Tamkang Journal of Science and Engineering, Vol. 13, No. 3, pp. 327–336 (2010)

Simultaneous Perspective-Based Mixed-Model Assembly Line Balancing Problem

Horng-Jinh Chang[1] and Tung-Meng Chang[1,2]*

[1]Graduate Institute of Management Sciences, Tamkang University,
Tamsui, Taiwan 251, R.O.C.
[2]Department of Industrial Engineering & Management, Tungnan University,
Taipei, Taiwan 222, R.O.C.

Abstract

In this paper, we discuss a mixed-model assembly line balancing problem with multi-manned workstations, where workers simultaneously perform different tasks on the same product and workstation. This situation requires that the product is large-sized such as vehicle's manufacturing. A mathematical model for the mixed-model assembly line balancing problem with simultaneous production (MALBPS) is developed to decide the optimal number of workstations. A coding system, Four-Position Code (FPC), is proposed to re-code the tasks to tackle this issue, and a computerized coding program written in C++ to generate those FPCs is also provided. An illustrative example has been solved by Lingo 9.0 extended version, and the simulation analyses and some computational properties of the model are also given.

Key Words: Assembly Line Balancing, Multi-Manned Workstation, Simultaneous Production

1. Introduction

Assembly is a process by which subassemblies, manufactured parts and components are put together to make the final products. An assembly line balancing (ALB) is an important problem in the industrial production. In the most common statement of the assembly line balancing problem, a set of tasks having fixed duration time is assigned to a set of sequential workstations without violating the precedence constraints and without having the work contents assigned at any workstations exceed the cycle time. Baybars [1] and Becker and Scholl [2] provided a survey of assembly line balancing problem; Boysen et al. [3] also presented a classification scheme for ALB to ease communication between researchers and practitioners. In addition, a comprehensive review of ALB and its solution procedures are provided by Scholl and Becker [4]. For line balancing problems, some different objectives are optimized. They are related to minimizing the idle time of the assembly line [2,3].

Dimitriadis [5] examined paced single-model assembly lines with multi-manned workstations, which are widely used in producing large-sized products such as the case of vehicle's final assembly in the vehicle manufacturing, and proposed a two-level heuristic algorithm to tackle the case when workers simultaneously performing the tasks are assigned to a single workstation. During the cycle time, each multi-manned workstation has several workers simultaneously performing different tasks on the same individual product. Multi-manned assembly line in practice has several advantages over a traditional assembly line such as shorter line length, less throughput time and work in process, and lower material handling costs. Moreover, because of high variety demand scenarios, manufacturing firms are usually requested to meet the consumers' expectation quickly for their demands of various products. Therefore, mixed-model assembly lines, in which various product models similar in product characteristics can be assembled simultaneously, are be-

---

*Corresponding author. E-mail: tmchang@mail.tnu.edu.tw coming popular in recent decades. Gokcen and Erel [6] proposed a binary integer programming model for the mixed-model assembly line balancing problem (MALBP). Balancing assembly lines is an NP-hard combinatorial optimization problem [7,8]. Hence, the combinatorial nature of the MALBP makes it difficult to obtain optimal solution. Whenever a simultaneous production is allowed for the entire tasks in an assembly line, the mixed-model assembly line balancing problem become more complicated.

According to Van Zante-de Fokkert and de Kok [9], they stated that two mostly used methods are used to reduce the multiple product models into a single one. The first approach combines the precedence diagrams of the different product models into a single, which is the so-called combined precedence diagram. The second method uses adjusted task processing times. There are many researches using the concept of a combined precedence diagram to transform different product models into an equivalent single product model [6,10–12,13,14].

In this paper, a mathematical programming model for solving the mixed-model assembly line balancing problems with simultaneous production is developed to decide the optimal number of multi-manned workstations. In order to solve such a complicated problem, a coding system, Four-Position Code (FPC), is proposed to re-code the tasks, and a computerized coding program written in C++ to generate those FPCs is also provided.

The rest of the paper is organized as follows. Section 2 describes the problem and the solving procedures, and presents a coding system to cope with the situation of simultaneous production. Section 3 develops the model formulation for MALBPS. The proposed model is further clarified by an illustrative example in section 4. In section 5, simulation analyses for various cycle times and performance of the proposed model are conducted. Some conclusions are given in section 6.

2. Problem Statement and Coding System

2.1 Problem Statement

This paper considers the situation of several different product models to be produced one after another in a paced assembly line with multi-manned workstations. The product model stops during the cycle time at each multi-manned workstation where there are several workers simultaneously performing different tasks on the same individual product model. Each product model is composed of several parts, and each part has its task routes to be performed. Each task route consisting of the tasks from source node (task) to sink on the precedence diagram represents the partial manufacturing process of a part. And, different parts may require the same tasks to be made during their manufacturing, and thus the manufacturing flows of the tasks of these parts can be expressed as a precedence diagram, where the nodes on the diagram are numbered according to a topological ordering and the processing time of each task is constant and deterministic. The objective of this work is to minimize the number of workstations, and then further identifies the number of workers of each multi-manned workstation, and computes the total idle rate for the MALBPS under given cycle times.

This kind of simultaneous operation requires the product to be of sufficient size such as vehicle's final assembly. The product is released from a multi-manned workstation when all workers have completed their work. However, in a multi-manned assembly line, some tasks of a part can be delayed by the tasks of other parts assigned to the same workstation. Therefore, balancing multi-manned assembly lines needs to consider the sequence-dependent completion time of tasks, which is a feature specific to multi-manned assembly line. Moreover, some preconditions must be fulfilled so that all these tasks of different parts can be manufactured simultaneously in a multi-manned assembly line [5].

From the point of view in industrial engineering, for a worker performing some consecutive tasks of a part is smoother than doing different tasks of different parts. Especially, there are ergonomic relations among these consecutive tasks. In addition, under considering the simultaneous production, creating the combined precedence diagram may lead to more common tasks. A task simultaneously belonging to different task routes of a part, to different parts, or to different product models is called common task. So, in this paper the task assignments to a multi-manned workstation are performed in terms of the task routes of individual part. That is, the partial tasks belonging to a part, which are assigned into a multi-manned workstation, are performed by one worker.

Besides, for each product model, the amount of time available at each workstation is called cycle time. The difference between the cycle time and the sum of the task times assigned at any workstation is called workstation's idle time. Hence, the total idle time of an assembly line is the sum of workstation's idle time over all workstations, and the total idle rate is a ratio of the total idle time to the time available in the line.

2.2 Coding System

When the precedence diagrams of all parts are transformed into the combined precedence diagram, more precedence relations between tasks and more common tasks assignments have to deal with. Hence, when the task assignments to a multi-manned workstation are performed by the task routes of individual part, balancing the line will encounter the following questions: (a) task codes belonging to a common task must assign to the same workstation, (b) when the common tasks belonging to different task routes of a part are assigned to the same workstations, they will be assigned repeatedly and the processing times of the common tasks will also be computed repeatedly. In order to deal with the above-mentioned situations during assignments, a coding system, Four-Position Code , is presented to re-code the tasks on the combined precedence diagram so as to tackle these questions. The coding system is an easy way to convert the tasks on the combined precedence diagram into FPCs, and can help resolve the repeated time computation for common task assignments. In addition, based on the optimal solution, the line designer can use FPCs quickly to identify the workstations where all tasks of each part are assigned, and further to allocate the workers to each multi-manned workstation.

A FPC uses four positions (M, P, R, S) to represent each task on the combined precedence diagram. For a task code (M, P, R, S), M denotes the product model number where such a given task should be belonged; P means the part number where a given task should be assigned; R shows the task route number where a given task should be belonged; S indicates the operation sequence number of a given task in its specific product model, part, and task route. Because there may have some common tasks among different task routes, among different parts, and among different product models, the proposed coding system probably make a task to have one more codes, and these codes have to be assigned into the same workstation during assignments.

In order to easily construct the model formulation for MABLPS and solve it in optimization tools, a flow chart of solving procedures (shown in Figure 1) is provided. The primary procedures of this flow are as follows.

Step 1. Generate the combined precedence diagram. The precedence diagrams of different product models can be transformed into the combined precedence diagram.

Step 2. Based on the precedence diagram of each product model and the combined precedence diagram, the precedence matrices can be created individually. A precedence matrix is an upper-triangular matrix with an *ab*-th entry of 1 if task *a* precedes immediately task *b*; otherwise is zero [15].

Step 3. Input the generated individual precedence matrices and combined precedence matrix into the 'Codes Generator' to generate the task routes of each part and the Four-Position Code for each task on the combined precedence diagram.

Step 4. Run the mathematical model in an optimization tool. After FPCs of all tasks are generated, the proposed mathematical model can be solved in an optimization tool under simultaneous assignment perspective.

Step 5. Analyze the results.

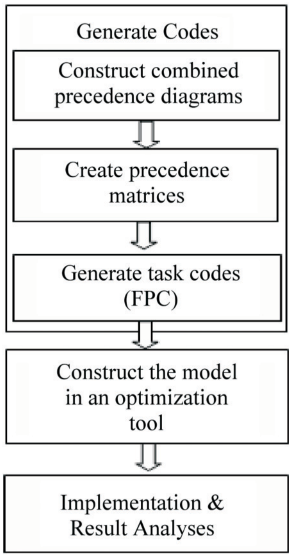


Figure 1. The flow chart for implementing Lingo modeling.

3. Model Formulation

3.1 Notations

The parameters used in the formulation are listed below:

M	total number of product models,
N	total number of tasks on the combined precedence diagram,
L	total number of FPCs where tasks on the combined precedence diagram are converted,
R_m	total number of common tasks of different task routes and different parts for product models m,
S	total number of common tasks of different product models,
$\overline{W}$	maximum number of assignable workstations,
$N_m^{\max}$	maximum number of task codes (FPCs) assignable to a multi-manned workstation, i.e. the maximum number of task codes of product model m which can be assigned into a multi-manned workstation given the cycle time, where $m = 1, 2, \ldots, M$,
P_m	total number of parts of product model m on the combined precedence diagram, where $m = 1, \ldots, M$,
CT_m	cycle time of product model m in an assembly line, where $m = 1, \ldots, M$,
t_{im}	processing time of the i-th FPC of product model m, where $i = 1, 2, \ldots, L$, $m = 1, \ldots, M$,
J_{ipm}	number of task codes of a task which the ith task code of part p of product model m corresponds to on the combined precedence diagram, where $i = 1, 2, \ldots, L$, $p = 1, 2, \ldots, P_m$, $m = 1, \ldots, M$,
IP_m	set of all arcs (i, j) for all tasks of product model m on the combined precedence diagram, i.e. the set of precedence relations of all task codes corresponding to tasks on the combined precedence diagram, where $m = 1, \ldots, M$,
I_{hm}	subset of FPCs of the h-th common task among the different task routes, and the different parts for product model m, $h = 1, 2, \ldots, R_m$, $m = 1, 2, \ldots, M$,
K_h	subset of FPCs of the h-th common task among different product models, $h = 1, 2, \ldots, S$,
A_{pmw}	subset of FPCs belonging to the p-th part of product model m assigned to the w-th workstation, where $p = 1, 2, \ldots, P_m$, $w = 1, 2, \ldots, \overline{W}$, $m = 1, \ldots, M$,
B_m	subset of all FPCs of product model m, where $m = 1, \ldots, M$.

The variables are as follows:

W	number of assigned workstations,
x_{iw}	binary variable, where $x_{iw} = 1$ if Four-Position Code i is assigned to workstation w; otherwise, $x_{iw} = 0$,
V_{wm}	binary variable, where $V_{wm} = 1$ if workstation w is utilized for model m; otherwise, $V_{wm} = 0$,
N_w	binary variable, where $N_w = 1$ if workstation w is utilized by all models; otherwise, $N_w = 0$.

3.2 Assumptions

The assumptions of the proposed model are described as follows:

1. The tasks of all parts of each product model can be expressed as a precedence diagram, and the parts can be produced simultaneously in an assembly line.
2. The processing time of each task associated with each product model is known constant.
3. Common tasks among the parts and among the product models do not need to have the same processing times.
4. Task codes belonging to the common tasks of different task routes, of different parts, and of different product models have to assign into the same workstations.
5. The number of multi-manned workstations needed is the same for all product models.
6. Work in process (WIP) inventory buffer is not considered between workstations.

3.3 Model Formulation

The mathematical model for solving the mixed-model multi-manned assembly line balancing problem with simultaneous production is formulated as follows.

$$Min \quad W = \sum_{w=1}^{\overline{W}} N_w \tag{1}$$

s.t.

$$\sum_{w=1}^{\overline{W}} x_{iw} = 1 \qquad i = 1,2,\ldots,L \tag{2}$$

$$\sum_{i \in A_{pmw}} \frac{1}{J_{ipm}} t_{im} x_{iw} \le CT_m \quad \forall p, w, m \tag{3}$$

$$\sum_{w=1}^{\overline{W}} (w x_{jw} - w x_{iw}) \ge 0 \quad \forall (i, j) \in IP_m, m = 1,2,\ldots,M \tag{4}$$

$$\sum_{w=1}^{\overline{W}} (wx_{iw} - wx_{jw}) = 0 \qquad i \neq j, \forall i, j \in I_{hm}, \\ h = 1,2,\ldots,R_m, m = 1,2,\ldots,M \tag{5}$$

$$\sum_{w=1}^{\overline{W}} (wx_{iw} - wx_{jw}) = 0 \quad i \neq j, \forall i, j \in K_h, h = 1,2,\ldots,S \tag{6}$$

$$\sum_{i \in B_m} x_{iw} - N_m^{\max} \cdot V_{wm} \leq 0, \quad \forall m, w, m = 1,2,\ldots,M, w = 1,2,\ldots,\overline{W} \tag{7}$$

$$\sum_{m=1}^{M} V_{wm} - M \cdot N_w = 0 \qquad w = 1,2,\ldots,\overline{W} \tag{8}$$

$$x_{iw}, V_{wm}, N_w \in \{0,1\} \quad \forall i, m, w \tag{9}$$

The proposed model for the MALBPS is to minimize the number of workstations of the assembly line as given in equation (1). Equation (2) guarantees that every task (task code) of each product model on the combined precedence diagram is assigned to exactly one workstation. Equation (3) shows that the sum of the processing times of the tasks for each part of each product model within a workstation cannot be greater than the cycle time of the product model. When a common task belonging to a part is assigned to a workstation, it will be assigned repeatedly during assignments. That is, the processing time of the common task assigned to the workstation will be computed repeatedly. So, the term $\frac{1}{J_{ipm}}$ considered in the equation (3) will resolve the problem of repeated computation. Equation (4) means that the precedence relations of all tasks have to be observed, where *i* is the immediate precedence task of *j*. For each product model, equation (5) denotes that the different FPCs representing the same task simultaneously belonging to different task routes of a part, to different parts, or to different task routes and different parts have to be assigned to the same workstations. And, equation (6) represents that the FPCs showing the same task simultaneously belonging to different product models have also to be assigned to the same workstations. Equations (7) and (8), developed by Gokcen & Erel [6], are cited and modified for the workstations constraints. The equations will make sure that the number of workstations is the same for all product models. The last equation, equation (9), shows that x_{iw}, V_{wm}, and N_w are binary variables. Besides, x_{iw} and W are decision variables of the proposed model.

According to the optimal solution, the total idle rate, Z, of the line can be computed as follows:

$$Z = \frac{IT}{TA} \tag{10}$$

where $IT = \sum_{m=1}^{M} \sum_{w=1}^{W} \sum_{p=1}^{p_m} (CT_m - \sum_{i \in A_{pmw}} \frac{1}{J_{ipm}} t_{im} x_{iw})$, it indicates the sum of the idle time of all assigned multi-manned workstations, and $TA = \sum_{m=1}^{M} \sum_{o=1}^{p_m} W_{pm} \times CT_m$, it denotes the total time of assignments available for all multi-manned workstations in an assembly line, where W_{pm} means the number of workstations of part p of product model m needed on the line.

4. Illustrative Example

In this section, an illustrative example is provided to show how to utilize coding system to solve the MALBPS problem in Lingo. Suppose that there are two product models to be produced simultaneously in a multi-manned assembly line. Product model 1 has three different parts, and product model 2 has four different parts. The manufacturing flow of each part for two product models is shown in Figure 2. The part 1 of product model 1 consists of two task routes, and all other parts have only one task route. So, each product model consists of

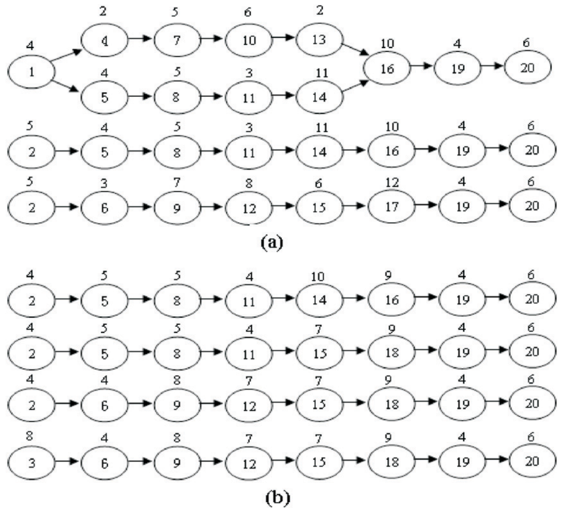


Figure 2. The manufacturing flows of each part for (a) model 1, (b) model 2 of the illustrative example.

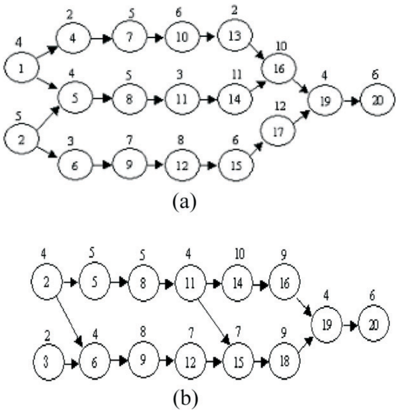
four different task routes, respectively. Besides, every part of each product model can be simultaneously produced in the assembly line. The precedence diagrams for both product models can be expressed as Figure 3. The numbers above the nodes on each precedence diagram represent task processing times.

The steps to solve this illustrative example are described as follows. First, construct the combined precedence diagram from the precedence diagrams of the two product models (shown in Figure 4). In total, there are actually eight task routes except some redundant routes on the combined precedence diagram (listed in Table 1). All kinds of common tasks are listed in Table 2. The part 2 of product model 1 and the part 1 of product model 2 have the same task route (part) 2, 5, 8, 11, 14, 16, 19, and 20.

Second, generate the precedence matrices (shown in Figure 5) and the combined precedence matrix (shown in Figure 6) from the precedence diagrams of the two product models and the combined precedence diagram, respectively. Third, the precedence matrices are inputted into the 'Codes Generator' to create the FPCs of each task. The code of every task is summarized in Table 3.

After the entire FPCs of every task of each product model are generated, the next step is to build the proposed mathematical model in Lingo syntax. Each task code and the precedence relations between tasks for the two product models are declare and defined by the Four-Position Code. Then, the code sets of tasks belonging to common tasks are declared. For example, FPCs in J_6 set, where J_6 = {(1,3,1,2), (2,3,1,2), (2,4,1,2)}, mean the FPCs for common task 6 simultaneously belonging to the part 3 of product model 1 and the parts 3 and 4 of product model 2. FPCs in I_{51} set, where I_{51} = {(1,1,2,2), (1,2,1,2)}, represent the codes of common task 5 for part 1 and part 2 of product model 1 (shown in Table 3).They must be assigned to the same workstations.

Theoretically, the desired limits on cycle times should be $t_{\max} \le CT \le \sum t_i$, where $t_{\max}$ and $\sum t_i$ mean the longest task processing time among all tasks and the total



(a)

(b)

Figure 3. The precedence diagrams for (a) model 1, (b) model 2 of the illustrative example.

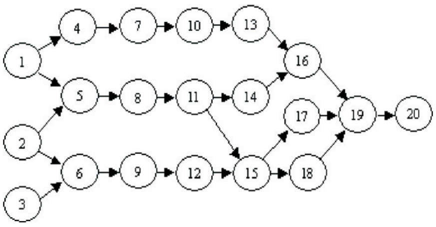


Figure 4. The combined precedence diagrams of the illustrative example.

Table 1. The total task routes on the combined precedence diagram

Model	Part	Route	Task Unit (in order)							
1	1	1	1	4	7	10	13	16	19	20
		2	1	5	8	11	14	16	19	20
	2	1	2	5	8	11	14	16	19	20
	3	1	2	6	9	12	15	17	19	20
	*	1	1	5	8	11	15	17	19	20
	*	1	1	5	8	11	15	18	19	20
	*	1	2	5	8	11	15	17	19	20
2	1	1	2	5	8	11	14	16	19	20
	2	1	2	5	8	11	15	18	19	20
	3	1	2	6	9	12	15	18	19	20
	4	1	3	6	9	12	15	18	19	20
	*	1	3	6	9	12	15	18	19	20

*denote the redundant task routes on the combined precedence diagram.

Table 2. The common tasks for the combined precedence diagram

Common tasks for model 1	Common tasks for model 2	Common tasks for two models
1	2	2
2	5	5
5	6	6
8	8	8
11	9	9
14	11	11
16	12	12
19	15	14
20	19	15
	20	16
		19
		20

$$P_1 = \begin{array}{c|cccccccccccccccccc}
 & 1 & 2 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 19 & 20 \\ \hline
1 & - & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & & - & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
11 & & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
12 & & & & & & & & & & & - & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
13 & & & & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 \\
14 & & & & & & & & & & & & & - & 0 & 1 & 0 & 0 & 0 \\
15 & & & & & & & & & & & & & & - & 0 & 1 & 0 & 0 \\
16 & & & & & & & & & & & & & & & - & 0 & 1 & 0 \\
17 & & & & & & & & & & & & & & & & - & 1 & 0 \\
19 & & & & & & & & & & & & & & & & & - & 1 \\
20 & & & & & & & & & & & & & & & & & & -
\end{array}$$

(a)

$$P_2 = \begin{array}{c|cccccccccccccc}
 & 2 & 3 & 5 & 6 & 8 & 9 & 11 & 12 & 14 & 15 & 16 & 18 & 19 & 20 \\ \hline
2 & - & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & & - & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & & & - & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & & & & - & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & & & & & - & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & & & & & & - & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
11 & & & & & & & - & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
12 & & & & & & & & - & 0 & 1 & 0 & 0 & 0 & 0 \\
14 & & & & & & & & & - & 0 & 1 & 0 & 0 & 0 \\
15 & & & & & & & & & & - & 0 & 1 & 0 & 0 \\
16 & & & & & & & & & & & - & 0 & 1 & 0 \\
18 & & & & & & & & & & & & - & 1 & 0 \\
19 & & & & & & & & & & & & & - & 1 \\
20 & & & & & & & & & & & & & & -
\end{array}$$

(b)

Figure 5. The precedence matrices of (a) model 1, (b) model 2 of the illustrative example.

$$P_{12} = \begin{array}{c|cccccccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ \hline
1 & - & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & & - & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
11 & & & & & & & & & & & - & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
12 & & & & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
13 & & & & & & & & & & & & & - & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
14 & & & & & & & & & & & & & & - & 0 & 1 & 0 & 0 & 0 & 0 \\
15 & & & & & & & & & & & & & & & - & 0 & 1 & 1 & 0 & 0 \\
16 & & & & & & & & & & & & & & & & - & 0 & 0 & 1 & 0 \\
17 & & & & & & & & & & & & & & & & & - & 0 & 1 & 0 \\
18 & & & & & & & & & & & & & & & & & & - & 1 & 0 \\
19 & & & & & & & & & & & & & & & & & & & - & 1 \\
20 & & & & & & & & & & & & & & & & & & & & -
\end{array}$$

Figure 6. The precedence matrix of the combined diagram of the illustrative example.

task time required to assemble one unit, respectively. And, the desired range of the number of workstations should be $1 \le W \le \overline{W}$, where $\overline{W}$ means the maximal number of workstations available, and it usually can be set by management with the consideration of budget, floor space, and operational limitations. According to the above-mentioned statements, in this paper the cycle times for both product models and the maximal number of workstations available are given as $CT_1 = 16$, $CT_2 = 14$, and $\overline{W} = 6$, respectively.

Through the FPCs, the proposed model formulation for the MALBPS can be expressed in Lingo syntax; the built-in 'Integer Solver' is selected as solving method because the mathematical model is an integer linear programming program. The problem is solved

Table 3. The codebook of each task for the proposed example

Model	Part	Task	Code	Model	Part	Task	Code	Model	Part	Task	Code
1	1	1	(1,1,1,1)	1	2	19	(1,2,1,7)			15	(2,2,1,5)
			(1,1,2,1)			20	(1,2,1,8)			18	(2,2,1,6)
		4	(1,1,1,2)		3	2	(1,3,1,1)			19	(2,2,1,7)
		5	(1,1,2,2)			6	(1,3,1,2)			20	(2,2,1,8)
		7	(1,1,1,3)			9	(1,3,1,3)		3	2	(2,3,1,1)
		8	(1,1,2,3)			12	(1,3,1,4)			6	(2,3,1,2)
		10	(1,1,1,4)			15	(1,3,1,5)			9	(2,3,1,3)
		11	(1,1,2,4)			17	(1,3,1,6)			12	(2,3,1,4)
		13	(1,1,1,5)			19	(1,3,1,7)	2		15	(2,3,1,5)
		14	(1,1,2,5)			20	(1,3,1,8)			18	(2,3,1,6)
		16	(1,1,1,6)	2	1	2	(2,1,1,1)			19	(2,3,1,7)
			(1,1,2,6)			5	(2,1,1,2)			20	(2,3,1,8)
		19	(1,1,1,7)			8	(2,1,1,3)		4	3	(2,4,1,1)
			(1,1,2,7)			11	(2,1,1,4)			6	(2,4,1,2)
		20	(1,1,1,8)			14	(2,1,1,5)			9	(2,4,1,3)
			(1,1,2,8)			16	(2,1,1,6)			12	(2,4,1,4)
	2	2	(1,2,1,1)			19	(2,1,1,7)			15	(2,4,1,5)
		5	(1,2,1,2)			20	(2,1,1,8)			18	(2,4,1,6)
		8	(1,2,1,3)		2	2	(2,2,1,1)			19	(2,4,1,7)
		11	(1,2,1,5)			5	(2,2,1,2)			20	(2,4,1,8)
		14	(1,2,1,4)			8	(2,2,1,3)				
		16	(1,2,1,6)			11	(2,2,1,4)				

utilizing Lingo 9.0 extended version software on a 1.8 GHz personal computer. Five workstations are utilized in the optimal solution, and the workers of each multi-manned workstation for two product models are allocated (shown in Table 4). Based on the optimal solution, the total idle rate is 0.256. In the simultaneous production perspective, Figure 7 reveals that all tasks are simultaneously assigned to every multi-manned assembly line. The optimal number of workstations indicates that the total idle rate resulting from such a solution is minimal.

5. Simulations and Performance of the Model

For the above-mentioned illustrative example, three conditions are conducted to simulate the changes of the optimal solution with different cycle times. First, the cycle times of the two product models are synchronously increased. Second, CT_1 is increased under a fixed CT_2. The last, CT_2 varies under a fixed CT_1. In this example, the longest task processing times of the two product models are 12 minutes and 10 minutes, and the allowed largest cycle times among different parts of both product models are 62 minutes and 53 minutes, respectively. Besides, the maximal number of workstations available is set to 6. The simulated results are listed in Table 5, 6, and 7 and summarized as follows:

1. The optimal assigned number of workstations is decreasing if the cycle times are synchronously increasing (shown in Table 5).
2. For the same optimal assigned number of workstations during the simulation, the total idle rate increases when the pair of cycle times increases synchronously.
3. When the pair of cycle times is given as CT_1 =14 minutes and CT_2 =12 minutes, the total idle rate, 0.274, is minimization, and the optimal assigned number of workstations (W) equals to 6 (shown in Table 5). To be more specific, the production rate of this design is maximal.
4. Based on Tables 6 and 7, they reveal that the optimal assigned number of workstations is determined by the less cycle time under the pair of given cycle times

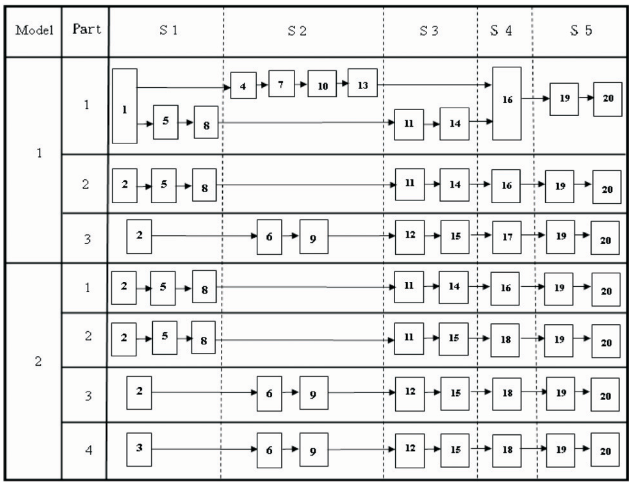


Figure 7. The graph of the optimal workstation assignments for all tasks.

Table 4. Optimal workstation assignments of the illustrative example

Workstation	Tasks	Model 1 Tasks	Workers for model 1	Model 2 Tasks	Worker for model 2
1	1,2,3,5,8	1,2,5,8	3	2,3,5,8	4
2	4,6,7,9,10,13	4,6,7,9,10,13	2	6,9	2
3	11,12,14,15	11,12,14,15	3	11,12,14,15	4
4	16,17,18	16,17	3	16,18	4
5	19,20	19,20	3	19,20	4

Table 5. The optimal solution for various cycle times under $\overline{W}$ = 6

	CT_1	12	14	16	18	20	22	24	26	…	32	34	…	62
W	CT_2	10	12	14	16	18	20	22	24	…	30	32	…	53
1		-	-	-	-	-	-	-	-	-	-	-	…	.110
2		-	-	-	-	-	-	-	-	-	.180	.230	…	-
3		-	-	-	-	-	.245	.262	.322	-	-	-	…	-
4		-	-	-	-	.329	-	-	-	-	-	-	…	-
5		-	-	.256	.342	-	-	-	-	-	-	-	…	-
6		-	.274	-	-	-	-	-	-	-	-	-	…	-

(CT_1, CT_2). Table 6 shows that the optimal assigned number of workstations determined by the less cycle time $CT_2 = 12$ is six workstations with minimal total idle rate. In addition, Table 7 reveals that the optimal assigned numbers of workstations are also determined by the less cycle times.

We also have attempted to solve problems with various sizes. Table 8 depicts the sizes of the problems, the number of variables and the runtimes. The Flexibility ratio (F-ratio), developed by Dar-El [16], measures the precedence relations between tasks. If H is the number of zero cells above the diagonal in the matrix, then the F-ratio is defined as

$$F-ratio = \frac{2H}{D(D-1)}$$

where D is the number of tasks in the problem. As depicted in Table 8, the expected results reveal that as increasing the number of tasks and the number of variables, while decreasing the F-ratio values, the computational requirements increase. For the problems having same number of tasks, the F-ratio values of the problems and the number of routes mainly affects the runtimes needed to obtain the optimal solutions. Besides, when the problem size is with up to 50 tasks on the combined diagram, the optimal solution is hard to obtain.

Table 6. The optimal solution for various CT_1 under $\overline{CT_2}$, $\overline{W} = 6$

	CT_1	14	16	18	20	22	24	32	36	…	62
W	CT_2	12	12	12	12	12	12	12	12	…	12
1		-	-	-	-	-	-	-	-	…	-
2		-	-	-	-	-	-	-	-	…	-
3		-	-	-	-	-	-	-	-	…	-
4		-	-	-	-	-	-	-	-	…	-
5		-	-	-	-	-	-	-	-	…	-
6		.274	.315	.323	.354	.351	.392	.451	.481	…	.598

Table 7. The optimal solution for various CT_2 under $\overline{CT_1}$, $\overline{W} = 6$

	CT_1	22	22	22	22	22	22	22	22	…	22
W	CT_2	14	16	18	24	26	30	32	40	…	53
1		-	-	-	-	-	-	-	-	…	-
2		-	-	-	-	-	-	-	-	…	-
3		-	-	-	.271	.306	.366	.391	.477	…	.576
4		-	.319	.358	-	-	-	-	-	…	-
5		.301		-	-	-	-	-	-	…	-
6		-	-	-	-	-	-	-	-	…	-

Table 8. Experimentation results

Number of tasks	F-ratio	Number of routes	Number of variables	Number of constraints	number of iterations	Elapsed runtime (sec.)
10	0.71	5	93	87	0	0*
10	0.69	7	125	132	0	0
10	0.67	9	157	188	0	0
20	0.88	5	345	174	3692	2
20	0.82	8	537	322	13819	8
20	0.81	11	729	499	17448	17
30	0.94	7	871	390	64014	152
30	0.91	11	1351	729	606045	1736
50	0.96	9	1687	499	895881	1368
50	0.94	12	1909	729	5611603	11950

* denotes the elapsed runtime less than 1 second.

6. Conclusion

To the best of our knowledge, there is no one considering multi-manned workstations assignment for the mixed-model assembly line balancing problems. For a multi-manned assembly line, considering simultaneous production perspective makes the multi-manned assembly line balancing problems complicated.

This paper has presented a mathematical model for MALBPS and it can be categorized as an Integer linear Programming model. A coding system, Four-Position Code, and its associated computerized program 'Codes Generator' are proposed to make the above-mentioned problem solvable when the task assignments to a multi-manned workstation are performed in terms of the task routes of individual part. Through conducting the simulation, the number of workstations of the line and the number of workers for each multi-manned workstation can be determined by the proposed model so as to decide the related production decisions for various demands of products. The experimentation revealed that the proposed model is hard to obtain the optimal solution when the problem size is with up to 50 tasks on the combined precedence diagram. Especially, when the F-ratio value of the problem is smaller, the model would be harder to obtain the optimal solution. So, the development of a valid solution procedure to reduce the computational effort is needed. In the future research, the multi-model multi-manned assembly line balancing problems with simultaneous production perspective are recommended.

References

[1] Baybars, I., "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," *Management Science*, Vol. 32, pp. 909–932 (1986).

[2] Becker, C. and Scholl, A., "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, Vol. 168, pp. 694–715 (2006).

[3] Boysen, N., Fliedner, M. and Scholl, A., "A Classification of Assembly Line Balancing Problems," *European Journal of Operations Research*, Vol. 183, pp. 674–693 (2007).

[4] Scholl, A. and Becker, C., "State-of-the-Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing," *European Journal of Operations Research*, Vol. 168, pp. 666–693 (2006).

[5] Dimitriadis, S. G., "Assembly Line Balancing and Group Working: A Heuristic Procedure for Workers' Groups Operating on the same Product and Workstation," *Computers & Operations Research*, Vol. 33, pp. 2757–2774 (2006).

[6] Gokcen, H. and Erel, E., "Binary Integer Formulation for Mixed-Model Assembly Line Balancing Problem," *Computers and Industrial Engineering*, Vol. 34, pp. 451–461 (1998).

[7] Gutjahr, A. L. and Nemhauser, G. L., "An Algorithm for the Line Balancing Problem," *Management Science*, Vol. 11, pp. 308–315 (1964).

[8] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, WH Freeman, San Francisco (1979).

[9] Van Zante-de Fokkert, J. I. and de Kok, T. G., "The Mixed and Multi Model Line Balancing Problem: A Comparison," *European Journal of Operations Research*, Vol. 100, pp. 399–412 (1997).

[10] Thomopoulos, N. T., "Mixed Model Line Balancing with Smoothed Station Assignments," *Management Science*, Vol. 16, pp. 593–603 (1970).

[11] Macaskill, J. L. C., "Production-Line Balancing for Mixed-Model Lines," *Management Science*, Vol. 19, pp. 423–434 (1972).

[12] Dar-El, E. M. and Nadivi, A., "A Mixed-Model Sequencing Application," *International Journal of Production Research*, Vol. 19, pp. 69–84 (1981).

[13] Erel, E. and Gokcen, H., "Shortest-Route Formulation of Mixed-Model Assembly Line Balancing Problem," *European Journal of Operations Research*, Vol. 116, pp. 194–204 (1999).

[14] Bukchin, J., Dar-El, E. M. and Rubinovitz, J., "Mixed Model Assembly Line Design in a Make-to-Order Environment," *Computers and Industrial Engineering*, Vol. 41, pp. 405–421 (2002).

[15] Hoffmann, T. R., "Assembly Line Balancing with a Precedence Matrix," *Management Science*, Vol. 9, pp. 551–562 (1963).

[16] Dar-El, E. M., "MALB - A Heuristic Technique for Balancing Large Single-Model Assembly Lines," *AIIE Transactions*, Vol. 5, pp. 343–356 (1973).

Manuscript Received: Jun. 2, 2008
Accepted: Dec. 31, 2008