedded lot sizing and traveling salesman problem. The relaxation results in a new combinational problem related to the minimum spanning tree problem. The information about frequency of production, obtained from this relaxation, is used to find heuristic solution for the entire problem. Wagner and Davis (2001) developed a heuristic method which provided a range of solutions and management can choose one of them which are useful for production environment. This heuristic outperformed Dobson method. Honng-chooun and Karimi (2001a, 2001b) in two papers divided the problem to two subsections including lot scheduling and sequence determination. For the first one they suggested a non linear integer programming and for the second one, they used a tabu search algorithm.

But also, ELSP in multi stage systems has been paid attention by researchers. EL-Najdawi and Kleindorfer (1993) studied ELSP in multi-product flow shops and common cycle time approach for items. They developed an optimization frame to determine a common cycle and production scheduling. Ouenniche and Boctor (1999) studied the impact of sequence on ELSP in flow shops.

They proposed a mixed integer non linear programming and heuristic algorithm to determine the sequence and the length of common cycle T. Torabi et al (2004) considered ELSP in flexible job shops with identical machines and common cycle approach for items in a finite planning horizon and proposed an optimal solution for the problem. Also, they (2006) investigated lot and delivery scheduling problem in a simple supply chain where a single supplier produces multiple components on a flexible flow line and delivers them directly to an assembly facility. They proposed a hybrid genetic algorithm for finding near optimal solutions. Jenabi et al (2007) have considered the production-scheduling problem in flexible flow shops with unrelated machines and presented two heuristic algorithms to solve the problem. Akrami et al (2006) added intermediate buffers between work stations to problem and limiting the buffers causes blocking the flow of downstream machines. They used genetic algorithm (GA) and Tabu search (TS) to solve the problem.

Ouenniche et al (2001a, 2001b, 2001c) used three heuristic methods to determine the economic lot sizing and scheduling problem with the approach of basic period in flow shops for cyclic scheduling. The methods were G-group, two group and Powers of two. Torabi and Jenabi (2008) addressed the problem of lot sizing, scheduling, and delivery of several items with basic period approach in a two-echelon supply chain over a finite planning horizon. Single supplier produces the items through a flexible flow line and delivers them directly to an assembly facility where the transfer of sub-lots between adjacent stages of supplier's production system (i.e., lot streaming) is allowed in order to decrease the manufacturing lead time.

The best of our knowledge up to now is a paper found related to production system of flow shops with assuming sequence dependent set up times and common cycle for items that Heidari and torabi (2008) developed a mixed integer programming model for problem and a GA for finding solutions.
However, in this paper, we consider the production scheduling problem in flow shop systems and with assuming sequence dependent set up times and basic period approach for items where all parameters (such as demand rates) are deterministic and constant over a given finite planning horizon.

The outline of this paper is as follows. In section 2, some general information about the problem is presented. In section 3 the model is developed. Proposed method to solve the model is mentioned in the section 4. Computational results are expressed in section 5. Conclusion and remarks is the last part of this research.

2. PROBLEM STATEMENT

The some general information of our problem is explained in the following lines:

- Lot size for each product in different stages is the same.
- Machines are accessible continually and each machine can process a product at a special time.
- The manufacturing system pays a linear inventory holding cost for semi-product and end product.
- Preemption is not allowed, that is, at a given stage, once the processing of a lot has started, it must be completed without interruption.
- Lot streaming is not allowed namely sub batches of each product can't be transferred to next stage, before all of process finishes.

- Buffers are unlimited between two subsequent stages.
- Zero switch rule is applied namely the production of each product is started when its inventory reaches zero.
- Capacity and resource constraint doesn't exist in production stages so at least one feasible lot schedule is possible.
- Planning horizon covers the production cycles.
- All parameters of the problem such as demand and production rates set up times and inventory holding costs are deterministic and constant over planning horizon.
- Planning horizon is assumed to be finite and its length is determined by management.
- Shortage is not allowed.
- End product delivery is done continuously.
- Set up time structure and costs are sequence dependent.
- External demand is just for end product.

3. PROBLEM FORMULATION

The notations used for the problem formulation are defined as follows:

Parameters

n : number of products

m : number of stages (machines)

i,u : products index

j : stage (machine) index

d_i : demand rate of product

p_{ij} : production rate of product *i* in stage *j*

t_{ij} : process time for product lot *i* in stage *j*

S_{uij} : set up time originated from switching product u to *i* in stage *j*

C_{uij} : set up cost originated from switching product u to *i* in stage *j*

h_{ij} : inventory holding cost of each unit of work in process (WIP) between two subsequent stages of *j* and *j+1* per time unit

h_i : inventory holding cost per unit of finished product *i* per time unit

M : a very big number

PH : the length of finite planning horizon

Decision variables

F : the length of basic period.

k_i : time multiplier of product *i* that it is a integer number

T_i : length of the time between two subsequent runs of product *i* ,called cycle time of product *i* $(T_i = k_i \times F)$

H : the least common multiple k_i s

r : the number of global cycles ($H \times F$) during planning horizon

σ_k : the vector of assigned products to basic period *k*

σ_{kj} : sequence vector of products in basic period *k* on machine *j*

n_k : the number of products assigned to basic period *k*

Q_i : production lot of product *i* in different stages ($Q_i = d_i \times T_i$)

b_{ij} : start time of processing product in stage *j* (after related set up operation)

$$x_{iljk} = \begin{cases} 1 & \text{If product } i \text{ is assigned to position } l \text{ in } \sigma_{kj} \\ 0 & \text{Other wise} \end{cases}$$

Since there is value added for item in each stage, h_{ij} is non-decreasing, i.e., $h_{i,j-1} \le h_{ij}$.

Now, we can model the problem in the form of a binary mixed non-linear problem (BMNLP). The objective is to minimize the average set up cost and inventory holding costs WIP and end product. Set up cost for product other than ones produced at the beginning of each period is calculated as follows:

$$C_{uij}\, x_{uljk}\, x_{i,l+1,j,k} \quad i,u = 1,\dots,n_k \quad i \ne u \quad l = 1,\dots,n_k - 1 \quad j = 1,\dots,m \quad k = 1,\dots,H \quad H = lcm(k_1,\dots,k_n) \tag{1}$$

We suppose that set up cost at beginning of production in first basic period from global cycle is equal to last product in σ_{Hj} to first product in σ_{1j} $(j = 1,\dots,m)$.

$$C_{uij}\, x_{u,n_H,j,H}\, x_{i1j1} \quad i,u = 1,\dots,n_k \quad i \ne u \quad j = 1,\dots,m \quad k = 1,\dots,H \quad H = lcm(k_1,\dots,k_n) \tag{2}$$

And set up cost at the beginning of production except to first basic period is equal to last product in σ_{kj} to first product in $\sigma_{k+1,j}$ $(k = 1,\dots,H-1 \, and \; j = 1,\dots,m)$.

$$C_{uij}\, x_{u,n_H,j,k}\, x_{i,1,j,k+1} \quad i,u = 1,\dots,n_k \quad i \ne u \quad j = 1,\dots,m \quad k = 1,\dots,H \quad H = lcm(k_1,\dots,k_n) \tag{3}$$

So total set up cost per unit time are like this:

$$TC_{Set\,up}=\frac{1}{HF}\left(\begin{array}{l}\sum_{j=1}^{m}\sum_{l=1}^{n-1}\sum_{i=1}^{n}\sum_{\substack{u=1\\u\neq i}}^{n}\sum_{k=1}^{H}C_{uij}\,x_{u\,lj\,k}\,x_{i,l+1,j,k}+\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{\substack{u=1\\u\neq i}}^{n}\sum_{k=1}^{H-1}C_{uij}\,x_{u\,n\,k\,j\,k}\,x_{i,1,j,k+1}\\ \quad+\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{u=1}^{n}C_{uij}\,x_{u\,n\,H\,j\,H}\,x_{i\,1\,j\,1}\end{array}\right) \tag{4}$$

Also, two kinds of inventory are considered, WIP and end product inventories. These can be seen in Figures 1&2.

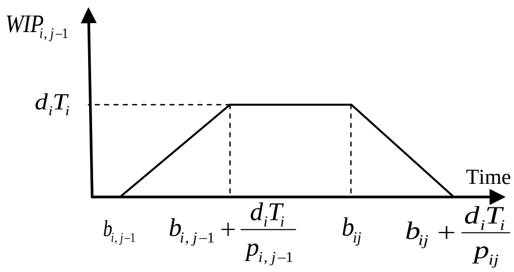

Figure 1 diagram of level changes of WIP inventory *i* between stages *j-1* and *j*

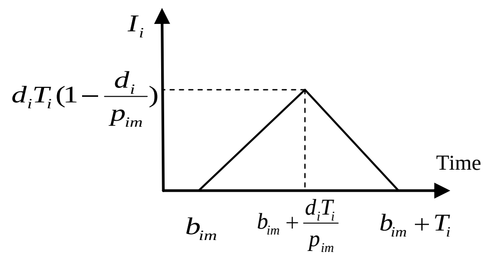

Figure 2 diagram of level changes of end product inventory *i*

Figs 1 and 2 show the evolution of WIP inventory between stages *j-1*and *j* and the inventory level of end product *i* at one cycle, respectively. From Figs. 1 and 2 can seen that the average inventory levels per time unit are:

$$WIP_{i,j-1}=\frac{1}{2}d_i^2k_iF\left(\frac{1}{p_{ij}}-\frac{1}{p_{i,j-1}}\right)+d_i\left(b_{ij}-b_{i,j-1}\right) \tag{5}$$

$$I_i=\frac{1}{2}k_id_i\left(1-\frac{d_i}{p_{im}}\right)F \tag{6}$$

Therefore, total inventory holding costs of all products at time unit is:

$$TC_{Holding\ costs}=\sum_{i=1}^{n}\left[h_i\frac{k_id_i}{2}\left(1-\frac{d_i}{p_{im}}\right)+\frac{k_id_i^2}{2}\sum_{j=2}^{m}h_{i,j-1}\left(\frac{1}{p_{ij}}-\frac{1}{p_{i,j-1}}\right)\right]F+\sum_{i=1}^{n}\sum_{j=2}^{m}h_{i,j-1}d_i\left(b_{ij}-b_{i,j-1}\right) \tag{7}$$

Thus we can model the problem like this:

$$\begin{array}{l}Min\ Z=\sum_{j=1}^{m}\sum_{l=1}^{n-1}\sum_{i=1}^{n}\sum_{\substack{u=1\\u\neq i}}^{n}\sum_{k=1}^{H}\frac{1}{HF}C_{uij}\,x_{u\,l\,j\,k}\,x_{i,l+1,j,k}+\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{\substack{u=1\\u\neq i}}^{n}\sum_{k=1}^{H-1}\frac{1}{HF}C_{uij}\,x_{u\,n\,k\,j\,k}\,x_{i,1,j,k+1}\\ \quad+\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{\substack{u=1\\u\neq i}}^{n}\frac{1}{HF}C_{uij}\,x_{u\,n\,H\,j\,H}\,x_{i\,1\,j\,1}+\sum_{i=1}^{n}\left[h_i\frac{k_id_i}{2}\left(1-\frac{d_i}{p_{im}}\right)+\frac{k_id_i^2}{2}\sum_{j=2}^{m}h_{i,j-1}\left(\frac{1}{p_{ij}}-\frac{1}{p_{i,j-1}}\right)\right]F+\sum_{i=1}^{n}\sum_{j=2}^{m}h_{i,j-1}d_i\left(b_{ij}-b_{i,j-1}\right)\\ Subject\ to:\\ b_{i,j-1}+\frac{d_ik_iF}{p_{i,j-1}}\le b_{ij}\qquad i=1,2,....,n\qquad j=2,....,m\end{array} \tag{8}$$

$$b_{uj} + \frac{d_u k_u F}{p_{u\,j}} + S_{uij} - b_{ij} \le M\left(2 - x_{u\,lj\,k} - x_{i,l+1,j,k}\right) \quad i = 1,2,....,n \quad u \ne i \quad l < n \quad j = 1,2,....,m \quad k = 1,2,....,H \quad H = lcm\left(k_1, k_2, ..., k_n\right) \tag{9}$$

$$\sum_{i=1}^{n} x_{i\,lj\,k} \le 1 \qquad j = 1,2,...,m \quad l = 1,2,....,n \quad k = 1,2,....,H \quad H = lcm\left(k_1, k_2, ..., k_n\right) \tag{10}$$

$$\sum_{i=1}^{n} x_{i,l+1,j,k} \le \sum_{u=1}^{n} x_{uljk} \quad u \ne i \quad l < n \quad j = 1,2,....,m \quad k = 1,2,....,H \quad H = \left(k_1, k_2, ..., k_n\right) \tag{11}$$

$$\sum_{l=1}^{n} x_{i\,l\,j\left(t+bk_i\right)} = \sum_{l=1}^{n} x_{i\,l\,j\left(t+(b+1)k_i\right)} \quad i = 1,2,....,n \quad j = 1,2,....,m \quad t = 1,2,....,k_i \quad b = 0,1,....,\frac{H}{k_i} - 2 \tag{12}$$

$$\sum_{l=1}^{n} x_{iljk} = \sum_{l=1}^{n} x_{i,l,j+1,k} \qquad i = 1,2,...,n \quad j = 1,2,....,m \quad j < m \quad k = 1,2,....,H \quad H = \left(k_1, k_2, ..., k_n\right) \tag{13}$$

$$b_{ij} \ge S_{uij} - M\left(2 - x_{u\,nH\,jH} - x_{i\,1j\,1}\right) \quad i = 1,2,....,n \quad u = 1,2,...,n \quad u \ne i \quad j = 1,2,....,m \quad H = lcm\left(k_1, k_2, ..., k_n\right) \tag{14}$$

$$b_{ij} \ge S_{uij} - M\left(2 - x_{u\,nk\,j\,k} - x_{i,1,j,k+1}\right) \quad i = 1,2,....,n \quad u = 1,2,...,n \quad u \ne i \quad j = 1,2,....,m \quad k = 1,2,...,H-1 \qquad H = lcm\left(k_1, k_2, ..., k_n\right) \tag{15}$$

$$b_{im} + \frac{d_i k_i F}{p_{im}} \le F \qquad i = 1,2,....,n \tag{16}$$

$$H \times F \times r = PH \qquad H = \left(k_1, k_2, ..., k_n\right) \qquad r \ge 1 \ \ and\ \text{int} \tag{17}$$

$$F \ge 0; \quad b_{ij} \ge 0\ \forall i, j\ ; \quad x_{iljk} = \{0,1\}\ \forall i, l, j, k \tag{18}$$

Constraint 8 shows that at each stage, no product can be produced before it is completed at the previous stage. Constraint 9 expresses that at each stage, no product can be processed before completion of its predecessor in the production sequence. Constraint 10 states that in basic period k in stage j, and lth position at most just one product can exist. Constraint 11 assures that in each stage and basic period, one product can be assigned position of $l+1$ if another product is to be assigned to the position of l. Constraint 12 determines the assignment of products in proper basic periods during H basic periods. Constraint 13 shows that if product i in basic period k is assigned to one of the positions of σ_{kj}, it should be given to one of the positions $\sigma_{k,j+1}$. Constraint 14 states that in the first basic period, processing first product in sequence vector of each stage after its set up time is possible. Constraint 15 is like the previous one but is applied to basic periods except the first one. Constraint 16 assures the completion time of each product in final stage is less than or equal to basic period time. Constraint 17 shows that planning horizon is an integer multiplier of global cycle ($H \times F$). Constraint 18 shows that decision variables are not negative.

4. PROPOSED METHOD TO SOLVE THE PROBLEM

It's necessary to mention that solving our model needs knowing the amounts of k_is definitely. Moreover, these amounts are from key decision variables that should be determined from solving model but no explicit attitude exists directly to solve problem even for small size problems (Torabi and Jenabi, 2008). Thus to meet these goals, should lean on heuristics to find a good solution. In this way, to determine the k_i s amounts and vectors σ_k , G-group method is used (presented by Qunniche et al 2001c) but due to sequence dependent set up time we modify independent cycle time as follows:

$$T_i^* = Min\left[\max\left\{\sqrt{\frac{2\bar{C}_i}{h_i d_i\left(1-{d_i}/{p_{im}}\right)+\sum_{j=2}^{m} h_{i,j-1} d_i^2\left({1}/{p_{ij}}+{1}/{p_{i,j-1}}\right)}}\ ,\max\left\{\frac{\bar{S}_{ij}}{1-\frac{d_i}{p_{ij}}}\right\}\right\},PH\right] \quad j=1,2,\ldots,m \tag{19}$$

Where $\bar{C}_i = \frac{\sum_{j=1}^{m}\sum_{u=1}^{n} C_{iuj}}{mn}$, $\bar{S}_{ij} = \frac{\sum_{u=1}^{n} S_{iuj}}{n}$ $\forall j$

For determining sequence, lot size and production scheduling, hybrid vibration damping optimization, hybrid particle swarm optimization and hybrid genetic algorithm are used.

4.1. Hybrid particle swarm optimization (HPSO)

Particle swarm optimization (PSO) was developed by Kenedy and Eberhart (1995) for the first time to optimize continuous nonlinear functions. This is a meta heuristic method based on population in order to find minimal objective function. In this algorithm, each member of swarm called particle, moves in search space. Movement of particles in search space is influenced by experience and knowledge of their own and neighbors. Therefore status of other particles of a group affects the way to search a particle. Particle swarm optimization is based on this matter that at every moment, each member adjusts its position in search space according to the best its position and the best position its whole neighbor.

In this algorithm a repeat process is done to improve the solution. During this process, group members, evaluate the fitness of various solutions continually and gradually moves to areas where they have the best position. Then, the best found solution by all group members is identified as the best position. Each group member delivers its information to its neighbors, so they are able to recognize positions in which other members have been more successful.

Usually, a group is modeled in multi dimensional by using position and velocity. In fact, members of a group communicate to other members that have better positions and adjust their position and velocity. Hence, a particle needs the following information to make a suitable change in position and velocity:

1. Current velocity of the member.
2. Distance from the best position of the member up to now.
3. Distance from the best position in the whole feasible space up to now.

In this research, we extend the proposed PSO by Tasgetiren et al (2004) for our problem to find sequences that optimize objective function.

One of the most important subjects in designing PSO is the representation of solutions. Due to this, we construct a direct relation between problem domain and particles, thus position of each particle and its velocity are showed as H matrixes $m \times n_k$. In HPSO, there are ρ particles in the population. A particle φ at iteration t can be showed as $X_{\varphi}^{t} = \left(\left[x_{\varphi ji}^{t}\right]_1, \ldots, \left[x_{\varphi ji}^{t}\right]_H\right), i = 1, \ldots, n_k; j = 1, \ldots, m$ where $\left[x_{\varphi ji}^{t}\right]_k$ is matrix of particle position compared to *i* and *j* dimensions in the basic period *k*. Since PSO is a continuous optimization algorithm we use the smallest position value (SPV) rule to convert continuous position matrix to a discrete sequence matrix. Figure 3 illustrates how this happens and how to reach the sequence matrixes of a particle.

According to SPV rule, after sorting the first row of particle position matrix (matrix on the left) in ascending order, indexes relevant to these amounts become (4,3,2,1) and if vector σ_1 (products vector assigned to first basic period) is sorted according to recent vector, sequence vector becomes (5,3,2,1).

The HPSO search mechanism iteratively explores a solution space by adjusting particle velocity. For proposed HPSO, velocity of particle φ can be showed as $V_{\varphi}^{t} = \left(\left[v_{\varphi ji}^{t}\right]_1, \ldots, \left[v_{\varphi ji}^{t}\right]_H\right)$ where $\left[v_{\varphi ji}^{t}\right]_k$ is matrix of particle velocity compared to *i* and *j* dimensions in the basic period *k*.

To update velocity each particle at any iteration using from the following equation:

$$[v_{\phi ji}^{t}]_k = w\,[v_{\phi ji}^{t-1}]_k + c_1 r_1\left([p_{\phi ji}^{t-1}]_k - [x_{\phi ji}^{t-1}]_k\right) + c_2 r_2\left([g_{ji}^{t-1}]_k - [x_{\phi ji}^{t-1}]_k\right) \quad \phi = 1, \ldots, \rho \quad j = 1, \ldots, m \quad i = 1, \ldots, n_k \quad k = 1, \ldots, H \tag{20}$$

σ_1:

1	2	3	5

σ_2:

1	2	4	6	7

X_{φ}^{t}:

1.5	0.99	-1.66	-2.34
2.9	-3.81	1.85	-0.3
-3.78	2.5	-1.38	-1.91

2.18	3.56	1.57	-0.26	3.47
-2.29	0.21	-1.46	0.79	2.16
-3.37	-1.26	1.41	1.4	2.66

5	3	2	1
2	5	3	1
1	5	3	2

6	4	1	7	2
1	4	2	6	7
1	2	6	4	7

Figure 3 an example of how to convert particle position matrixes to sequence matrixes

Where the *w* is inertia weight, c_1 is the cognitive parameter, c_2 is the social parameter, r_1 and r_2 are random numbers uniformly distributed in U(0,1), $P_{\varphi}^{t} = \left(\left[p_{\varphi ji}^{i}\right]_1, \ldots, \left[p_{\varphi ji}^{i}\right]_H\right)$ denotes position matrixes set of best solution that particle φ has attained until t, and $G_{\varphi}^{t} = \left([g^{i}_{ji}]_1, \ldots, [g^{i}_{ji}]_H\right)$ denotes position matrixes set of best solution in population until t.

Also, to update particle position is used from the following equation:

$$[x^{t}_{\phi ji}]_k = [x^{t-1}_{\phi ji}]_k + [v^{t}_{\phi ji}]_k \qquad \phi = 1,\ldots,\rho \quad j = 1,\ldots,m \quad i = 1,\ldots,n_k \quad k = 1,\ldots,H \tag{21}$$

4.1.1. Primitive population

Primitive elements of position and velocity matrixes are randomly generated as follows:

$$[x^{0}_{\phi ji}]_k = x_{\min} + (x_{\max} - x_{\min}) \times r_1 \quad \phi = 1,\ldots,\rho \quad j = 1,\ldots,m \quad i = 1,\ldots,n_k \quad k = 1,\ldots,H \tag{22}$$

Then, sequence matrixes for particles position matrixes are attained by SPV rule.

$$[v^{0}_{\phi ji}]_k = v_{\min} + (v_{\max} - v_{\min}) \times r_2 \quad \phi = 1,\ldots,\rho \quad j = 1,\ldots,m \quad i = 1,\ldots,n_k \quad k = 1,\ldots,H \tag{23}$$

4.1.2. Fitness function

Fitness function shows the efficiency of each potential solution. In our problem, after recognition of σ_{kj} vectors, fitness value for each particle achieved by solving nonlinear model P_1. Problem P_1 is derived by replacing amounts of variables x_{iljk} adopted with σ_{kj} vectors.

Model P_1

$$Min\ Z = \frac{A}{HF} + + \sum_{i=1}^{n}\left[h_i \frac{k_i d_i}{2}\left(1 - \frac{d_i}{p_{im}}\right) + \frac{k_i d_i^2}{2}\sum_{j=2}^{m} h_{i,j-1}\left(\frac{1}{p_{ij}} - \frac{1}{p_{i,j-1}}\right)\right]F + \sum_{i=1}^{n}\sum_{j=2}^{m} h_{i,j-1} d_i \left(b_{ij} - b_{i,j-1}\right)$$

Subject to:

$$b_{i,j-1} + \frac{d_i k_i F}{p_{i,j-1}} \le b_{ij} \qquad i = 1,2,\ldots.,n \qquad j = 2,\ldots.,m \tag{24}$$

$$b_{\sigma kj(i-1),j} + \frac{d_{\sigma kj(i-1),j}\, k_{\sigma kj(i-1),j}\, F}{p_{\sigma kj(i-1),j\ j}} + S_{\sigma kj(i-1),\sigma kj(i),j,j} \le b_{\sigma kj(i),j} \quad i = 2,\ldots.,n_k \quad j = 1,2,\ldots,m \\ k = 1,\ldots,H \quad H = lcm(k_1,\ldots,k_n) \tag{25}$$

$$b_{\sigma 1j(1),j} \ge S_{\sigma Hj(nH),\sigma 1j(1),j} \qquad j = 1,2,\ldots,m \quad H = lcm(k_1,\ldots,k_n) \tag{26}$$

$$b_{\sigma k+1,j(1),j} \ge S_{\sigma k+1,j(1),\sigma kj(nk),j} \quad j = 1,2,\ldots,m \quad i = 2,\ldots.,n_k \quad k = 1,\ldots,H-1 \quad H = lcm(k_1,\ldots,k_n) \tag{27}$$

$$b_{im} + \frac{d_i k_i F}{p_{im}} \le F \qquad i = 1,2,\ldots.,n \tag{28}$$

$$H \times F \times r = PH \qquad H = \left(k_1, k_2, \ldots., k_n\right) \qquad r \ge 1 \ \ and \text{ int} \tag{29}$$

$$F \ge 0; \quad b_{ij} \ge 0\ \forall i,j\ ;$$

Where A is total set up cost. Model P_1 can be solved through proposed iterative procedure by Torabi and Jenabi (2008) like this:

- Initial step: Let *r=1* and solve the associated linear problem. Let Z_r denote the optimal objective value.
- Iterative step: Let *r=r+1* and solve the corresponding linear problem. If this model has no feasible solution, stop; otherwise, let Z_{r+1} denote the current optimal objective value. If $Z_{r+1} < Z_r$ set $F^* = \frac{PH}{r \times H}$; otherwise, the best objective value remains unchanged. Repeat the iterative step.

It's necessary to mention that this method operates as an explicit enumeration method and assures the optimal solutions for our model.

4.1.3. Personal best

At the beginning of algorithm, position matrixes for each particle are considered as the personal best and are updated other iteration.

4.1.4. Global best

At the beginning of algorithm, position matrixes related to best particle of swarm are considered and updated in other iterations.

4.1.5. Hybrid PSO with variable neighborhood search (VNS)

In order to obtain better solution for our problem, PSO with proposed VNS by Mladenovic and Hansen (1997) is combined. Global best at iteration is considered as initial solution to execute this algorithm. In this research, from interchange operator to create neighbor and from shift operator to find local optimum is used (for more details see Gen and Cheng, 1997).

4.1.6. Stop criterion

The stop criterion determines when HPSO will stop. The HPSO stops if predetermined number of iteration is reached.

Pseudo code HPSO algorithm is given as following:

```
Initialize parameters
Initialize population
Find sequence matrixes by SPV rule
Evaluate the fitness of each particle in the swarm
Do {
  Find the personal best
  Find the global best
```

Find sequence matrixes by SPV rule

Evaluate the fitness of each particle in the swarm

Apply VNS for the global best

Update velocity

Update position

} while (termination condition)

4.2. Proposed hybrid vibration damping optimization (HVDO)

Vibration damping optimization (VDO) was firstly developed by Mehdizadeh and Tavakkoli Moghadam (2008) which originated from mechanism vibration damping process.

VDO algorithm acts in this way: begins from a random start solution (in primitive domain), then new solution is generated and evaluated in the neighborhood of previous solution by using a neighboring suitable structure. If the new solution is better as compared to preceding solution, namely it decreases the vibration energy, it accepts as the current solution; otherwise accepts with Rayleigh probability. This action allows our system escape from convergence to local optimum. These steps continue until reaching to the force loop stop criterion, after that damping scheme is applied to decrease vibration domain and again continue above steps till reaching to the damping loop stop criterion.

4.2.1. Representation of solution

Each solution of this algorithm is considered as H matrixes $m \times n_k$. Each row of them shows the sequence of products on each machine in each basic period.

4.2.2. Initial solution

In proposed VDO algorithm by Mehdizadeh and Tavakkoli Moghadam (2008) only one of solution is considered, but because of having more diversity in solutions and possibility of search in solution space, HVDO is started with a randomly generated population of L solutions.

4.2.3. Neighboring search mechanism for each solution

For the movement in solution space and discovering better solution, we need to design operators such that can produce the neighboring solution by perturbation solutions. Nearchou (2004) referred some of these operators. In this research, to make perturbation in each solution, random interchange operator has been used.

4.2.4. Fitness function

For calculation fitness function of each solution in this algorithm, the method introduced in section 4.1.2 has been applied.

4.2.5. Iterated Hill Climbing (IHC) method

To improve the performance of the algorithm and to avoid convergence to local optimum, proposed IHC method by Nearchou (2004) is used. In fact, after establishing neighborhood population, and before reducing the vibration domain, IHC method is used randomly on neighborhood population with predetermined probability (P_{IHC}). Implement of this method on solution X that has been chosen randomly among solutions population are as following:

- $BEST \leftarrow X$
- Two products having the most process time in X are placed in two start position Y and other products of X are copied in same position of Y.
- If we assume that Y has n members, $n-2$ times for products put in position 3 to n two products are randomly chosen and interchanged with each other and each time, their fitness function is evaluated. If this amount is less than fitness function *BEST* then $BEST \leftarrow Y$.
- Returning *BEST* solution.

4.2.6. Stop criterion

To stop the algorithm in this research, vibration domain is so reduced that a predetermined small number is reached.

Pseudo code proposed HVDO is given as following:

Primitive step:

$A \leftarrow A_0$ / Initialize initial vibration domain/

$\gamma \leftarrow \gamma_0$ / Initialize Parameter of vibration damping /

$\sigma \leftarrow \sigma_0$ / Initialize parameter of Rayleigh probability/

$A_{min} \leftarrow A_f$ /Initialize final vibration domain/

$t_{max} \leftarrow t_0$ / Initialize counter for force loop/

Generate a random initial population π of L solutions

Evaluate population π

Set $X^* \leftarrow best\ of\ population\ (\pi)$

Main step:

$z \leftarrow 0$

Do { /Damping loop/

For each solution X_k ($k=1,\ldots, L$)

$t \leftarrow 0$

Do { /Force loop/

$t \leftarrow t+1$

Apply suitable neighboring structure and generate solution X'_k

Evaluate X'_k

$E \leftarrow$ fitness (X'_k)-fitness (X_k)

If $E<0$ then

Set $X_k \leftarrow X'_k$

else

Generate a random number r uniformly between 0 and 1

If $r<1-e^{\frac{-A^2}{2\sigma^2}}$ then set $X_k \leftarrow X'_k$

End if

} while ($t<t_{max}$)

End for

For each solution X_k ($k=1,\ldots, L$)

Generate a random number r uniformly between 0 and 1

If $r < P_{IHC}$ then apply IHC method on X_k

End for

Set $X \leftarrow$ best population (π)

If fitness(X) < fitness (X^*) then set $X^* \leftarrow X$

$A \leftarrow Ae^{\frac{-\gamma t}{2}}$

$z \leftarrow z+1$

} while ($A>A_{min}$)

Return X^*

4.3. Hybrid genetic algorithm (HGA)

Also, we used hybrid genetic algorithm proposed by Torabi et al (2006) for solving problem. In this way, each solution is considered as H matrixes $m \times n_k$. Fitness function was calculated with method defined in section 4-1-2 and stop criterion is reaching to definite number of iterations.

5. PARAMETER SETTING

We used Taguchi method for parameter setting (see the paper of Chen et al, 2007). For determining of proper orthogonal array, was used Taguchi reference (1987).

5.1. HPSO parameter setting

We used parameter setting only for Cognitive factor, social factor, inertia weight, number of particles and stop criterion. The considered levels are as follows:

- Cognitive factor (c_1): four levels (1,2,3 and 4).
- Social factor (c_2): four levels (1,2,3 and 4).
- Inertia weight (w): four levels (0.8, 0.9, 1.1 and 1.2).
- Number of particles (ρ): two levels (*n* and *2n*).
- Stop criterion (ψ): two levels (*nm* and *2nm*).
- Other parameters in a level ($x_{\min} = -4$, $x_{\max} = 4$, $v_{\min} = -4$, $v_{\max} = 4$, $\beta = 0.975$).

According to the results of experiments, the optimal factors are $c_1 = 1, c_2 = 3,\ w = 1.1, \rho = 2n$ and $\psi = 2nm$.

5.2. HGA parameter setting

We used parameter setting only for population size, crossover probability, tournament selection parameter and stop criterion. The considered levels are as follows:

- Population size (pop_size): two levels (*n* and *2n*).
- Crossover probability (p_c): four levels (0.7, 0.75, 0.8 and 0.85).
- Mutation probability (p_m): one level (1- p_c).
- Tournament selection size (τ): four levels (0.5, 0.6, 0.7 and 0.8).
- Stop criterion (ψ): two levels (*nm* and *2nm*).

The optimal factors are $pop\_size = n$, $p_c = 0.7$, $p_m = 0.3, \tau = 0.8$ and $\psi = 2nm$.

5.3. HVDO parameter setting

We used parameter setting only for initial vibration domain, Rayleigh probability, final vibration domain and vibration damping parameter. The considered levels are as follows:

- Initial vibration domain (A_0): four levels (500, 5000, 50000 and 500000).
- Rayleigh probability (σ_0): four levels (500, 5000, 50000 and 500000).
- Final vibration domain (A_f): four levels (1, 0.1, 0.01 and 0.001).
- Parameter of vibration damping (γ_0): four levels (0.1, 0.3, 0.5 and 0.7).
- Other parameters in a level (*L*=2*n*, $t_{\max} = n, P_{IHC} = 0.01$).

The optimal factors are $A_0 = 50000$, $\sigma_0 = 5000$, $A_f = 0.001$ and $\gamma_0 = 0.5$.

6. NUMERICAL EXPERIMENTS

In this section, in order to evaluate and compare the performance of proposed algorithms, from point of view of quality and computation time of the solutions, we consider 6 sets of 5, 10 products and 2,5,10 stages; for each set of problems 20 random instances (120 problems) are generated and parameters needed for these problems are extracted from the following uniform distributions:

$d_i \sim U(50{,}500)$ $p_{ij} \sim \mathrm{U}(10000{,}20000)$ $\mathrm{h}_{i1} \sim \mathrm{U}(1{,}10)$ $\mathrm{S}_{uij} \sim \mathrm{U}(0.01{,}0.15)$

In addition, input parameter of G- group method is considered *3*. It's necessary to mention that h_{ij} accounts should increase with *j*. After random generation $h_{i1}\ \forall i$, for other stages value of h_{ij} is achieved from $h_{ij} = h_{i,j-1} + 5 \times U(1{,}5)$. It is also assumed that there is correlation between set up time and set up cost. So the values of C_{uij} are calculated as $C_{uij} = 15000 \times S_{uij} + 1000 \times U(0{,}1)$.

G-group method, HGA, HPSO and HVDO algorithms are coded in MATLAB programming language and are executed on a personal computer with a Pentium 4 processor running at 2 GHz. All of 120 instances of the problem are solved with HPSO, HGA and HVDO algorithms. Since we do not know the global optimal cost for each instance of the problem, due to nonlinear structure of our model and great computation time, hence for evaluating the quality of solutions, gotten total costs from HPSO, HGA and HVDO are compared with an associated lower bound (LB). We can calculate the performance rate by this relation $\theta = \frac{TC - LB}{LB} \times 100$ where the *TC* is the total cost of a problem instance obtained by each algorithm. To compute LB, we use the best sequence vector originated from algorithms. So variables x_{iljk}, are correspondingly replaced with *0* and *1*. Then, we omit the second set of constraints (constraints set 25 of Problem P_1) and solve the NLP model by iterative procedure expressed in 4-1-3 and the objective is adopted as the LB.
To compare the performance of proposed meta heuristic algorithms, λ index is used which consists of: $\lambda = \frac{S_{\mathrm{Alg}} - S_{\min}}{S_{\min}} \times 100$ where S_{Alg} and $S_{\min}$ are solutions by each algorithm and the best solution resulted by proposed algorithms, respectively.

Table 1 and 2 show the computational results for θ index and λ index, respectively. We make the following observation from numerical experiments:

- Comparing Meta heuristic methods, we observed that HPSO results are often better than HVDO and HGA.
- We observed that θ for instance problems increases when the problem size increases. This matter can happen because of an increase in the difference between lower bound and corresponding optimal costs or can be because of performance of proposed algorithms resulted from increase in corresponding solution space. Since corresponding optimal costs are not definite we can't remark on this context.
- We saw when the sequence vector is the same in all stages, performance rate decreases outstandingly.

- At last, proposed algorithms are considered as first tries to solve the problem and computational results indicate their effectiveness and efficiency of HPSO for obtaining optimal solutions or near optimal ones for small, medium size problems, and give at least feasible good solutions for large size problems in a rational time

Table 1 computational results for θ index

Problem size ($n \times m$)	The average θ for HPSO (%)	The average θ for HGA (%)	The average θ for HVDO (%)	The average CPU time of HPSO (s)	The average CPU time of HGA (s)	The average CPU time of HVDO (s)
5×2	1.78	1.93	2.8	151.09	208.90	346.85
5×5	2.51	3.22	5.89	518.02	596.51	555.16
5×10	2.91	6.44	4.4	1357.43	1193.50	1482.34
10×2	1.4	1.66	3.36	1081.39	1744.24	1708.95
10×5	5.86	7.9	9.17	2402.82	2204.36	2760.9
10×10	12.32	12.94	24.06	3265.64	3899.34	4315.02

Table 2 computational results for λ index

Problem size ($n \times m$)	The average λ for HPSO (%)	The average λ for HGA (%)	The average λ for HVDO (%)
5×2	1.36	1.61	3.92
5×5	0.33	3.73	11.03
5×10	0.5	2.4	4.88
10×2	0.23	4.86	7.11
10×5	0.72	3.71	3.36
10×10	2.06	4.69	10.68

7. CONCLUSION

In this paper, basic period approach for solving the economical lot size and scheduling problem is used where the flow shop production system exists and assumptions of set up time dependant sequence and finite planning horizon are considered. First, a mixed integer non linear model is developed to formulate the problem and to solve one optimally. Then, considering the Np-hardness of problem, G-group heuristic, hybrid particle swarm optimization (HPSO) and hybrid vibration damping optimization (HVDO) are used to solve the model.

Computational results show that proposed algorithms have a satisfactory performance, and the HPSO outperforms the HVDO with respect to the solution quality and computation time.

REFERENCES

[1] Akrami B., Karimi B., Moattar Hosseini S.M. (2006), Two meta heuristic methods for common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: The finite horizon case; *Applied Mathematics and computation* 183; 634-645.

[2] Chen Y.K., Hsieh K.L., Chang C.C. (2007), Economic design of the VSSI $\overline{X}$ control charts for correlated data; *International journal of production economics* 107; 528-539.

[3] Dobson G., Yano C.A. (1994), Cyclic scheduling to minimize inventory in batch flow line; *European Journal of Operational Research* 141; 441-461.

[4] Eberhard R.C., Kennedy J. (1995), A new optimizer using particle swarm theory; *In: Proceeding of the Sixth International Symposium on Micro Machine and human Science*, Japan; 39-43.

[5] El-Najdawi M., Kleindorfer P.R. (1993), Common cycle lot size scheduling for multi product multi stage production; *Management Science* 39; 872-885.

[6] Gen M., Cheng R. (1997), Genetic Algorithm and Engineering Design; John Wiley &Sons; New York.

[7] Hanssman F. (1962), Operation Research in Production and Inventory Control; John Wiley &Sons; New York.

[8] Heydari M., Torabi S.A. (2008), The Economic Lot Scheduling Problem in Flow Lines with Sequence dependent Set ups; *World Academy of Science, Engineering and Technology* 47; 473-479.

[9] Hong-Choon O.H., Karimi I.A. (2001), Planning production on a single processor with sequence dependent set ups: part 1; *Computer and chemical Engineering* 5; 1021-1030.

[10] Hong-Choon O.H., Karimi I.A. (2001), Planning production on a single processor with sequence dependent set ups: part 2; *Computer and chemical Engineering* 25; 1021-1030.

[11] Hsu W. (1983), The general feasibility test for the scheduling lot sizes for the several products on one machine; *Management science* 29; 93-105.

[12] Irani S.A., Gunaseenal U. (1988), Signal machine set up dependent sequencing using a set up complexity raking scheme; *Journal of manufacturing systems* 7; 11-23.

[13] Jenabi M., Fatemi Ghomi S.M.T., Karimi B., Torabi S.A. (2007), Two hybrid meta heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines; *Applied Mathematics and Computation* 186; 230-245.

[14] Maxwell W.L. (1964), The scheduling of economic lot sizes; *Navel Research Logistics Quarterly* 11; 89-124.

[15] Mehdizadeh E., Tavakkoli Moghaddam R. (2008), Vibration damping optimization; *Proceeding of the International Conference Operations Research and Global Business*, Germany, Septamber 3-5.

[16] Mladenovic N., Hansen P. (1997), Variable neighborhood search; *Computers and Operations Research* 24; 1097-1100.

[17] Moon I., Silver E.A., Choi S. (2002), Hybrid genetic algorithm for the economic lot scheduling problem; *International Journal of Production Research* 40; 809-824.

[18] Nearchou A.C. (2004), A novel meta heuristic approach for the flow shop scheduling problem; *Engineering Applications of Artificial Intelligence* 17; 289-300.

[19] Ounniche J., Boctor F.F. (1999), The impact of sequencing decisions on multi item lot sizing and scheduling in flow shops; *International Journal of Production Research* 37; 2253-2270.

[20] Ounniche J., Boctor F.F. (2001a), The two group heuristic to solve the multi product, economic lot sizing and scheduling problem in flow shops; *European Journal of Operational Research* 129; 539-554.

[21] Ounniche J., Boctor F.F. (2001b), The multi product, economic lot sizing problem in flow shops: the powers of two heuristic; *Computers and Operations Research* 28; 1165-1182.

[22] Ounniche J., Boctor F.F. (2001c), The G-group heuristic to solve the multi product, sequencing, lot sizing and scheduling problem in flow shops; *International Journal Production Research* 39; 81-98.

[23] Taguchi G. (1987), System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs; UNIPUB; New York, USA.

[24] Tasgetiren M.F., Liang Y.C., Sevkli M., Gencyilmaz G.(2004), Particle swarm optimization algorithm for makespan and maximum lateness minimization in permutation flow shop sequencing problem; *In: proceeding of the Fourth International Symposium on Intelligent Manufacturing Systems*, Sakarya, Turkey; 431-441.

[25] Torabi S.A., Karimi B.,Fatemi Ghomi S.M.T.(2004), The common cycle economic lot scheduling in flexible job shops: the finite horizon case; *International Journal of Production Economics* 97; 52-65.

[26] Torabi S.A., Fatemi Ghomi S.M.T., Karimi B. (2006), A hybrid genetic algorithm for the finite horizon economic and delivery scheduling in supply chains; *European Journal of Operational Research* 173; 173-189.

[27] Torabi S.A., Jenabi M. (2008), Multiple cycle economic and delivery scheduling problem in a two echelon supply chain; *The International Journal of Advanced Manufacturing Technology* 43(7-8); 785-798. DOI 10.1007/s00170-008-1752-6.

[28] Wagner J., Davis J. (2001), A search heuristic for the sequence dependent economic lot scheduling problem; *European Journal of Operational Research* 141; 133-146.