Journal of Optimization in Industrial Engineering, Vol.11, Issue 1, Winter and Spring 2018, 157-167
DOI: 10.22094/JOIE.2017.568.62

A Multi-Objective Mixed-Model Assembly Line Sequencing Problem with Stochastic Operation Time

Parviz Fattahi [a,*], Arezoo Askari[b]

[a] *Associate Professor, Department of Industrial Engineering, Alzahra University, Tehran, Iran*
[b] *M.sc., Department of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran*

Received 21 December 2016; Revised 14 June 2017; Accepted 20 September 2017

Abstract

In today's competitive market, those producers who can quickly adapt themselves to diverse demands of customers are successful. Therefore, in order to satisfy these demands of market, Mixed-model assembly line (MMAL) has an increasing growth in industry. A mixed-model assembly line (MMAL) is a type of production line in which varieties of products with common base characteristics are assembled on. This paper focuses on this type of production line in a stochastic environment with three objective functions: 1) total utility work cost, 2) total idle cost, and 3) total production rate variation cost that are simultaneously considered.In real life, especially in manual assembly lines, because of some inevitable human mistakes, breakdown of machines, lack of motivation in workers and the things alike, events are not deterministic, so weconsider operation time as a stochastic variable independently distributed with normal distributions; for dealing with it, chance constraint optimization is used to model the problem. At first, because of NP-hard nature of the problem, multi-objective harmony search (MOHS) algorithm is proposed to solve it. Then, for evaluating the performance of the proposed algorithm, it is compared with NSGA-II that is a powerful and famous algorithm in this area. At last, numerical examples for comparing these two algorithms with some comparing metrics are presented. The results have shown that MOHS algorithm has a good performance in our proposed model.

Keywords: Mixed-model assembly line sequencing, Stochastic operation time, Chance constraint.

1. Introduction

In today's competitive world, due to some challenges for producers, such as diversification in customers' demand, competitive price, etc., maintaining the marketplace becomes a vital issue for producers. In the past, manufacturers were able to produce a large quantity of products by using a single-model assembly line, but they are nowadays able to produce many products with high variety using mixed-model assembly lines (MMAL). This approach is feasible when different models can be assembled without a significant changeover delay between them Hyun et al. (1998). It is known that a sequencing problem in MMAL falls into NP-hard class of combinatorial optimization problems; thus, a large-sized problem may be computationally intractable Hyun et al. (1998). MMAL is widely used by production systems due to its advantages such as: more flexibility, better part usage rates, and their ability to answer various demands of their customers without possessing large product inventory in recent years Manavizadeh et al. (2013).

The effective utilization of a mixed-model assembly line requires solving two problems in the sequential manner: (1) designing and balancing the line; (2) determining the production sequence for different models Tavakkoli-Moghaddam & Rahimi-Vahed (2006). In this research, we assume that the assembly line has already been balanced and our goal is just sequencing the line.

In real-life application, especially in manual assembly lines, because of some inevitable human mistakes, breakdown of machines, lack of motivation in workers and the things alike, events are not deterministic; so, weconsider operation time as stochastic. Some approaches are introduced to model the stochastic problem in which one of them is chance-constrained approach, which was first introduced by Charnes and Cooper (1959). Based on Elyasi & Salmasi (2013), this approach is suitable for solving optimization problems with random variables in constraints and sometimes in objective functions as well. The constraints are guaranteed to be satisfied with a specified probability or confidence level-α- using the known probability density/cumulative distribution of the uncertain variables.

Chance-constrained programming is applicable to models where (optimal) decisions have to be made prior to realizing random effects Birge (1997).

*Corresponding author Email address: p.fattahi@alzahra.ac.ir

As mentioned above, mixed-model assembly line production system is one of the requirements of today's competitive market. In addition, since many researchers have studied it in this field, numerous studiesexisthere. In this section, we will review a part of this rich literature. Monden(Monden, 1983) defined two goals for the sequencing problems: (1) leveling the load on each station on the assembly line and (2) keeping a constant rate of part usage. Goal chasings I and II (GC-I and GC-II) were the considered approaches developed by Toyota Corporation. Miltenburg (1989) considered the second above mentioned goal and formulated the mixed-model sequencing problem as a nonlinear integer programming. Then, because of time complexity function of the proposed problem, he applied two heuristic procedures to solve it. Miltenburg et al. Miltenburg, et al. (1990), in another research, solved the same problem with a dynamic programming algorithm. Sarker and Pan Sarker & Pan (1998),in their research, considered a mixed-model assembly line with either closed or open stations with the goal of minimizing the costs of idle and utility times. Then, to solve the problem, they developed two models for both closed-station and open-station systems to determine line parameters optimally. Results of their research indicate that the minimum total cost of utility and idle times in an open-station system is less than that in a closed-station system for a given line length. Xiaobo and Ohno (1997), for minimizing the conveyor stoppage and finding an optimal or sub-optimal sequence of mixed models, proposed two algorithms. One of them is branch and bound methods used to find an optimal solution to small-sized problems and another is simulated annealing algorithm used to obtain a good sub-optimal solution to large-scale problems. Also, with numerical example, they show that the simulated annealing algorithm is about 100 times faster than the branch and bound algorithm to find an optimal solution. Hyun et al. (1998) proposed a new genetic algorithm to solve multiple objective sequencing problems in mixed model assembly lines. They, between a rich set of criteria based on which to judge the sequences of product models in terms of line utilization, considered this tree objective function: minimizing total utility work, keeping a constant rate of part usage, and minimizing total setup cost. Korkmazel and Meral , (2001) focused on bi-criteria sequencing methods for the mixed-model assembly line in JIT production systems. They considered two major goals: 1) smoothing the workload on each workstation in the assembly line; 2) keeping a constant rate of usage of all parts used in the assembly line. In this research, at first, the sum of deviations of actual production from the desired amount is minimized by using some well-known solution methods for goal 2, and then the best approach is extended for both goals, simultaneously. Kim and Jeong , (2007) ponderedoverhow to optimize the input sequence of product models with sequence-dependent setup time in Mixed-Model Assembly Line using the conveyor system that minimizes the unfinished works within stations. They presented a generalized formulation of the product sequencing-problem in MMAL and suggested a Branch & Bound algorithm for finding the optimal sequence and a heuristic algorithm for solving large-scale problems. Also, in order to find an initial solution for branch and bound (B&B) and heuristic procedure, they proposed a minimum setup time procedure.

Bard et al. Bard et al. (1994) considered mixed-model assembly line with these two objective functions: minimizing the overall line length and keeping a constant rate of part usage. Then, for solving the problem, they used weighted sum approach and a proposed tabu search (TS) algorithm. Mansouri (2005), in his research, presented a multi-objective genetic algorithm (MOGA) approach to a JIT sequencing problem where variation of production rates and the number of setups were simultaneously optimized. Since, these two objectives are typically inversely correlated with each other and optimizing these goals simultaneously is challenging, this type of problem is NP-hard. Then, in order to search for locally Pareto-optimal or locally non-dominated frontier, the MOGA approach was used. Tavakkoli-Moghaddam and Rahimi-Vahed (2006) considered the multi-criteria sequencing mixed-model assembly line problem with three objectives: 1) minimizing total utility work;2) total production rate variation; 3) total setup cost. At first, these three objectives were weighted by their relative importance weights, and then they were presented with new mathematical formulations for these objectives; a memetic algorithm (MA) wasproposed to determine suitable sequences. Rahimi-Vahed and Mirzaei (2007), in their research, considered three objective functions presented in(Tavakkoli-Moghaddam & Rahimi-Vahed, 2006). Due to the complexity of the problem, they applied a hybrid multi-objective algorithm based on shuffled frog-leaping algorithm (SFLA) and bacteria optimization (BO). Fattahi and Salehi (2009) considered sequencing problem to minimize total utility and idle cost with a variable launching interval between products on the assembly line that involves two optimization problems (the sequencing problem and launching interval problem). Since this problem is NP-hard, they proposed a hybrid meta-heuristic algorithm based on the simulated annealing approach and a heuristic approach.

Another important classification of assembly lines is considering the deterministic or nondeterministic assumption of this paper dealing with nondeterministic operation time. If assembly operations are performed using machines/robots that are more advanced or highly qualified and motivated operators, then the operations may have almost constant operation times. In real-life application, especially in manual assembly lines, because of some inevitable human mistakes, breakdown of machines, lack of motivation in workers and the things alike, events are not deterministic, so we consider that operation time is stochastic. To tackle uncertainty of the problem, we chose

Chance-Constrained approach that was firstly introduced by Charnes and Cooper (1959).

Table 1
Notation used in the model

Indices	
i	product, $i \in \{1,2,\dots,I\}$
j	station, $j \in \{1,2,\dots,J\}$
m	model, $m \in \{1,2,\dots,M\}$
Input parameters	
M	Number of models
I	Total number of products to be sequenced ($I=\sum_{m=1}^{M} d_m$)
J	Number of stations
d_m	Demand of model m in minimal part set
t_{mj}	Complation time for model m at jth station
μ_{mj}	Mean value of t_{mj} for model m at jth station
σ_{mj}^2	Variance of t_{mj} for model m at jth station
V_c	Coveyor speed Launch interval of products to the assembly line
γ	Launch interval of products to the assembly line
L_j	The line length of station j
C_U	Cost of utility worker per unit time at jth station
C_{ID}	Cost of idle worker per unit time at jth station
υ_{im}	Sequence dependent production rate variation cost fo position i of the sequence
Decision variables	
x_{im}	Binary variable (if model m is in ith product $x_{im} = 1$, otherwise $x_{im} = 0$
Z_{ij}	Starting position of the task on ith position in station
U_{ij}	Utility timeforith product instation j
ID_{ij}	Idle time for ith product in station j

In the present paper, three objectives are considered simultaneously: 1) total utility work cost;2) total idle cost; 3) total production variation cost. The rest of this paper is organized as follows: in section 2, the problem description and formulation under deterministic assumption is proposed. Section 3 presents the stochastic model. Section 4 is about MOOPs. The proposed MOHS algorithm is presented in section 5. Section 6 includes comparison metrics and numerical experiments; eventually, the last section presents the conclusion and main results.

2. Problem Description and Formulation under Deterministic Assumption

A mixed-model assembly line usually consists of a number of stations linked by a conveyor belt moving at a constant rate (Fattahi & Salehi, 2009). It is assumed that the conveyor moves from left to the right side of the station with a constant speed. In this research, we assumed that all the stations are closed type. Closed type stations haveboundaries, and the workers are not allowed to cross the station's boundaries. Because of this assumption, each station's starting point is zero and the finished point is equal to L_j. The tasks, which are allocated to each station, are properly balanced, and their operation times are stochastic with Normal distribution. Products are launched onto the conveyor belt with a fixed rate and also the worker's moving time is ignored. While performing his/her work, the operator moves downstream on the conveyor and completes the next product by moving upstream. The speed of operators is also equal to conveyor speed.

The design of an MMAL involves several issues such as determining operator schedules, product mix, and launch intervals. First, we shall consider the operator schedules. Two types of operator schedules, early start schedule and late start schedule, are found in the literature Bard et al.(1992). An early start schedule is more common in practice and, thus, is used in this paper. While this schedule allows worker's idle time, it can shorten the line length. Second, minimum part set (MPS) production, a strategy widely accepted in mixed model assembly lines(J. F. Bard et al., 1992), is used. (MPS is a vector representing a product mix, such that$(d_1,\dots,d_M) = (\frac{D_1}{h},\dots,\frac{D_M}{h})$, where M is the total number of modelsD_M is the number of products of model type m which needs to be assembled during an entire planning horizon, and h is the greatest common divisor of$D_1, D_2, \dots, D_M$. This strategy operates in a cyclical manner. The number of products produced in one cycle is $I=\sum_{m=1}^{M} d_m$. Obviously, h time's repetition of producing the MPS products can meet the total demand in the planning horizon Bard et al. (1992). (Fig 1).

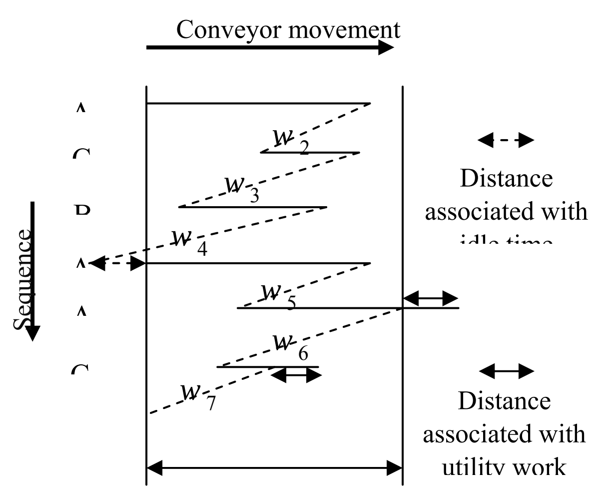

Fig. 1 . Operations in a closed workstation

Before introducing the objectives and model, notations of the problem are described in Table (1).

2.1. Minimizing total utility work cost

In this paper, we assumed that stoppage of conveyor belt is not allowed, so when the regular worker could not finish his/her work in the work zone, the utility workers assist them to finishing the work. In other words, this objective function tries to present a sequence of products to reduce the overload cost of the work. The following proposed model is an extension of the model presented in(Hyun et al., 1998).

$$\min \sum_{J=1}^{J} C_U \left(\sum_{i=1}^{J} U_{ij} + Z_{(I+1)j}/v_c\right)$$

St:

$$\sum_{m=1}^{M} x_{im} = 1 \quad for\ i = 1,..,I \quad (1)$$

$$\sum_{i=1}^{I} x_{im} = d_m \quad for\ m = 1,..,M \quad (2)$$

$$z_{1,j+1} = \sum_{l=1}^{j} L_l \quad for\ j = 1,...,J-1 \quad (3)$$

$$z_{i+1,j} = z_{ij} + v_c.(\sum_{m=1}^{M} x_{im}.t_{mj} - U_{ij} - \gamma + \cdot + ID_{i+1,j})\ for\ i = 1,..,I-1 \quad for\ j = 1,..,J \quad (4)$$

$$U_{ij} \geq (z_{ij} + v_c.\sum_{m=1}^{M} x_{im}.t_{mj} - \sum_{l=1}^{j} L_l)/v_c \quad for\ i = 1,..,I-1 \quad for\ j = 1,..,J \quad (5)$$

$$U_{Ij} \geq [z_{Ij} + v_c.\sum_{m=1}^{M} x_{Im}.t_{mj} - (\sum_{l=1}^{j-1} L_l + v_c.\gamma)]/v_c \quad for\ j = 1,..,J \quad (6)$$

$$x_{im} \in \{0,1\}; \quad (7)$$

$$U_{ij} \geq 0, z_{1j} = 0, z_{ij} \geq 0; \quad (8)$$

$$i = 1,..,I \quad j = 1,..,J \quad m = 1,..,M;$$

The above-mentioned objective function shows the total cost of unfinished work i at station j; the second term in the objective function is in order to account for the utility work that may be required at the end of a cycle. Constraint (1) is used to ensure that exactly one product is assigned to each position in a sequence. Eq. (2) is introduced to show that demand for each model is satisfied in a cycle. Eq. (3) indicates that the operation for the first product of every cycle must be started at the left boundary of the station. In Eq. (4), the starting position of the worker at each station j for product (i+1) in a sequence is indicated. Eq. (5) indicates time of utility work for it product at station j in a sequence. Constraint (6) indicates the utility work time for end product I at this station in a sequence and Z_{1j} shows that stations are closed type.

2.2. Minimizing total idle cost

The aim of this objective function is calculating idle times that happened for each operator at each station; therefore, it tries to minimize its cost per unit time.

$$\min \quad \sum_{j=1}^{J}\sum_{i=1}^{I}(C_{ID}.ID_{ij})$$

St:

$$ID_{ij} \geq (\sum_{l=1}^{j-1} L_l - (z_{i-1,j} + v_c \sum_{m=1}^{M} x_{i-1,m}.t_{mj} - \cdot v_c.U_{i-1.j} - v_c.\gamma))/v_c \quad (9)$$

$$ID_{ij} \geq 0;\ for\ i = 2,..,I-1 \quad for\ j = 1,..,J$$

$And\ Eqs.(1)\_(8)$

This objective function shows sequence-independent idle time of workers, so it is ignored when no product enters his/her station. Eq. (9) indicates idle time for product i at station j.

2.3. Minimizing total production variation cost

A theoretical basis of this model was presented by Miltenburg Miltenburg (1989), and modified by Tavakkoli-Moghaddam & Rahimi-Vahed (2006) to the figure presented here. According to this objective function, when the demand rate of parts is constant over time, the objective is significant to achieve a successful operation of the system. Thus, the objective can be achieved by matching the demand with the actual production Tavakkoli-Moghaddam & Rahimi-Vahed (2006).

$$\min \quad \sum_{i=1}^{I}\sum_{m=1}^{M}\left(\upsilon_{im}\left|\sum_{l=1}^{i}\frac{x_{lm}}{i} - \frac{d_m}{I}\right|\right)$$

St:

$Eqs.(1),(2) and\ (7)$

3. Stochastic Formulation of the Proposed Model

According to assumptions in this paper t_{mj} is a stochastic parameter assumed to be independent and normally distributed with known means μ_{mj} and variances σ_{mj}^2, as $(t_{mj} - N(\mu_{mj}, \sigma_{mj}^2))$, with the confidence level of $\alpha(0 \leq \alpha \leq 1)$. So, $Z_{1-\alpha}$ is α-quantile of the cumulative standardized normal distribution function. It should be mentioned that the value of $\alpha$is determined by the decision-maker. It should also be noted that chance constraint with α=1is equivalent to a deterministic constraint. Following our discussion, constraints (4)-(6) and (9) can be extended to probabilistic model components as follows:

$$z_{i+1,j} = z_{ij} + v_c.(\sum_{m=1}^{M} x_{im}.\mu_{mj} + Z_{1-\alpha}\sqrt{\sum_{m=1}^{M}\sigma_{mj}^2 x_{im}^2} - U_{ij} .-\gamma + ID_{i+1,j}) \quad for\ i = 1,..,I-1 \quad for\ j = 1,..,J \quad (10)$$

$$U_{ij} \geq \frac{\left(z_{ij} + v_c.(\sum_{m=1}^{M} x_{im}.\mu_{mj} + Z_{1-\alpha}\sqrt{\sum_{m=1}^{M}\sigma_{mj}^2 x_{Im}^2}) - \sum_{l=1}^{j} L_l \right)}{v_c}$$

$$for\ i = 1,..,I-1 \qquad for\ j = 1,..,J \qquad (11)$$

$$U_{Ij} \geq \left(z_{Ij} + v_c.(\sum_{m=1}^{M} x_{Im}.\mu_{mj} + Z_{1-\alpha}\sqrt{\sum_{m=1}^{M}\sigma_{mj}^2 x_{Im}^2}) - .(\sum_{l=1}^{j-1} L_l + v_c.\gamma) \right) / v_c \qquad for\ j = 1,..,J \qquad (12)$$

$$ID_{ij} \geq \left(\sum_{l=1}^{j-1} L_l - \left(z_{i-1,j} + v_c.(\sum_{m=1}^{M} x_{i-1,m}.\mu_{mj} + .Z_{1-\alpha}\sqrt{\sum_{m=1}^{M}\sigma_{mj}^2 x_{i-1,m}^2}) - v_c.U_{i-1.j} - v_c.\gamma \right) \right) / v_c$$

$$for\ i = 2,..,I-1 \qquad for\ j = 1,..,J \qquad (13)$$

As Hyun et al.(Hyun et al., 1998) claimed, finding production sequences with desirable levels of all objectives is NP-hard. Computation of total number of sequences for a MMAL problem is as follows:

$$Total\ sequences = \frac{\left(\sum_{m=1}^{M} d_m\right)!}{\prod_{m=1}^{M}(d_m!)} \qquad (14)$$

As can be seen, by increasing the size of the problem, the number of the feasible solutions increases exponentially. Thus, solving these types of problems for the optimal solution within reasonable time is not usually possible. Therefore, to solve these problems, the meta-heuristic approach is proposed.

4. Multi-Objective Optimization Problems

Most of the optimization problems in the real world involve simultaneous optimization of several conflicting objectives, which are called Multi-Objective optimization problems (MOOP). Obtaining an optimal solution in MOOP is not like single objective optimization problems, because a single objective optimization problem will be terminated upon obtaining an optimal solution, while finding a single solution is always difficult for a MOOP.Therefore, in MOOP, it is common to find a set of solutions depending on non-dominance criterion. Let us consider MOOPs as follows, which consists of multiple conflicting objectives to be optimized simultaneously and the various equality and inequality constraints.

$$optimize : F(x) = \{f_1(x), f_2(x), \dots, f_J(x)\}$$

$$s.t : g_k(x) \quad \begin{bmatrix} \leq \\ \geq \\ = \end{bmatrix} 0 \quad ; \quad k = 1,2,\dots,m$$

$$x \in E^n$$

In above formulation, x is decision vector, J is the number of objective functions, and k is the number of equality and inequality constraints.

In a minimization multi-objective problem, assume we have two solutions, x_1 and x_2. Therefore, solution x_1 is said to dominate solution x_2 if and only if:

1. $f_i(x_1) \leq f_i(x_2) \quad \forall i \in \{1,2,\dots,J\}$
2. $f_i(x_1) < f_i(x_2) \quad \exists i \in \{1,2,\dots,J\}$

Solutions which dominate the others, but do not dominate themselves, are called non-dominated solutions Rahimi-Vahed & Mirzaei (2007). On the other hand, the solutions that are non-dominated within the entire search space are called as Pareto optimal solutions.

Traditionally, MOOPs were solved by weighted sum approach, ε-constraint approach, and goal attainment method. The weighted sum approach converts MOOP to a single objective optimization problem by giving suitable weights to the objectives; Bard et al. (1994) had used this method. The ε-constraint method optimizes one of the preferred objective functions using the other objective functions as constraints Mavrotas (2009) explained moreon this method. The above-mentioned approaches need multiple runs to obtain a Pareto optimal solution and require much computational time resulting in a weakly non-dominated solution.

Recently, multi-objective evolutionary algorithms have been used to solve MMALs problems Mansouri (2005); Rahimi-Vahed & Mirzaei (2007). In comparison with traditional methods for solving multi-objective problems, it is proved that the evolutionary algorithms have better performance because of their ability to obtain a Pareto optimal solution just in a single run. Since evolutionary algorithms use a population of solutions, they can be easily extended to maintain a diverse set of solutions in a single run Sivasubramani & Swarup (2011).

5. The Proposed Multi-objective Harmony Search Algorithm

5.1. Harmony search algorithm

Harmony Search (hereafter HS) algorithm is a relatively new population-based meta-heuristic algorithm introduced by Geem et al. (2001). HS, like most of the meta-heuristic algorithms, is a nature-inspired algorithm, which mimics the improvisation of music players. The harmony in music is analogous to the optimization solution vector, and the musician's improvisations are analogous to the local and global search schemes in optimization techniques Sivasubramani & Swarup (2011). By this assumption of analogy between improvisation and optimization, we can have these considerations: fantastic harmony in music is considered as a global optimum in optimization problem, and so, aesthetic standard in music is determined by the objective function in optimization problem; also, pitches of instruments are desired values of the variables, and each practice is the same in each iteration. It is remarkable that the HS algorithm uses a stochastic random search, instead of a gradient search, and is simple in concept, few in parameters, and easy in implementation.

The HS optimization algorithm has been applied successfully to various engineering optimization problems such as satellite heat pipe design Geem & Hwangbo (2006), vehicle routing . Geem et al. (2005) application to pipe network design Geem et al. (2002) and water network design Geem (2006). Mahdavi et al. (2007) in their research, described an improved harmony search (IHS) algorithm for solving optimization problems which employs a novel method for generating new solution vectors to enhance accuracy and convergence rate of harmony search (HS) algorithm. The optimization procedure of the HS algorithm is as follows Sivasubramani & Swarup (2011):

1. Initialize the optimization problem and algorithm parameters.
2. Initialize the harmony memory.
3. Improvise a new Harmony memory.
4. Update the harmony memory.
5. Check for stopping criteria. Otherwise, repeat steps 3 to

5.2. Initialize the optimization problem and algorithm parameters

In the first step, let us consider the optimization problem as follows:

$$\min imize \quad f(x)$$
$$s.t: x_i \in X_i = 1,2,...,N$$

where $f(x)$is an objective function;x is the set of each decision variableX_i; N is the number of decision variables,X_i is the set of the possible range of values for each decision variable, that is, $x_i^L \leq X_i \leq x_i^U$; $x_i^L$and$x_i^U$are the lower and upper bounds for each decision variable. The HS algorithm parameters are also specified in this step. These are the harmony memory size (HMS), or the number of solution vectors in the harmony memory; harmony memory considering rate (HMCR); pitch adjusting rate (PAR); the number of improvisations (NI), or stopping criterion Mahdavi et al. (2007).

HMCR and PAR are used as parameters which improve the solution vector.

$$\left[\begin{array}{cccc|ccc} x_1^1 & x_2^1 & \dots & x_n^1 & f_1(x^1) & f_2(x^1) & f_3(x^1) \\ x_1^2 & x_2^2 & \dots & x_n^2 & f_1(x^2) & f_2(x^2) & f_3(x^2) \\ . & . & . & . & . & . & . \\ x_1^{HMS} & x_2^{HMS} & \dots & x_n^{HMS} & f_1(x^{HMS}) & f_2(x^{HMS}) & f_3(x^{HMS}) \end{array}\right]$$

Fig 2. Harmony memory matrix

5.3. Initialize the optimization problem and algorithm parameters

The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. The HM is similar to the number of population in other evolutionary algorithms (Sivasubramani & Swarup, 2011).

In this step, the HM matrix is filled with many randomly generated solution vectors equal to the HMS between its minimum and maximum limits.

5.4. Improvise a new harmony memory

A new Harmony vector, $X' = (x'_1, x'_2, ..., x'_N)$, is generated based on three rules: (1) memory consideration, (2) pitch adjustment, and (3) random selection. Generating a new harmony is called as improvisation Sivasubramani & Swarup (2011).

In the memory consideration, the value of decision variables $x'_1$for the new vector is chosen from$(x_1^{'1} - x_1^{'HMS})$, and also other decision variables are selected in the same manner. The harmony memory considering rate (HMCR) varies between 0 and 1, and it is the rate of choosing one value from the historical values stored in HM, while (1 _ HMCR) is the rate of randomly selecting one value from the possible range of values as follows:

$$x'_i = \begin{cases} x'_i \in (x_i^1, x_i^2, ..., x_i^{HMS}) \rightarrow with\ probability\ of\ \ HMCR \\ x'_i \in X_i \qquad \rightarrow with\ probability\ of\ \ (1\text{-}HMCR) \end{cases} \quad (15)$$

For instance, HMCR with value 0.8 indicates that the HS algorithm will choose the decision variable from historically stored values in the HM with an 80% probability or from the possible range with a 20% probability. Next, every component obtained by the memory consideration is examined to determine whether it should be pitch-adjusted or not. For this operation, the PAR parameter is used, which is the rate of pitch adjustment decision for X_i as follows:

$$x'_i = \begin{cases} Yes \quad with\ probability\ PAR, \quad so \rightarrow x'_i = x'_i \pm rand()*bw, \\ No \quad with\ probability\ (1\text{-}PAR) \rightarrow (\text{the rate of doing nothing}) \end{cases} \quad (16)$$

where bw is an arbitrary distance bandwidth, and rand ( ) is a random number between 0 and 1.

Only after choosing a value from the HM, the Pitch adjusting process is performed. A PAR of 0.25 means that the algorithm will choose a neighboring value with 25% * HMCR probability. The HMCR and PAR parameters introduced in the harmony search help the algorithm find globally and locally improved solutions, respectively Lee & Geem (2005).

In this step, HM consideration, pitch adjustment, or random selection are applied to each variable of the new harmony vector.
PAR and bw in HS algorithm are very important parameters in fine-tuning of optimized solution vectors and can be potentially useful in adjusting convergence rate of algorithm to optimal solution Mahdavi et al. (2007). In traditional HS algorithm, the values of these parameters were fixed. But, based on Mahdavi et al. (2007), PAR and bw change dynamically with generation number as follows:

$$PAR(gn) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \times gn \quad (17)$$

where:
PAR_{min} : minimum pitch adjusting rate
PAR_{max}: maximum pitch adjusting rate
NI : number of solution vector generations
gn : generation number

$$BW(gn) = BW_{\max} \exp\left(\frac{Ln(\frac{BW_{\min}}{BW_{\max}})}{NI} \times gn \right) \quad (18)$$

where:
bw (gn): bandwidth for each generation
bw_{min} : minimum bandwidth
bw_{max} : maximum bandwidth

5.5. Update the harmony memory

In multi-objective harmony search, this step is different from that of the basic HS algorithm. In this paper, in order to extend the HS algorithm, we used the non-dominated sorting and crowding distance method proposed by Deb et al. (2002) to find a Pareto optimal solutions toour multi-objective MMAL problem with competing objectives.

5.6. Check for stopping criteria. Otherwise, repeat step 3 to 4

When the number of improvisations has been met, The HS algorithm is stopped. Otherwise, Steps 3 and 4 are repeated until the termination criterion is satisfied.

The proposed approach to solvingmulti-objective MMAL problem is described in the following steps:

1) Input the system parameters as mentioned in table 1, and upper and lower bounds of our variables include x_{im}, z_{ij}, U_{ij} and ID_{ij}.
2) Choose the harmony memory size HMS, pitch adjusting rate PAR, bandwidth Band the maximum number of improvisation NI.
3) Initialize the harmony memory HM as explained, while all the control variables are randomly generated within their limits in the first generation.
4) In this section, start the improvisation based on three rules mentioned in section 5.4.
5) Evaluate the three objective functions for each solution vector in HM.
6) Improvise the new harmony memory as explained.
7) Perform thenon-dominated sorting and ranking on the combined existing and new harmony memory.
8) Choose the best harmony memory from the combined solution vectors for the next improvisation.
9) Check the stopping criteria, and if it has been reached to the maximum, go to next step. If not, go to step 5.
10) The non-dominated solution vectors in the HM are the problem Pareto optimal solutions.

Based on available work inthe literature, the input parameters are defined as follows:

Table 2
The parameters of the HS algorithm.

HMS	HMCR	PAR	NI
10	0.85	0.3	100
20	0.93	0.7	500
30	0.99	0.9	1000

Then, the best combination of the parameters that can be used in the model of this research is chosen.

6. Comparison Result

6.1. Multi-objective metrics

Most of the problems in multi-objective optimization methods approximate the Pareto-optimal front by a set of non-dominated solutions. Because of the conflicting and incommensurable nature of some of the criteria in multi-objective problems, making decision about how to evaluate the quality of these solutions is so important. Totally, comparing the solutions of two different algorithms is not straightforward. So, based on Behnamian (2009), for the evaluation of algorithms, we use two metrics as follows:
1. MID (mean ideal distance): The closeness between Pareto solution and ideal point (0, 0),
2. SNS: The spread of non-dominance solution.

$$MID = \frac{\sum_i \sqrt{\sum_{j=1}^{m} f_{ij}^2}}{n} \quad (19)$$

wheren is the number of non-dominated set and f_{ij}^2 is objective function j for Pareto solution i. It should be mentioned that the lower value of MID indicates the better solution quality, and vice versa; the higher value of SNS indicates the better solution quality (more diversity in the obtained solution).

$$SNS = \sqrt{\frac{\sum_{i=1}^{n} (MID - C_i)^2}{n-1}} \quad (20)$$

$$C_i = \sqrt{f_{ij}^2} \quad (21)$$

6.2. *Experimental result*

In this section, several numerical examples are given to illustrate the solution methods of the proposed scheduling problems with stochastic processing times. On the other hand, because this problem is NP-hard, a meta-heuristic algorithm is proposed to solve it. MOHS algorithm is proposed in this research. In the following, the performance of the proposed MOHS algorithm is compared with that of the well-known NSGA-II algorithm. These meta-heuristic algorithms have been coded in MATLAB R2013a and executed on an Intel(R) Core™ i5 CPU (2.4 GH) and Windows 8.1 using 2.99 GB RAM.

Based on the assumption in this research, processing times are stochastic parameters that are distributed normally. In the following, it is assumed that the variances of processing times are proportional to their means. In other words, for each model m in station j, $\sigma_{mj}^2 = \lambda\mu_{mj}$ where $\lambda$is a strictly positive constant and σ_{mj}^2 are the mean and variance of the processing time of model m in station j. Considering such a relationship between variances and means for the normal distributions is not uncommon Elyasi & Salmasi (2013). For instance, Sarin et al.(1991), Cai and Zhou (1997),; Elyasi et al. (2013) considered this assumption for the normally distributed processing times. Here, we consider that $\lambda = 0.5$ and μ_{mj} are generated randomly from the uniform distribution on the interval [5, 15]. The confidence level, α, is set to 0.975 (Z_α =1.96).

In the following, we present a brief description about NSGA-II algorithm.

6.3. *NSGA-II algorithm*

Deb et al. (2002) suggested an elitist multi-objective genetic algorithm in which the parent and offspring population (each of the same size N) are combined together and evaluated using: (1) a fast non-dominated sorting method;(2) an elitist approach;(3) an efficient crowded-comparison mechanism.

When more than N population members of the combined population exist in the non-dominated set, only those that are maximally apart from their neighbors according to the crowding distance are chosen(Rahimi-Vahed & Mirzaei, 2007).

6.4. *Small-sized problems*

In this research, the experiments include small- and large-sized problems with the following general assumptions:

(1) The conveyor speed (V_c) is set to 1;
(2) Utility work cost and Idle cost for each station areset to 1;
(3) The number of MPS is set to 1;
(4) All sequence-dependentproduction rate variance costs of each station are set to 1;

In this section, we present six test problems carried out on small-sized problems. These test problems are generated based on (Tavakkoli-Moghaddam & Rahimi-Vahed, 2006). We consider that there are three stations and three types of products as mentioned in Table 3. The six MPSs shown in Table 4 are tested. The number of feasible solutions in Table 4 is computed from Eq. (14) and also the last column in this table refers to the launch interval time for each problem instance. Moreover, the introduced test problems are solved with MOHS and NSGA-II, and their comparison results are shown in Tables 5. CPU time in tables shows the time required for solving these problems. Time is considered because of its important role in sequencing problems.

Table 3
Assembly time and workstations length

Workstation	Model 1	Model 2	Model 3	Workstation length
1	N(6.3,3.15)	N(5.9,2.95)	N(14.6,7.3)	2
2	N(14.1,7.05)	N(7.8,3.9)	N(6.8,3.4)	4
3	N(11.3,5.65)	N(10.5,5.25)	N(14.7,7.35)	2

Table 4
Info of test problem's

Problem	I	MPS	No. of feasible solution	Lunch interval(T)
1	6	(2,2,2)	90	6.6
2	6	(1,2,3)	60	6.5
3	6	(1,1,4)	30	6.6
4	8	(3,2,3)	560	6.2
5	8	(4,2,2)	420	6.2
6	8	(2,3,3)	560	6.2

Table 5
Comparison results of two algorithms for small size problems.

Algorithm	MOHS				NSGA-II			
Problem	Number of	Run	MID	SNS	Number of pare to	Run	MID	SNS
1	2	20.7s	276.097	0.002	2	4.2	276.49	0.000
2	2	21.6	286.35	0.003	2	7.1	286.75	0.000
3	1	19.5	297	___	**2**	4.5	297.43	0.000
4	**4**	20.5	367.48	0.007	2	10.9	367.98	0.000
5	**3**	19.7	355.74	___	2	10.1	356.23	0.000
6	2	20.5	370.03	0.003	2	10	370.5	0.001
Average			**325.4495**	**0.004**			325.8967	0.0003

*Bold values refer to the best values.

6. 5. Large-sized problems

In this section, large-sized problems are tested. Based on the above-mentioned literature, this problem for large sizes is NP-hard. So, it should be solved with meta-heuristic algorithms. In this research, we used MOHS algorithm. Large-sized problems, as mentioned above, were solved based on (Tavakkoli-Moghaddam & Rahimi-Vahed, 2006). The number of stations in such problems is fixed to 10 and length of each station (L_j) is generated from uniform distributions of U (2, 5). Based on (Manavizadeh et al., 2013), the number of product types isset to $(0.25\times I)+1$. Mean and Var for processing time for each product at each station are generated based on the idea mentioned in section 6.1. Six MPSs shown in Table 6 are provided and the proposed meta-heuristic algorithm is applied to each of them. For large-sized problems, idle cost and production rate variation cost are generated based on uniform distribution between 5 and 10, and utility worker cost is generated based on uniform distribution between 6 and 9. Table 7 illustrates the comparison results of the two algorithms for large-sized problems.

6.6. Parameter settings

Based on literature and some extensive experiments, different sets of parameters for the proposed algorithm were tested and, finally, the following sets were found:
Harmony memory size HMS = 20,
Harmony memory considering rate HMCR = 0.85,
Pitch adjusting rate PAR_{min} = 0.2 and PAR_{max} = 2,
Bandwidth BW_{min} = 0.45 and BW_{max} = 0.9,
The number of improvisations NI = 100.
For comparison purpose, this multi-objective problem has also been solved by NSGA-II and, at last, our proposed algorithm is compared with NSGA -II. For NSGA -II, based on literature and some experiments, the number of population NP = 20 and the number of generation Gen = 1000 were considered.
Fig. 3 shows Pareto optimal solutions toboth the proposed method and NSGA-II algorithm.

Table 6
Info of test problems

Problem	I	MPS	No. of feasible solution	Lunch interval(T)
1	20	(4,6,3,2,2,3)	$9.78*10^{12}$	10
2	20	(4,4,2,3,3,4)	$2.44*10^{12}$	10
3	20	(5,5,4,3,2,1)	$5.87*10^{12}$	10
4	30	(4,4,3,5,5,3,2,4)	$18.51*10^{24}$	10
5	30	(5,4,4,4,3,3,2,5)	$18.51*10^{24}$	10
6	30	(3,5,5,2,4,3,4,4)	$18.51*10^{24}$	10

Table 7
Comparison results of two algorithms for large-sized problems.

Algorithm	MOHS				NSGA-II			
Problem	Number of pare to solution	Run time	MID	SNS	Number of pare to solution	Run time	MID	SNS
1	**10**	27.6s	29401.65	16.1	7	24	29390.67	0.001
2	**9**	26.8	29300.25	11.3	6	24	29295.63	0.001
3	**7**	26.6	29067.14	1.3	6	24	29066.73	0.001
4	**6**	30.5	44720.86	15.4	5	30	44714.64	1.5
5	**10**	30.9	44848.72	1.7	8	29	44856.73	15.8
6	6	30.8	44859.18	17.8	**8**	30	45171.25	0.001
Average			**37032.97**	**10.6**			37082.61	2.9

*Bold values refer to the best values.

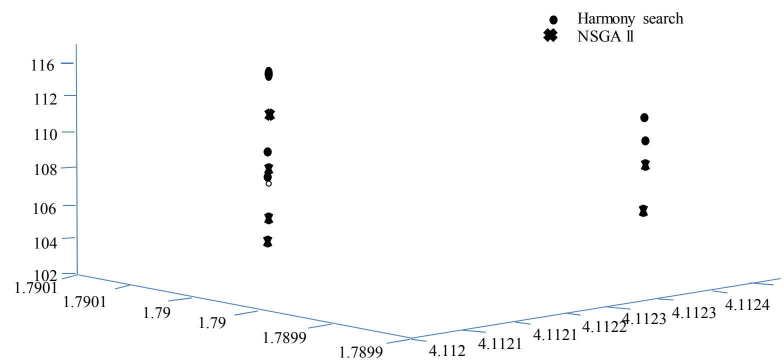

Fig. 3. Pareto optimal solutions for MOHS & NSGA-II

6.7. Comparison result

In order to validate the proposed algorithm, we used six test problems in small sizes and 6 test problems in large sizes and evaluated the performance and reliability of the proposed algorithm in comparison with NSGA-II of a single run. Two compression metrics (MID & SNS) were used to validate the efficiency of the algorithms. As it can be seen in average column, MOHS has better performance.

7. Result

This paper focuses on mixed model assembly line sequencing problems with stochastic processing times with three objective functions: 1) total utility work cost, 2) total idle cost, and 3) total production rate variation cost that are simultaneously considered. At first, Chance-constrained optimization is used to model the problem, then, because of NP-hardness nature of problem, the Multi-Objective Harmony Search (MOHS) algorithm is proposed to solve it.
Eventually, to validate the proposed algorithm, we used 12 test problems in small and large sizes and evaluated the performance and the reliability of the proposed algorithm in comparison with NSGA-II. Based on the results in table (5) for small size and table (7) for large size, two compression metrics (MID & SNS) were used to validate the efficiency of the algorithms, and accordingly, MOHS had better result. On the other hand, in small sizse, as shown in table (5), run times for NSGA-II are so better than MOHS, but for large size, table (7) shows that two algorithms have almost the same performance. Another comparison metric is Number of Pareto solution that MOHS in most of the problems had better performance. Totally, the obtained results acknowledge the better performance of the proposed algorithm in comparison with NSGA-II. In the future studies, this problem could be solved with other meta-heuristics and hybrid algorithms , and it would be a good suggestion to use some other methods to deal with uncertainty situation in a problem like fuzzy or scenario-based methods.

References

Bard, J., Shtub, A., & Joshi, S. (1994). Sequencing mixed-model assembly lines to level parts usage and minimize line length. *The International Journal of Production Research, 32*(10), 2431-2454.

Bard, J. F., Dar-Elj, E., & Shtub, A. (1992). An analytic framework for sequencing mixed model assembly lines. *The International Journal of Production Research, 30*(1), 35-48.

Behnamian, J., Ghomi, S. F., & Zandieh, M. (2009). A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications, 36*(8), 11057-11069.

Birge, J. R. (1997). Introduction to stochastic programming. *INFORMS J. Comput., 9*(2), 111-133.

Cai, X., & Zhou, S. (1997). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics (NRL), 44*(6), 531-557.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation, 6*(2), 182-197.

Elyasi, A., & Salmasi, N. (2013). Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming. *Mathematical and Computer Modelling, 57*(5), 1154-1164.

Fattahi, P., & Salehi, M. (2009). Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval. *The International Journal of Advanced Manufacturing Technology, 45*(9-10), 987-998.

Geem, Z., Kim, J. H., & Loganathan, G. (2002). Harmony search optimization: application to pipe network design. *International Journal of Modelling and Simulation, 22*(2), 125-133.

Geem, Z. W. (2006). Optimal cost design of water distribution networks using harmony search. *Engineering Optimization, 38*(03), 259-277.

Geem, Z. W., & Hwangbo, H. (2006). Application of harmony search to multi-objective optimization for satellite heat pipe design. Paper presented at the Proceedings of.

Geem, Z. W., Kim, J. H., & Loganathan, G. (2001). A new heuristic optimization algorithm: harmony search. *Simulation, 76*(2), 60-68.

Geem, Z. W., Lee, K. S., & Park, Y. (2005). Application of harmony search to vehicle routing. *American Journal of Applied Sciences, 2*(12), 1552-1557.

Hyun, C. J., Kim, Y., & Kim, Y. K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers & Operations Research, 25*(7), 675-690.

Kim, S., & Jeong, B. (2007). Product sequencing problem in Mixed-Model Assembly Line to minimize unfinished works. *Computers & industrial engineering, 53*(2), 206-214.

Korkmazel, T., & Meral, S. (2001). Bicriteria sequencing methods for the mixed-model assembly line in just-in-time production systems. *European Journal of Operational Research, 131*(1), 188-207.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer methods in applied mechanics and engineering, 194*(36), 3902-3933.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation, 188*(2), 1567-1579.

Manavizadeh, N., Tavakoli, L., Rabbani, M., & Jolai, F. (2013). A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority. *Journal of Manufacturing Systems, 32*(1), 124-137.

Mansouri, S. A. (2005). A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research, 167*(3), 696-716.

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation, 213*(2), 455-465.

Miltenburg, J. (1989). Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science, 35*(2), 192-207.

Miltenburg, J., Steiner, G., & Yeomans, S. (1990). A dynamic programming algorithm for scheduling mixed-model,just-in-time production systems. *Mathematical and Computer Modelling, 13*(3), 57-66.

Monden, Y. (1983). Toyota production system: practical approach to production management: Engineering & Management Press.

Rahimi-Vahed, A., & Mirzaei, A. H. (2007). A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. *Computers & industrial engineering, 53*(4), 642-666. doi:10.1080/03052150701512042

Sarin, S. C., Erel, E., & Steiner, G. (1991). Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research, 51*(2), 188-198.

Sarker, B. R., & Pan, H. (1998). Designing a mixed-model assembly line to minimize the costs of idle and utility times. *Computers & industrial engineering, 34*(3), 609-628.

Sivasubramani, S., & Swarup, K. (2011). Multi-objective harmony search algorithm for optimal power flow problem. *International Journal of Electrical Power & Energy Systems, 33*(3), 745-752.

Tavakkoli-Moghaddam, R., & Rahimi-Vahed, A. (2006). Multi-criteria sequencing problem for a mixed-model assembly line in a JIT production system. *Applied mathematics and computation, 181*(2), 1471-1481.

Xiaobo, Z., & Ohno, K. (1997). Algorithms for sequencing mixed models on an assembly line in a JIT production system. *Computers & industrial engineering, 32*(1), 47-56.

Zamami Amlashi, Z., & Zandieh, M. (2011). Sequencing Mixed Model Assembly Line Problem to Minimize Line Stoppages Cost by a Modified Simulated Annealing Algorithm Based on Cloud Theory. *Journal of optimization in Industrial Engineering*, (8), 9-18.

This article can be cited:Fattahi, P. & Askari A(2018). A Multi-objective Mixed-Mmodel Assembly Line Sequencing Problem With Sstochastic Operation Time.*Journal of Optimization in Industrial Engineering.11 (1), 157-167*

URL:http://www.qjie.ir/article_535416.html
DOI: 10.22094/joie.2017.568.62