Sequencing of mixed models on U-shaped assembly lines by considering effective help policies in make-to-order environment

M. Rabbani[a,*], N. Manavizadeh[b] and N. Shabanpour[a]

a. *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, P.O. Box 11155-4563, Iran.*
b. *Department of Industrial Engineering, Khatam University, Tehran, Iran.*

Abstract. Mixed-Model Assembly Line (MMAL) is a type of production line where a variety of products' models similar to products' characteristics are assembled in the same line. Many manufacturers tend to use mixed-model assembly line in their production lines, since this policy makes it possible to assemble various products in the Make-To-Order (MTO) environment. In this research, the sequence of U-type mixed-model assembly line is achieved through considering downstream help and storage of kits as effective help policies for reducing total line stoppages and tardiness in delivery time of products to customers. Since this problem is NP-hard, hybrid GA-Beam search algorithm is developed to solve the problem. Numerical experiments are used to evaluate the performance and effectiveness of the proposed algorithm. To the best of our knowledge, this is the first study that considers getting help from downstream worker or using storehouse of kits, which has ready-to-assemble parts in the conditions that workers cannot complete the remained task in the work horizon.

1. Introduction

In order to satisfy various preferences of customers, firms are forced to produce a large variety of models or basic products to maintain their competitiveness. A Mixed-Model Assembly Line (MMAL) is a type of production line that is capable of producing different small-lot productions due to insignificant setup costs while changing one model of a product to another one. It has several workstations that are connected to each other with a conveyor belt. These assembly lines are greatly executed by industries to manage diversification of customers' demands.

The efficient implementation of MMAL depends on the solution to two problems:

- Balancing the line;
- Determining the production sequence for different models based on the goals of the firms.

  In this research, it is assumed that assembly line has already been balanced.

  Mixed-Model Assembly Line sequencing Problems (MMALP) follow some different objectives that are mostly considered as cost manner:

- To keep a constant usage rate of every part used by the assembly line, see Xiabo et al. [1];
- To keep a constant rate of feeding of every model insert to the line, see Kubiak [2];
- To level the load at each workstation, see Yano and Rachamadugu [3];

*. *Corresponding author. Tel.: +98 21 88021067;*
   Fax: +98 21 88013102
   E-mail addresses: mrabani@ut.ac.ir (M. Rabbani);
   n.manavi@khatam.ac.ir (N. Manavizadeh)

- To minimize the conveyor stoppage, as required by the times of machines and number of products that must be produced, see Xiabo and Ohno [4,5].

Designing appropriate configuration of the line assembly has some effects on decreasing of stoppages and cooperation of workers. In a U-type assembly line, the moving of workers is easier than in the straight-type assembly line and workstations in opposite sides have a chance to get help from workers of both sides in a flexible manner; hence, stoppages of the line decrease in comparison with the traditional straight-line formation. Miltenburg [6] studied 114 Japanese and American real U-lines and reported impressive benefits of U-line implementations, including increased productivity, reduced work in process inventory, shorter throughput, and improved quality. Another benefit of U-type assembly line is its responsibility to changes of demands, because U-type lines are more likely to act in accordance with changes of orders [7].

Besides these objectives, to be more responsive and faster in meeting the demands of customers, producers employ different policies. In this problem, MTO approach is used to provide more flexibility against fluctuations in the orders of customers [7]; hence, due date is an important goal, which is considered in this study.

One of the most important elements for minimizing stoppages of line is employing effective help policies. Help policy is used in line in various ways to help workers of the line in reducing stoppages with due attention to the line configuration, such as objectives of the firms and skills of the workers. An example of help policy is getting help from downstream by the worker who fails to finish his/her assigned job. In the recent study, different help policies are used and compared based upon decreasing of stoppages.

The growing number of product variants causes more part numbers to be considered as another problem. This problem increases line side storage space, part numbers, and difficulties in delivery of parts. It also causes longer walking time of the operator and searching time [7]. One way to handle this problem is storage of kits in assembly line. In this study, storage of kits is used as a kind of help policy to reduce stoppages of line, which contains predetermined parts or subassemblies in specific containers to deliver them to the line.

2. Literature review

Following the first study on mixed-model assembly line conducted by Kilbridge and Wester [8], several researchers studied the mixed-model assembly line sequencing problem. Thomopoulos [9], Dar-El and Cother [10], and Dar-El and Cucuy [11] focused on minimizing line length and proposed an integer program-

ming and heuristic procedure to solve the problem. Okamura and Yamashina [12] offered a heuristic approach for minimizing line stoppages problem. Xiaobo and Ohno [4] minimized the total line stoppages time for a mixed-model assembly line in a JIT production system. They assumed that if a worker could not finish his/her work within the work zone, he/she should stop the line at the downstream boundary of his/her work zone until the remained work was finished. Xiaobo and Ohno [5] in another research minimized the total conveyor stoppage time. They characterized some useful properties of the sequencing problem besides the goal. Then, some lemmas were proofed for upper and lower bounds of problem and the problem was solved with a heuristic algorithm. Celano et al. [13] minimized the total conveyor stoppages time in a U-type manual mixed-model assembly line. They focused on management of human resource to achieve their goal. Yoo et al. [14] used the aid of Relief Man to decrease line stoppages. The relief man (RM) could support all workstations, but could not operate in more than two workstations simultaneously. They developed a hybrid method that employed meta-heuristics like SA (Simulated Annealing) and TS (Tabo Search) in a hierarchical manner and effectiveness of their model and appropriate numbers of RM according to the production rate were clarified under some experiments. Giard and Jeunet [15] provided an operational way to use the utility worker to avoid line stoppages by considering minimization of the number of utility workers. Kim and Jeong [16] studied using extra workers in MMAL to minimize unfinished works. In their research, operators worked in close stations and if they failed to complete their operation, they would get help from extra workers. Also, in this situation, they could not stop the conveyor. Boysen et al. [17] investigated the more realistic sequencing problem by considering real availability of utility workers and presented a binary linear program along with a complexity proof. Cevikan [18] considered walking-worker assembly systems as an effective method to achieve flexibility. Mirzapour et al. [19] used bypass subline to reduce stoppages of assembly line and to level the part usage rates. They proposed some priority rules for entering the parts to the bypass subline or to the mainline in unsynchronized situations. Manavizedeh et al. [20] minimized total times of stoppages and idleness in the straight mixed-model assembly line through various help policies. They categorized help policies into four sections: 1) Central PC (CP), 2) Vicinity Help (VH), 3) Idle Worker Help (IWH), and 4) Utility worker. They used scatter search for their NP-Hard in big-size problem as a metaheuristic algorithm. The results proved that VH policy was more appropriate for small-size problems, but IWH and CP policies were good at big-size problems. Guo et al. [21] solved an MMAL (Mixed-Model Assembly Line) scheduling

problem by minimizing the total Earliness/Tardiness (E/T) penalty costs and maximizing smoothness of the production flow. They used a developed bi-level genetic algorithm to solve their problem. Some papers considered MMALP (Mixed-Model Assembly Line Problem) in MTO environment. Bukchin et al. [22] studied mixed-model assembly line design to maximize the line throughput in MTO environment. Manavizadeh et al. [7] used MTO approach in the JIT environment by the use of kiting policy. They worked on minimizing the tardiness in preparing orders and its cost besides reducing kitting costs, stoppages of the line, and idleness time of labors. They also used relief-man as a soft facility when general workers could not finish their assembly function. Manavizadeh et al. [23] investigated sequencing problem in mixed-model assembly line in MTO environment. They minimized six objective functions: 1) Total utility workers, 2) total utility work costs, 3) total setup costs, 4) total idleness costs, 5) total operator errors, and 6) total tardiness. They proposed a new procedure to choose important orders to enter into the floor shop. By introducing "critical ration of each order" and "significance degree of customer and innovation product", the orders were ranked and entered into the floor shop; then, according to the objectives, the sequence of the products would be defined. Tavakkoli-Moghaddm and Rahimi-Vahed [24] considered three objectives simultaneously: 1) Total utility wastes, 2) total production rate variation cost, 3) total setup cost for MMALP. Fatahi and Salehi [25] minimized idleness costs and utility time costs with variable launching intervals between products on assembly line. Bautista and Cano [26] presented a formulation for the MMSP-W (Mixed-Model Sequencing Problem with Workload minimization) for production lines with serial workstations. Tamura et al. [27] proposed an explicit and complete formulation for the production sequencing problem in the mixed-model line, in which the objective function was to minimize the total line stoppage time. Kucukkoc and Zhang [28] considered simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines, in which a new assembly line system configuration for satisfying customized demands on time with existing resources was proposed. Mosadegh et al. [29] solved simultaneous balancing and sequencing of mixed-model assembly line in order to minimize total utility work. Hamzadayi and Yildiz [30] simultaneously solved balancing and sequencing for an efficient implementation of a mixed-model U-shaped assembly line in a JIT (Just In Time) environment. Dong et al. [31] addressed the line balancing problem and the model sequencing problem jointly to minimize the expectation of work overload time for a given combination of cycle time and number of workstations. Azzi et al. [32] aimed to introduce an innovative se-

quencing model for the mixed-model multi-line system, in the case of closed stations, in which it was possible to obtain, after a first long-term configuration, short-term balancing using an appropriate sequencing for a given production mix and characteristics of the assembly system. Hamzadayi and Yildiz [33] presented simultaneous tackling of the Mixed-Model U-shaped assembly Line (MMUL) balancing/model sequencing problems (MMUL/BS) with parallel workstations and zoning constraints, which allowed the decision maker to control the process to create parallel workstations and to work in different scenarios. Battini et al. [34] illustrated an innovative balancing-sequencing step-by-step procedure that aimed to optimize the assembly line performance and, at the same time, contain the buffer dimensions in the functions of different market demands and production mixes. The literature review presented above provides the main objective functions of mixed-model assembly line sequencing problems and their solution methods. Reviewing the literature of mixed-model assembly line sequencing problems shows that there are some important factors that can be added to the problem to make it more realistic. Many studies have been done on the mixed-model assembly lines. But, in this paper, we specifically focus on reducing the stoppages and tardiness. This is possible through various help policies and alternative launching interval time. In this study, we want to propose some new techniques to reduce line stoppages and tardiness in delivery of products in MTO environment for MMALP (Mixed-Model Assembly Line Problem).

3. Proposed model

In this section, the proposed model is described in details. First, the description of the problem is provided for two scenarios; second, objectives of the model are illustrated; third, assumptions are presented; and fourth, constraints of the model in these two scenarios are clarified.

3.1. Description of the problem

We consider a U-type mixed-model assembly line with several workstations in MTO environment. Workstations are connected to each other with conveyor, which has a constant speed. In addition, storage of kits happens inside the line. Moreover, a central computer controls the start time and end time of the workers' jobs and the number of parts in the storehouse of kits (Figure 1).

When a worker does not have enough time to finish his/her assigned operation, the CC (Central Computer) informs him/her whether to get help from downstream worker or to go to storehouse of the kits and get the needed part to complete the remained task in order to reduce stoppages time. Thus, using
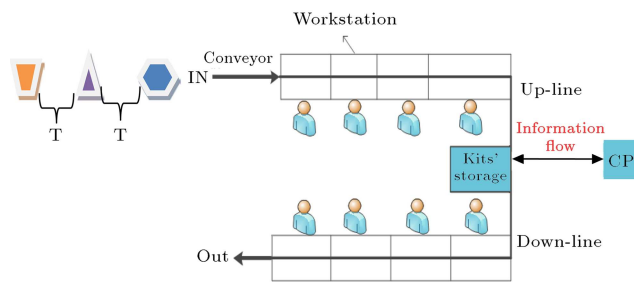
Figure 1. U-type mixed-model assembly line.

the storehouse of kits and getting help from the downstream worker are our help policies. Also, in MTO environment, the production is according to the order of customers; hence, we should reduce tardiness in the production. The launching intervals of products to the line are constant in scenario one and are variable in scenario two and the remained assumptions are the same for both scenarios.

3.2. Objectives

A multi-objective sequencing problem is considered in this model to attain two objectives, which are minimizing total conveyor stoppages time and minimizing the total tardiness in completion of each model.

3.3. Assumptions

1. A U-line consisting of K workstations and N products to be assembled has been considered. The products to be produced belong to different M models;

2. All jobs are ready at time zero;

3. There is no buffer in the original line between stations;

4. A single worker, denoted by j, is allocated for each workstation, j;

5. The workstations are linked through a conveyor belt moving with constant speed equal to v_c;

6. The U-line is organized in two branches, respectively, called up-line and down-line;

7. Workers move within their workstations on the interior of the U-line with a finite speed, $v > v_c$. The same walking speed is assumed for all the workers;

8. When a worker j does not have enough time to finish an operation on a product, i.e. a product passes through the downstream boundary of a generic workstation, j, before all the assigned tasks on that workstation are completed, the conveyor is stopped by the worker j, who completes his/her assigned operation alone or together with another operator in accordance with the help policy in effect;

9. The proposed model is solved in MTO environment;

10. The first workstation is a crossover workstation;

11. Workstations are of closed type;

12. We assume that $k/2$ stations are in the up-line and the rest are in the down-line.
    These assumptions are similar in both scenarios.

3.4. Scenario 1: Applying equal launching intervals

In this scenario, a central computer (cp) controls the start and end times of the workers' jobs. The cp informs the downstream worker who has enough free time; but if there is no downstream worker with enough available time, the cp informs the worker who causes stoppage to use the storehouse of kits. Also, the same launching interval (T) is applied for all products, which is the time between arrivals of two consequent products in the line (Figure 2).

Notations

Sets

i :   Product, $i \in \{1, 2, ..., N\}$

j :   Station, $j \in \{1, 2, ..., K\}$

m :   Model, $m \in \{1, 2, ..., M\}$

Parameters

v :   Speed of worker

v_c :   Speed of conveyor

ld :   Distance between the U-line parallel branches

l :   Workstation length

t_c :   Launching interval between products

t_{mj} :   Working time for model m at station j

d_m :   Demand of model m

du_{im} :   Due date for completing model m in the ith sequence

Decision variables

x_{im} :   1 if product i in a sequence, 0 otherwise

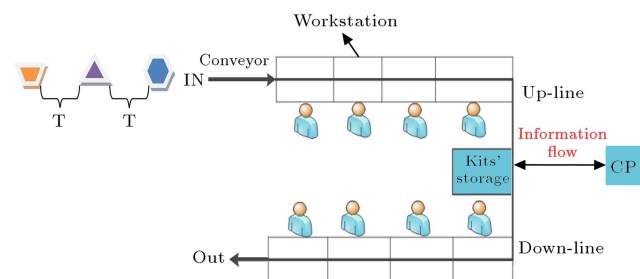z_{ij} :   Position of the ith product at station j

Figure 2. Schema of the first scenario.

stb_{ij} :   Stoppage time related to product i at station j before applying the help policy

sta_{ij} :   Stoppage time related to product i at station j after applying the help policy

hps_{ij} :   Reduced stoppage time related to product i at station j by the use of kitting storage

hpf_{ij} :   Reduced stoppage time related to product i at station j by the help of downstream worker

ta_{im} :   Tardiness in completing for model m in the ith sequence

The objective of sequencing is minimization of the total conveyor stoppage time and total tardiness due to completion of each model:

Objective function:

$$\text{Minimize } z = \sum_{i=1}^{N} \sum_{j=1}^{K} sta_{ij} + \sum_{i=1}^{N} \sum_{m=1}^{M} ta_{im}, \qquad (1)$$

Constraints:

$$\sum_{m=1}^{M} x_{im} = 1 \qquad \forall i, \qquad (2)$$

$$\sum_{i=1}^{N} x_{im} = 1 \qquad \forall m. \qquad (3)$$

Constraint (2) ensures that exactly one product is assigned to each position in a sequence. Constraint (3) guarantees that the demand of each model is satisfied in a cycle by considering one position for each model of a product in a sequence:

$$z_{i+1j} = \max\left\{ 0, \min\left(z_{ij} + v_c * \sum_{m=1}^{M} x_{im} * t_{mj} \right.\right.$$
$$\left.\left. - t_c * v_c, l - t_c * v_c \right) \right\}, \qquad (4)$$

$$z_{1j} = j * l, \qquad (5)$$

$$z_{i1} = 0, \qquad \forall i. \qquad (6)$$

Constraints (4)-(6) acquire position of the ith product at station j to investigate whether stoppage occurs or not. Constraint (4) shows that if stoppage does not occur, position of the product will be in the starting point of the next station. Also, if stoppage occurs, position of the product will be higher than the length of the station. Constraint (5) shows that the position of the first product at the end of an operation in each

station is equal to sum of the lengths of the station through which the product has passed. Constraint (6) depicts that position of a product in the first station is equal to zero:

$$stb_{ij} = \max\left\{ 0, \left(z_{ij} + v_c * \sum_{m=1}^{M} x_{im} * t_{mj} - l \right) \div v \right\}. \qquad (7)$$

Constraint (7) indicates the stoppage time related to product i at station j when the worker has not yet received help from the downstream worker or the storehouse of kits:

$$hpf_{ij} = stb_{ij} - \max\left\{ t_c - \sum_{m=1}^{M} x_{i-1m} * t_{mj+1}, 0 \right\}. \qquad (8)$$

Constraint (8) indicates the reduced stoppage time when the downstream worker has enough time to help the worker that has faced stoppage (Figure 3):

$$hps_{ij} = 2*$$
$$\sqrt{\left(l * \frac{j}{2} - z_{ij} - \sum_{m=1}^{M} x_{im} * t_{mj} \right)^2 + \left(\frac{ld}{4} \right)^{2/v}}. \qquad (9)$$

Constraint (9) calculates the reduced stoppage time when the downstream worker does not have enough time to help the worker who has faced stoppage and the worker uses the storehouse of kits (Figure 4):

$$sta_{ij} = \begin{cases} hpf_{ij}, & \text{If downstream worker has} \\ & \text{enough time to help} \\ hps_{ij}, & \text{Otherwise} \end{cases} \qquad (10)$$
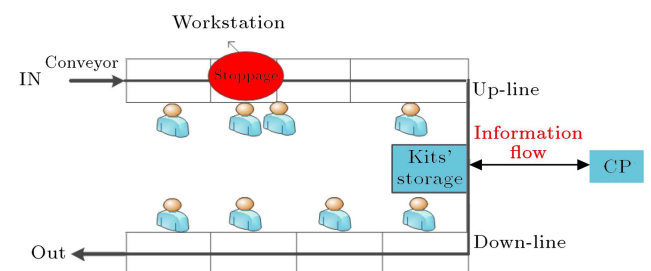

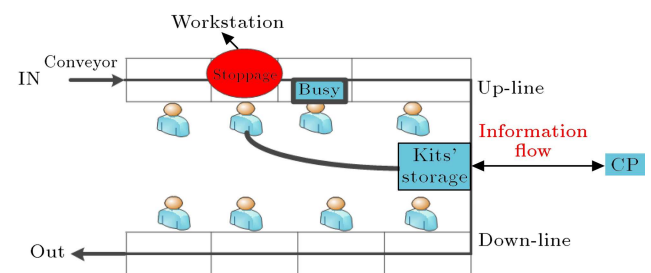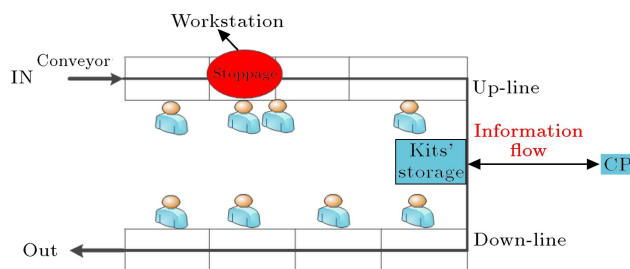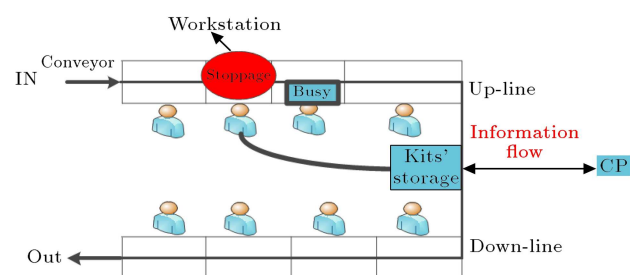**Figure 3.** Downstream help policy in Scenario 1.

Figure 4. Using storehouse of kits in Scenario 1.

Constraint (10) represents the stoppage time after applying the help policy:

$$ta_{im} = \max\left(\left|du_{im} - x_{im} * \sum_{j=1}^{K} t_{mj}\right|, 0\right). \qquad (11)$$

Constraint (11) is considered for MTO environment, which includes tardiness in completing the ith product.

3.5. Scenario 2: Applying a different launching interval

In this scenario policy, the central computer (cp), which controls the start and end times of the workers' jobs, selects the appropriate worker for helping the worker who faces stoppage. The cp informs the downstream worker who has enough available free time; but, if the downstream worker does not have enough available time, the cp informs the worker who causes stoppage to use kit. Also, a different launching interval is considered (Figure 5).

Notations

Sets

$i :$ Product, $i \in \{1, 2, ..., N\}$

$j :$ Station, $j \in \{1, 2, ..., K\}$

$m :$ Model, $m \in \{1, 2, ..., M\}$

Parameters

$v :$ Speed of worker

$v_c :$ Speed of conveyor

$ld :$ Distance between the U-Line parallel branches

$l :$ Workstation length

$a_i :$ Launching interval between products

$t_{mj} :$ Working time for model m at station j

$d_m :$ Demand of model m

$du_{im} :$ Due date for completing model m in the ith sequence
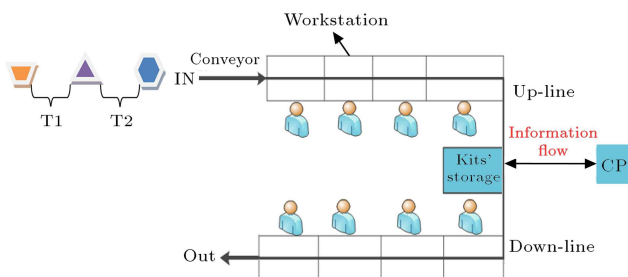
Figure 5. Schema of the second scenario.

Decision variables

$x_{im} :$ 1 if product i in a sequence is model, 0 otherwise

$z_{ij} :$ Position of the ith product at station j

$stb_{ij} :$ Stoppage time related to product i at station j before applying help policy

$sta_{ij} :$ Stoppage time related to product i at station j after applying help policy

$hps_{ij} :$ Reduced stoppage time related to product i at station j by the use of kitting storage

$hpf_{ij} :$ Reduced stoppage time related to product i at station j by the help of downstream worker

$ta_{im} :$ Tardiness in completing for model m in the ith sequence

The objective of sequencing is minimization of the total conveyor stoppage time and total tardiness due to completion of each model:

Objective function

$$\text{Minimize } z = \sum_{i=1}^{N} \sum_{j=1}^{K} sta_{ij} + \sum_{i=1}^{N} \sum_{m=1}^{M} ta_{im}. \qquad (12)$$

Constraints

$$\sum_{m=1}^{M} x_{im} = 1 \qquad \forall i, \qquad (13)$$

$$\sum_{i=1}^{N} x_{im} = 1 \qquad \forall m. \qquad (14)$$

Constraint (13) ensures that exactly one product is assigned to each position in a sequence. Constraint (14) guarantees that the demand of each model is satisfied in a cycle. Constraint (14) guarantees that the demand of each model is satisfied in a cycle by considering one position for each model of a product in a sequence:

$$z_{i+1j} = \max\left\{0, \min\left(z_{ij} + v_c * \sum_{m=1}^{M} x_{im} * t_{mj}\right.\right.$$

$$\left.\left. - a_i * v_c, l - a_i * v_c\right)\right\}, \qquad (15)$$

$$z_{1j} = j * l, \qquad (16)$$

$$z_{i1} = 0, \qquad \forall i. \qquad (17)$$

Constraints (15)-(17) calculate position of the ith product at station j whether the stoppage occurs or not. Constraint (15) shows that if stoppage does not occur,

the product position will be in the start of the next station. Also, if stoppage occurs, the product position will be more than the length of the station. Constraint (16) shows that the position of the first product at the end of an operation in each station is equal to the sum of the lengths of the stations through which the product has passed. Constraint (17) depicts that position of a product in the first station is equal to zero:

$$ stb_{ij} = \max\left\{0, \left(z_{ij} + v_c * \sum_{m=1}^{M} x_{im} * t_{mj} - l\right) \div v\right\}. \quad (18) $$

Constraint (18) indicates the stoppage time related to product i at station j when the worker has not yet received help from downstream worker or storehouse of kits:

$$ hpf_{ij} = stb_{ij} - \max\left\{t_c - \sum_{m=1}^{M} x_{i-1m} * t_{mj+1}, 0\right\}. \quad (19) $$

Constraint (19) indicates the reduced stoppage time when the downstream worker has enough time to help the worker that faces stoppage (Figure 6):

$$ hps_{ij} = 2* $$

$$ \sqrt{\left(l * \frac{j}{2} - z_{ij} - \sum_{m=1}^{M} x_{im} * t_{mj}\right)^2 + \left(\frac{ld}{4}\right)^{2/v}}. \quad (20) $$

Constraint (20) calculates the reduced stoppage time when the downstream worker does not have enough time to help the worker that faces stoppage and the worker uses the storehouse of kits (Figure 7):

Figure 6. Downstream help policy in Scenario 1.

Figure 7. Using storehouse of kits in Scenario 1.

$$ sta_{ij} = \begin{cases} hpf_{ij}, & \text{If downstream worker has} \\ & \quad \text{enough time to help} \\ hps_{ij}, & \text{Otherwise} \end{cases} \quad (21) $$

Constraint (21) represents the stoppage time after applying help policy:

$$ ta_{im} = \max\left(\left|du_{im} - x_{im} * \sum_{j=1}^{K} t_{mj}\right|, 0\right). \quad (22) $$

Constraint (22) is considered for MTO environment, which includes tardiness in completing the ith product.

4. Solution methodology

The scheduling problem in a mixed-model assembly line allows an optimal sequence with respect to a selected criterion to be carried out. The sequences are taken out of a set of N products belonging to M models. The generic solution is represented by using a binary encoding; each part can be associated with binary numbers obtaining a sequence such as 0, 1, 1, 1. The space of feasible solutions in such a scheduling problem is equal to the N! products to be processed. Therefore, the problem is NP-complete and can be solved as a permutation flow shop scheduling problem. To solve the problem, we use LINGO and MATLAB software. LINGO software gives exact results for small-sized problems. Also, for the large-sized problems, we use hybrid genetic algorithm-beam search with MATLAB software. Leu et al. [35] proposed genetic algorithms for scheduling mixed-model assembly lines in a JIT environment, showing how this approach outperformed other well established algorithms. Also, beam search technique has been incorporated into several AI applications developed at Carnegie-Mellon for job shop scheduling systems. More recently, Leo [36] used beam search to solve sequencing problem on a mixed-model assembly line and their results showed that beam search techniques were noticeably superior to several goal chasing algorithms. Although there are many variations of beam search, the basic approach involves searching a limited number of solution paths (sequence of models in the line) in parallel. The number of solution paths evaluated is called the beam width. The choice of solution paths is based on an evaluation function that estimates the "goodness" of a partial solution. Hence, the more effective the evaluation functions, the narrower the beam width is. Beam search navigates the solution space step by step without backtracking since no backtracking is allowed. There is no guarantee that the chosen paths are the optimal solutions; however, the technique is efficient because it uses only a fixed number of promising partial solutions at each stage. The advantage of beam search technique
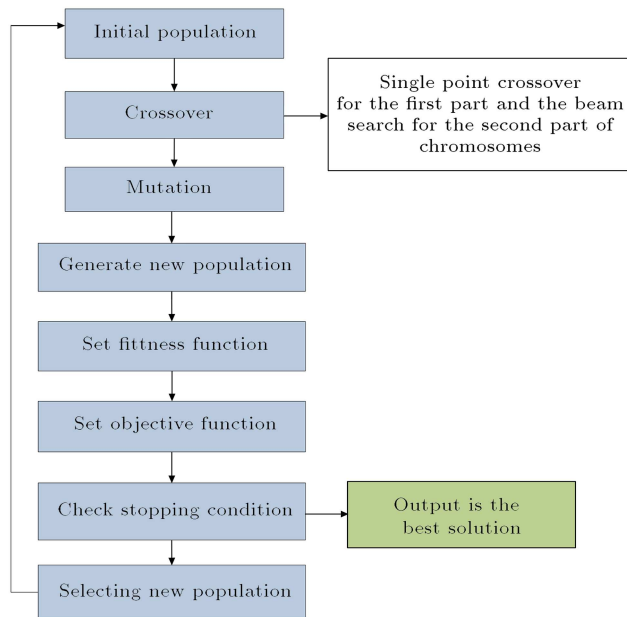
Figure 8. Hybrid GA-beam search flowchart.

is its ability to inspect a greater number of solution paths without significantly increasing computational complexity. In applying a hybrid GA-beam search to mixed-model assembly line sequencing problem, an individual binary encoding method has been adopted. To solve our proposed model in this paper, a hybrid GA-beam search is implemented as described in the following section (see Figure 8).

4.1. Initialization
In the first step, a number of feasible solutions have to be introduced as the initial population named the first generation. To do so, a set of N chromosomes are generated at random. The chromosome ch_i is represented by a sequence of genes $(g_i| = 1, ..., I)$. Each individual gene contains one bit of information: the ith product in a sequence is represented by the binary code of gene g_i in the sequence of the chromosomes.

4.2. Parent selection
A number of individuals are selected according to their fitness to create and form the next generation. This is carried out by using the roulette wheel selection without replacement. The simplified roulette wheel considers the quality of parents. Each chromosome, ch_i, in the population has an associated selection probability, p_i, that is proportional to its fitness value f_i. First, all chromosomes are ranked in ascending order based on their fitness values. Then, the selection probability of each chromosome is calculated as follows:

$$p_i = \exp\left(\frac{1 + f_i}{N + \sum_{l=1}^{N} f_l}\right);$$

$$f_i = -\beta * \left(\frac{f_{1i}(x)}{\max f_{1l}(x)} + \frac{f_{2i}(x)}{\max f_{2l}(x)}\right)$$

$$l = 1, ..., N$$

where $f_{ki}(x)$ is the kth objective function value for the ith chromosome in the population. Then, the cumulative probability c_i is calculated for each chromosome and a uniform random number in the interval [0,1] will determine the selected parents until the desired number of parents is reached. In addition to the above assumptions, we also assume that each chromosome must be selected only once. According to this assumption, the diversity in the mating pool will increase.

4.3. Crossover
The new method based on beam searching is proposed in this study for applying crossover. For this purpose, the first part of offspring 1 is obtained from one parent and also the first part of offspring 2 is obtained from another parent and the second part of the offspring is achieved by searching the remained sequences, which results in reduction of objective functions. This methodology is used to search better solutions. Moreover, it has less complexity than beam search method for finding efficient solutions because the first part of the sequence is given from the first parent and for the remained sequences, the beam search is applied.

4.4. Mutation
The mutation operator changes one or some of the genes in a single parent randomly. This operator has been used to increase the diversity. In this paper, two genes are randomly chosen and if they are not the same product, their position will change.

4.5. Elitist strategy
The idea of Elitist strategy is to provide the best chromosomes from the previous stage to the current stage without changing the gene structure. This guarantees existence of the best chromosomes in the succeeding generation. In this paper, an elitist strategy is used during the optimization process. During the evolution, the best chromosomes in the solution pool will be identified and collected.

This algorithm has been coded in the MATLAB R2011b and executed on an Intel (R) Core (TM) i3 CPU, 2.27 GHz, and Windows 7 using 4 GB of RAM.

5. Experimental results and discussion

For investigating the performance of the proposed hybrid GA-beam search, a number of problems are tested in this section.

5.1. Small-sized problems
In small size of the problem, we solve it with LINGO11

Table 1. Parameters of the small-sized problems.

Workstation	Processing time of models (t.u)		Workstation length (l.u)
	1	**2**	
1	340	550	800
2	400	670	800
3	700	440	800
4	520	690	800
Demand	2	2	
Due date (t.u)	1000, 4000	2500, 3600	
Launching interval for Scenario 1 (t.u)	600		
Launching interval for Scenario 2 (t.u)	500, 650, 600, 550		

Table 2. Test problem in small size.

	Sequence	Hybrid GA-beam search result	LINGO result
Scenario 1	1, 1, 2, 2	$Z = 955.5$	$Z = 955.5$
Scenario 2	1, 1, 2, 2	$Z = 856.3$	$Z = 856.3$

and MATLAB R2011b to ensure the accuracy of the proposed model. We consider 4 products, 2 models, and 4 workstations. Table 1 considers the processing time of every model in every station, workstation length, launching interval, demands, and due date for every model in the two proposed scenarios. All parameters are illustrated in unites, which are time unit (t.u) or length unit (l.u).

We deduce accuracy of the model from Table 2 because of the same sequence and similar results with those of LINGO and hybrid GA-beam search in both scenarios. Moreover, we obtain better results from scenario 2, which has a different launching interval.

5.2. Large-sized problems
For solving the medium and large sizes of the problem, we use the meta heuristic algorithm (hybrid GA-Beam

search) in the MATLAB R2011b. The performances of the proposed assembly line models have been tested by implementing a set of examples divided into six classes:

- Classes I and II: 6 stations, 20/30 products, and 3 models;
- Classes III and IV: 10 stations, 20/30 products, and 3 models;
- Classes V and VI: 20 stations, 20/30 products, and 3 models.

Each class includes 50 instances and the problems have been executed for the two scenarios. Parameters of the model are given from the paper of Celanoa et al. [10]. Working times of the models are randomly between 300-700 and due dates of the products are also randomly chosen between 1000-2000. Best sequencing of the products is shown in Table 3. To show the

Table 3. Sequencing of products.

Experiments	Scenario 1	Scenario 2
I	1,2,2,3,1,1,3,1,1,2,2,2,3,3,2,3,3,1,2,2	1,2,2,3,3,1,1,2,2,2,3,3,2,3,3,1,2,2,1,1
II	2,2,1,1,1,1,1,3, 3,3,2,2,3,3,1,1,1,3,3, 2,2,2,1,1,3,3,2,2,2,2	2,2,2,1,3,1,1,3, 2,3,2,1,3,3,1,1,1,3,3, 2,2,2,1,1,3,3,1,2,2,2
III	1,1,2,3,1,2,1,1,2,2,2,2,3,3,2,3,3,3,2,1	1,3,3,2,1,1,2,1,1,2,3,2,2,3,3,2,2,1,3,2,3
IV	1,1,1,1,2,1,2,1, 3,3,2,2,3,2,1,3,1,2,3, 2,3,2,1,1,3,3,2,3,2,2	1,1,2,1,1,1,2,1,3,3,2,2,3,2,3,1,1,2,3, 2,3,2,1,1,2,3,2,3,2,3
V	2,1,2,3,1,2,1,1,1,1,2,2,3,3,2,2,3,3,2,3	2,1,1,3,2,2,1,3,1,1,2,2,1,3,2,2,3,2,3,3
VI	3,2,1,2,3,1,1,3, 2,3,2,1,3,3,1,1,2,3,2, 2,1,2,1,1,2,3,2,3,2,1	3,2,2,2,3,1,1,3,3,3,1,1,3,3,1,1,2,2,2, 2,1,2,1,1,2,3,2,3,1,2

Table 4. Mean values of total stoppages and tardiness time.

Experiments	Scenario 1			Scenario 2		
	AI	AG	NO	AI	AG	NO
I	2133.25	2257.65	2200.5	2269.9	2291.1	2271.96
II	2395.2	2440.05	2445.21	2443.35	2455.53	2454.09
III	2266	2275.28	2285.94	2267.4	2274.63	2288.65
IV	2451.53	2455.55	2475.262	2424.25	2459.1	2442.89
V	231297	2375.54	2390.99	2232.76	2277.17	2257.47
VI	2433.6	2449.13	2459.68	2425.8	2439.6	2429.57

efficiency of our proposed techniques and the proposed solution methodology, several experiments have been considered for both scenarios:

- Downstream help or use of kitting storage is not allowed (NO);

- Downstream help or use of kitting storage is allowed (AI);

- Traditional GA is applied (AG).

The following parameters have been settled for the algorithm: number of iterations Nit = 100, probability of mutation is 0.65, profitability of crossover is 0.7. Table 3 reports the mean values of total stoppage times and total due date tardiness. The values are expressed in time units.

As can be seen in Table 4, hybrid GA-beam search yields better solutions than the traditional genetic algorithm does. Moreover, applying different launching intervals gives a better solution than the equal launching intervals do.

6. Conclusion and future research directions

In the Mixed-Model Assembly Line (MMAL) problem, determining the sequence for the arrival of the products in the production line is very important and effective on the cost of production. The suitable sequence can reduce the stoppages of line or tardiness in delivery of the products to the customers. Also, applying downstream help or using storehouse of kits when stoppage occurs can reduce total stoppages time and tardiness in delivery of products to the customers. Moreover, applying different launching intervals results in reduction of total stoppages and tardiness in the line. In small size of the problem, we solve it with the LINGO software and the hybrid GA-beam search and, then, the results are compared with each other to check the accuracy of the model. Since this problem is Np-hard, to solve it in the large size, we should use the meta-heuristic approach. The hybrid GA-beam search is one of the meta-heuristic methods that yield better solution than traditional GA do.

To ameliorate this problem, we can consider some other objectives to approach the real situation. This problem can be solved with the other meta-heuristic methods, like the PSO, SA, etc. or the hybrid of them.

Acknowledgment

References

1. Xiaobo, Z., Zhou, Z. and Asres, A. "A note on Toyota's goal of sequencing mixed models on an assembly line", *Computers and Industrial Engineering*, **36**, pp. 57-65 (1999).

2. Kubiak, W. "Minimizing variation of production rates in just-in-time systems: a survey", *European Journal of Operations Research*, **66**, pp. 259-271 (1993).

3. Yano, C.A. and Rachamadugu, R. "Sequencing to minimize work overloading assembly lines with product options", *Managment Science*, **37**, pp. 572-586 (1991).

4. Xiaobo, Z. and Ohno, K. "Algorithms for sequencing mixed models on an assembly line in a JIT production system", *Computers and Industrial Engineering*, **32**(1), pp. 47-56 (1997).

5. Xiaobo, Z. and Ohno, K. "Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages", *European Journal of Operations Research*, **124**, pp. 560-570 (2000).

6. Miltenburg, J. "Balancing and scheduling mixed-model Ushaped production lines", *International Journal of Flexible Manufacturing Systems*, **14**, pp. 119-151 (2002).

7. Manavizadeh, N., Hosseini, M. and Rabbani, M. "Solving a new mixed-model assembly line sequencing problem in a MTO environment", *World Academy of Science, Engineering and Technology*, **5**, pp. 408-417 (2011).

8. Kilbridge, M. and Wester, L. "The assembly line model-mix sequencing problem", In *Proceeding of the Third International Conference on Operations Research*, Oslo, Paris: English Universities Press, Dunod Editeur, **247**, pp. 247-259 (1963).

.

Advanced Manufacturing Technology, **44**, pp. 345-359 (2009).

35. Leu, Y.Y., Matheson, L.A. and Rees, L.P. "Sequencing mixed-model assembly lines with genetic algorithms", *Computers and Industrial Engineering*, **30**, pp. 1027-1036 (1996).

36. Leu, Y.Y. "Using beam search techniques for sequencing mixed-model assembly lines", *Annals of Operations Research*, **70**, pp. 379-397 (1997).

Biographies

Masoud Rabbani is a Professor of Industrial Engineering in the School of Industrial and Systems Engineering at University of Tehran. He has published more than 60 papers in international journals, such as European Journal of Operational Research, International Journal of Production Research, and International Journal of Production Economics, among others. His research interests include production planning, design of inventory management systems, applied graph theory in industrial planning, and productivity management.

Neda Manavizadeh received her MSc in Industrial Engineering from the College of Engineering, University of Tehran, in 2005. She obtained her PhD in Industrial Engineering in the College of Engineering, University of Tehran. Currently, she is an Assistant Professor in the Industrial Engineering Department, Khatam University, Tehran, Iran. Her research interests are production planning, supply chain management, operations management, and operations research (OR).

Nazanin Shabanpour is PhD candidate at Politecnico di Milano University. She graduated from Tehran University with MSc degree in Industrial Engineering. She also attended the International Institute of Industrial Engineering (IIE) conference held in Istanbul, Turkey. Her present research interests include manufacturing system design and disassembling line.