# Research on Balancing and Sequencing Problems of Flexible Mixed Model Assembly Lines with Alternative Precedence Relations

Yunfang Peng, Zhongyao Zhang, Yajuan Han, Beixin Xia

*School of Management, Shanghai University, Shanghai 200240, People's Republic of China*

To tackle the balancing and scheduling problems of flexible mixed model assembly lines with alternative precedence relations, If-then rules and AND/OR graphs are adopted as modelling tools to replace precedence graphs that have limitations in representing precedence relations. Constraint programming (CP) and mixed integer linear programming (MILP) models are established respectively, according to constraints among tasks represented by If-then rules and AND/OR graph. Computational experiments on varied scales of instances are carried out to test those models. The computational results reveal that If-then rules modelling achieves a better balancing and scheduling solution for the flexible mixed model assembly lines. Our empirical shows that CP model is simpler and faster than MILP.

**Keywords:** mixed-model assembly line; balancing and sequencing; If-then rules; alternative relations; AND/OR graphs; constraint programming;

## 1. Introduction

Multi-variety, small-volume, and customized production forms have become the mainstream mode for manufacturing companies, as customer demand for products changes from a single function to an integrated function with diverse and individualized characteristics. In modern industry, as a result of dealing with complex product environments, mixed-model assembly lines have proven to be a very effective type of assembly line.

In order to maximize the efficiency of mixed-model assembly lines, there are two significant problems that need to be solved: line balancing and model sequencing. Line balancing problem is the assignment of the tasks to workstations, and model sequencing considers the model operation sequence on the assembly line. Obviously, task assignment and model sequencing are interrelated sub-problems. The performance of balancing is affected by the sequence of models while the optimality of the model sequencing depends on the result of line balancing. At the same time, many constraints have to be considered in realistic situation, such as alternative precedence relations.

Generally, the precedence relationship among tasks is considered to be predetermined, but in reality, the assembly process of a product may have several alternative precedence relations. When the completion time of tasks depends on the processing sequence, various other subgraphs may appear, one of which must be selected. For example, many industrial products have multiple assembly plans such as the production of hand-held drills (Senin, Groppetti and Wallace 2002) and the sewing of clothes (Salum and Supciller, 2008). In the aforementioned literature, the major goal of the assembly line with alternative precedence relations is to balance it by the efficient assignment of tasks and route selection, and these two sub-problems must be solved simultaneously. When we combine alternative precedence constraint with the mixed-model assembly line, the mixed-model assembly line with alternative precedence relations emerges.

To the best of our knowledge, there is no research reported the balancing and sequencing of mixed-model assembly line with alternative precedence relations. In this paper, we solve task assignment problem, model sequencing problem and route selection problem simultaneously. The contribution of this paper may be summarized as follows:

(1) Different from the current literature that considers determined precedence relations in balancing and sequencing of mixed model assembly line, we consider alternative precedence relations.

(2) This paper applies the constraint programming (CP) to the balancing and sequencing problem of mixed-model assembly lines with alternative precedence relationships for the first time. And the performance of CP model is compared with MILP model.

(3) This paper summarizes the basic alternative precedence relations by If-then rules, which can extend to any complicated relations among tasks.

The rest of the paper is organized as follows. In section 2, we define the problem and introduce AND/OR graph and If-then rules. The MILP model and CP model proposed solving the balancing and sequencing mixed-model assembly line problem with alternative precedence relations are presented in Section 3 and Section 4, respectively. Section 5 illustrates a specific example to help understand the problem. Section 6 presents the computational problems and experimental comparisons. Finally, the concluding remarks and future research directions are provided in Section 7.

## 2. Literature review

Although assembly line balancing problem with alternative precedence relations and mixed-model assembly line balancing and sequencing problem have been thoroughly studied, no research has been reported where both problems are solved simultaneously. In this section, we present a brief literature review of assembly line with alternative precedence, a review on simultaneous balancing and sequencing of mixed-model assembly line and a review of a successful method, constraint programming (CP), which is applied to solve these problems.

The simple assembly line balancing problem (SALBP) is about how to assign tasks to stations, which is divided into two type problems by Baybars et al (1986): the simple assembly line balancing problem (SALBP) and the general assembly line balancing problem (GALBP). The characteristics of SABLP is well-known, such as task processing time is determined and assembly precedence is determined, in one word, the structure of problem is well-defined. In contrast, GALBP relaxes one or more assumptions of the SALBP, such as different line shapes (Sparling and Miltenburg 1998), space constraints (Hillier and Boling 1993) or parallel stations (Goekcen, Agpak and Benzer 2006), which has been summarized by Becker and Scholl (2006).

However, traditional assembly line assumes that all tasks processed in a predetermined relationship, which means no processing alternatives exist. However, a single precedence graph is usually not able to fully describe the problem especially when there are alternative precedence relations in the assembly line problem. In response to this drawback, Pinto et al. (1983) tried to combine assembly line balancing problem with the choice of manufacturing alternatives. After that, Capacho et al. (2005) developed a new GALBP with alternative subgraphs to overcome this as it considers processing alternatives within the precedence graph such that selecting the processes and assigning tasks to stations simultaneously. Then, Capacho and Pastor (2008) proposed a more rigorous definition and developed heuristic methods to solve this problem (Capacho and Pastor 2011). Scholl (2008) discuss a special case of ASALBP which considers sequence-dependent task times, then formulated an improved mixed integer program for the first time for ASLABP (Scholl 2009). Koc and Sabuncuoglu (2009) adopted the AND/OR graphs, which was proposed firstly by Homem (1990), to describe the alternative precedence and prove that the use of AND/OR graphs instead of a precedence graph can achieve better solution. However, the existing modelling tools are mostly based on traditional precedence graphs, rather than investigating more effective tools to solve the ALBP. Topaloglu and Salum (2012) adopted the If-then rules to describe the balancing problem with alternative precedence relations and show how to map a rule-based model to a CP model and an IP model and this mapping can’t be implemented by graph-based models. It’s shown that rule-based model achieves better results in solving route selection problem and balancing problem simultaneously. However, these papers only consider the simple assembly line balancing problem with alternative precedence relations and not consider the sequencing problem.

In mixed assembly line balancing and sequencing (MALB/S) problems, Previously, balancing and sequencing were solved hierarchically. This approach focuses on balancing the assembly line firstly, subsequently, the sequencing problem is solved under the constraint of the balancing result. This method has been adopted by, e.g., Sawik et al. (2002), Uddin et al. (2010) and was also applied with multi-objectives (Huang, Zhang and Zhou 2010; Faccio and Gamberi 2016).

However, since Kim et al. (2000a) pointed that these two problems of balancing and sequencing for mixed-model assembly lines are tightly interrelated with each other and proposed a co-evolutionary algorithm to solve MALB/S problem, the simultaneous approach results in better solutions compared to the hierarchical approach in terms of quality of solutions. The research on the simultaneous approach have drawn more attention. Battini et al. (2009) solved the MALB/S problem with finite buffer capacity and using simulation software to test it. Mosadegh, Zandieh and Ghomi (2012) developed a MILP model for MALB/S with and proposed a simple simulate algorithm to solve it. Then Saif et al. (2014) extended the multi-objective MALB/S by involving multiple objectives and proposed an artificial bee colony algorithm. Due to the diversity of assembly line layout, some scholars have studied the MALB/S under different assembly line types such as U-type assembly line (Kim, Kim and Kim 2000b; Kim et al. 2006; Kara, Ozcan and Peker 2007; Hamzadayi and Yildiz 2013), two-sided assembly line (Nilakantan, Li and Tang 2017), parallel assembly line (Ozcan et al. 2010) and parallel two-sided assembly line (Kucukkoc and Zhang 2014; Kucukkoc and Zhang 2016). In the follow-up, some researchers added more problems or constraints to MALB/S, which extended the scope of this problem. Li et al. (2017) studied the balancing and sequencing problems of mixed model assembly lines with robotic distribution which considered three subproblem simultaneously. Defersha and Fantahun (2018) proposed a hybrid genetic algorithm to simultaneously solve balancing, sequencing and workstation planning problems.

As a very effective method for solving combinatorial optimization problems, constraint programming (CP) is especially successful in the field of scheduling. Bockmayr and Pisaruk (2001) firstly solved simple assembly line balancing problem with a CP approach. They present a combined MIP and CP for type-I single model lines. Pastor et. al (2007) expanded on this basis, and compared the performance of CP and IP in simple assembly line balancing problem through the computational experiment. Result of these experiments shows good performance of CP compared to MIP for SALBP-1. Then CP was compared with branching heuristics for SALBP-2 (Deville et al. 2008; Schaus et al. 2009), and still got good result. Topaloglu and Salum (2012) pointed that due to the modelling capabilities of CP, it is much easier to represent the rule-based model. Öztürk et. al (2013) proposed a CP model for flexible MALB/S problem and found that the CP model is significantly better than the MIP model in solving speed and problem scale. Buckchin and Raviv (2018) used the CP approach to summarize and compare on various assembly line balancing problems. The result show that CP model outperforms MILP for medium to large problem instances. Compared with the heuristic algorithm, in the example of 1000 tasks, SALOME provides better performance, however, CP can provide good close to optimal solutions for problems with complex parameters.

From the current literature review, no research has proposed the balancing and sequencing of mixed-model assembly line problem with alternative precedence relations. Furthermore, constraint programming (CP) attracts more and more attention in assembly line field, which has been proved to be an efficient method. Therefore, this paper formulates a CP model for the first time and its performance is compared with a MILP model.

## 2. Problem definition

In a mixed-model assembly line with alternative precedence relationships, multiple tasks are assigned to each workstation for processing. Different products, or models, are assembled in a certain proportion and order. And each product (or model) can be assembled in different precedence relationship, this is the alternative precedence relationship. This article includes three sub-problems: task assignment, product sequencing, and precedence selection. This section presents the basic assumptions based on Öztürk (2010).

(1) Each station can perform at most one task at any given time.

(2) A product can be assembled by multi-route, but, only one route can be selected for processing finally.

(3) Each task type can be assigned to more than one station.

(4) The total space required for the tasks assigned to each station must not exceed the station's finite work space available.

(5) The revisiting of stations is not allowed.

(6) The objective is to minimize the latest completion time of the last job in the assembly line.

On the mixed model assembly line, there are assembly sequence constraints between the operation processes that move unidirectionally along the line. The processing of the next task can be performed only if all the predecessor activities of a certain task are completed. Generally, a precedence graph is used to represent the constraint of assembly order between the tasks. As shown in Figure 1 (a), the circle represents tasks, the numbers in the circles denote the serial number, the arrows indicate the sequence of processes, and the number above the circle is the processing time of stations. However, when the assembly line is flexible, such as the precedence between tasks is optional, a single precedence graph cannot fully describe the problem. Hence, Koc and Sabuncuoglu (2009) using an AND/OR graph (AOG), such as the logical statement 'If (C1 and C2) or C3 or C4, then C5', instead of a precedence diagram to represent the process sequence. As shown in Figure 1 (a), arcs are signs that visually distinguish between AND relationships. However, as the scale of the problem expands, the entanglement of a large number of hyperarcs in AOG will affect the solution efficiency of the MIP model.

*<<<<<Insert Figure 1 (a) and (b) about here>>>>>*

In order to avoid that the precedence graph cannot describe alternative precedence relationship and the hyperarcs in AOG are entangled with each other. We adopt the If-then rules to describe the constraints. One of the contributions of this paper is to sort out the basic constraints of the task sequence in the balancing and sequencing problems and describe them separately through the If-then rules, the precedence graph, and the AOG. Table 1, which is a summary language description table of basic precedence, shows that the If- then rules has the advantage of being descriptive, concise, and efficient for the description of the constraints for assembly lines when compared to precedence graph and AOG.

*<<<<<Insert Table 1 about here>>>>>*

## 3. MILP model formulation

The traditional method which using the precedence graph as the modelling tool to solve the balancing and sequencing problems of assembly line needs to derive the precedence graph before performing mathematical model. However, as the scale of the problem increases, the precedence graph is difficult to derive, and cannot describe the compatibility relationship between the assembly processes. Therefore, it is a better way to adopt the AOG and the If-then rules as the modelling tools instead of the precedence graph when studying balancing and sequencing problems of assembly lines.

### *3.1. Parameters and variables*

The notations utilized in this section are below:

**Parameters:**

| | |
|---|---|
| $t$ | index for task |
| $s$ | index for station |
| $p$ | index for model |
| $r$ | index for route |
| $j$ | designed (task, product) pairs indicate which product requires which task |
| $T$ | the set of all tasks, $t \in T = \{1, 2, \cdots, \mathrm{NT}\}$. |
| $S$ | the set of all products, $s \in S = \{1, 2, \cdots, \mathrm{NS}\}$. |

| | |
|---|---|
| $P$ | the set of all products, $p \in P = \{1, 2, \cdots, \mathrm{NP}\}$. |
| $R$ | the set of all routes, $r \in R = \{1, 2, \cdots, \mathrm{NR}\}$. |
| $J$ | the set of all jobs, where *j.task* and *j.product* refer to the corresponding task and product of job *j* respectively. |
| $M$ | a large enough integer, $M = \sum_{j \in J} d_{sj}$. |
| $\mathrm{P}_{jr}$ | the predecessor activity of job *j* on route *r*. |
| $S(j)$ | the set of all stations for the product of job *j*. |
| $S(t)$ | the set of all stations for the product of process *t*. |
| $a_{st}$ | production capacity requirement of task *t* on station *s*. |
| $b_s$ | total production capacity of station *s*. |
| $d_{sj}$ | assembly processing duration for job *j* on station *s*. |

**Decision variable:**

| | |
|---|---|
| $c_{srj}$ | the completion time of the job *j* on the station *s* on path *r*. |
| $A_{spr}$ | the arrival time of the product *p* on the station *s* on path *r*. |
| $D_{spr}$ | the departure time of the product *p* on the station *s* on path *r*. |
| $C_{max}$ | latest completion time. |
| $x_{sjr}$ | 1 if job j is assigned to station s in route r; or 0. |
| $y_{str}$ | 1 if task t is assigned to station s in route r; or 0. |
| $u_{vpr}$ | 1 if product p is the v-th product in route r; or 0. |
| $z_{sjkr}$ | 1 if job j is assigned to station s with route r; or 0. |

### *3.2. Description of MIP model*

The descriptions of the MIP model are as following:

$$\text{Min} \quad C_{max} \tag{1}$$

subject to:

$$\sum_{s \in S(j)} x_{sjr} = 1, \forall\, j \in J, r \in R \tag{2}$$

$$\sum_{s \in S(j)} x_{s1r} = \sum_{s \in S(j)} x_{wjr}\,, \forall\, j, w \in J, r \in R \tag{3}$$

$$c_{sjr} \leq M x_{sjr}, \forall\, j \in J, s \in S, r \in R \tag{4}$$

$$c_{shr} - c_{sjr} - d_{sh} x_{shr} + M(1 - z_{sjhr}) \geq 0, \forall\, j, h \in J, s \in S(j) \cap S(h) | j \neq h, r \in R \tag{5}$$

$$x_{sjr} + x_{shr} - 2\left(z_{sjhr} + z_{shjr}\right) \geq 0, \forall\, j, h \in J, s \in S(j) \cap S(h) | j \neq h, r \in R \tag{6}$$

$$x_{sjr} + x_{shr} - z_{sjhr} - z_{shjr} - 1 \leq 0, \forall\, j, h \in J, s \in S(j) \cap S(h) | j \neq h, r \in R \tag{7}$$

$$c_{whr} - c_{sjr} - d_{wh} + M(1 - x_{whr}) \geq 0, \forall\, j \in P(h), s \in S(j), w \in S(h), r \in R \tag{8}$$

$$\sum_{s \in S(j)} j x_{sjh} \leq \sum_{w \in S(h)} h x_{whr}, \forall\, j \in P_{hr}, r \in R \tag{9}$$

$$\sum_{t \in T(t)} a_{st} y_{str} \leq b_s, \forall\, s \in S, r \in R \tag{10}$$

$$y_{str} = 0, \forall\, t \in T, s \notin S(t), r \in R \tag{11}$$

$$\sum_{p \in P} u_{vpr} = 1, \forall\, v \in P, r \in R \tag{12}$$

$$\sum_{v \in P} u_{vpr} = 1, \forall\, p \in P, r \in R \tag{13}$$

$$D_{spr} - A_{sqr} - M\left(2 - u_{v-1,p,r} - u_{v,p,r}\right) \leq 0, \forall\, p, q, v \in P, s \in S | p \neq q, v > 1, r \in R \tag{14}$$

$$A_{s,p,r} \geq D_{s-1,p,r}, \forall\, p \in P, s \in S | s > 1, r \in R \tag{15}$$

$$x_{sjr} \leq y_{s,j.task,r}, \forall\, j \in J, s \in S(j), r \in R \tag{16}$$

$$y_{str} \leq \sum_{j \in J | j.task = t} x_{sjr} \quad \forall\, t \in T, s \in S, r \in R \tag{17}$$

$$A_{s,j.product,r} - c_{sjr} + d_{s,j.task} x_{sjr} - M\left(1 - x_{sjr}\right) \leq 0, \forall\, j \in J, s \in S(j), r \in R \tag{18}$$

$$D_{s,j.product,r} - c_{sjr} \geq 0, \forall\, p \in P, s \in S, r \in R \tag{19}$$

$$D_{s,p,r} - A_{s,p,r} - \sum_{j \in J | j.product = p} d_{s,j.task} x_{sjr} \geq 0, \forall\, p \in P, s \in S, r \in R \tag{20}$$

$$D_{spr} \leq C_{max}, \forall\, p \in P, s \in S, r \in R \tag{21}$$

$$c_{sjr} \geq 0, x_{sjr} \in \{0,1\}, \forall\, j \in J, s \in S(j), r \in R \tag{22}$$

$$\mathrm{z_{sjhr}} \in \{0,1\}, \forall\, j, h \in J, s \in S, r \in R \tag{23}$$

$$y_{str} \in \{0,1\}, \forall\, t \in T, s \in S, r \in R \tag{24}$$

$$u_{vpr} \in \{0,1\}, \forall\, p, v \in P, r \in R \tag{25}$$

$$A_{spr} \geq 0, \forall\, p \in P, s \in S, r \in R \tag{26}$$

Objective function (1) minimizes the latest completion time. Constraint (2) indicates that each task is assigned to exactly one station. Constraints (2) and (3) guarantee that jobs are assigned to the same route. Constraint (4) states that the task can be processed only if it is assigned to eligible stations. Constraints (5) - (7), where *j* and *h* indicate different jobs, represent that two distinct jobs cannot be processed at the same station simultaneously. Constraint (8) defines the precedence relationship for different jobs, while Constraint (9) avoids rework of the same job on the same station. Constraint (10) ensures that the cumulative production capacity requirements for task within the station do not exceed the total production capacity. Constraint (11) prevents the task from being assigned to a noneligible station. Constraints (12) and (13) ensure that the same product is placed in the same location at the station when sorted at different stations. For different products on the same station, Constraint (14) avoids the overlap between the current product and the next product, *p*, *q* represents a different product respectively. Constraint (15) prevents the processing time of the same product from overlapping on different stations, and Constraint (16) indicates that the task corresponding to the job is assigned to the eligible station. Constraint (17) means that the job can be assigned to the station when at least one task corresponding to the task is assigned to the station. Constraints (18) and (19) represent the relationship between the job completion time and the product arrival time and departure time, respectively. Constraint (20) channel product arrival and departure time by enforcing that departure time of any product in any station is later than arrival time of that product to that station and the total assembly time on that station. Constraint (21) determines the scope of the scheduled completion time. Constraints (22) - (26) give domains of decision variables.

## 4. CP model formulation

In the AOG, the number of hyperarcs is often more than the number of tasks, and the hyperarcs are intertwined, causing the complex constraints in the MIP model. As the scale of the problem increases, the derivation of the AND/OR map becomes difficult, which not only affects the modelling efficiency of the MIP model but also affects the solution efficiency of the MIP model. However, the maximum number of rules is the number of tasks when using the If-then rules, and the expansion of the problem does not cause the expansion of the CP model. Moreover, the CP model can succinctly describe logical constraints through logical symbols (such as '∩', '∪', '→', etc.), avoiding complicated constraints.

As an effective way to solve scheduling problems, the CP model is usually solved using ILOG CP Optimizer which is a combinatorial optimization software. Before introducing the CP model, some basic variables that apply to the ILOG Scheduler environment, and the special constraints built on these variables that can effectively improve the efficiency of the CP model in ILOG software are introduced (Laborie and Rogerie 2018).

### 4.1. Constraint introduction

#### *4.1.1. Interval variable*

The interval variable represents the time interval of the tasks that need to be scheduled (a task or an activity is carried out, a resource is being used). The variable contains a starting value, an ending value, and an interval. An important feature of interval variables is they can be optional. There are some constraints for interval variables in this article:

(1) $endOf(a)$ represents the end time of the interval variable a.

(2) $endBeforeStart(a, b)$ represents the end time of a is larger than the start time of b which indicates that the interval variable b can be processed only if the processing of the interval variable a has been completed.

(3) $sizeOf(a)$ represents the scale of the interval variable a.

(4) $presenceOf(a)$ indicates whether the interval variable a exists or not.

(5) '=>' indicates that if the constraint before the symbol is satisfied when the constraint after the symbol is satisfied

#### *4.1.2 Sequence variable*

Sequence variables are defined by a set of interval variables whose values denote the ordering of tasks in the set. In general, sequence variables do not impose constraints on the position of interval variables, but their positions can be constrained by constraints.

(1) $noOverlap(\{a_1, \dots, a_n\})$ represents a series of interval variables a that does not overlap each other.

(2) $span(a, \{b_1, \dots, b_n\})$ indicates that the interval variable a start from $b_1$ and ends with $b_n$ which are in the set $\{b_1, \dots, b_n\}$.

### 4.2. Decision Variables

$x_j$ the station where task *j* is located.

$\alpha_{sj}$ the task *j* arranged in the station *s*.

$\beta_{sp}$ the product *p* arranged in the station *s*.

$P(r)$ the predecessor activity of task *r*

### 4.3 Description of the CP model

The descriptions of the CP model are as following:

$$\text{Min} \quad C_{max} \tag{27}$$

Subject to:

$$x_j = s \Leftrightarrow presenceOf(\alpha_{sj}) = 1, \forall\, j \in J, s \in S \tag{28}$$

$$noOverlap(\alpha_{sj}), \forall j \in J, s \in S \tag{29}$$

$$endBeforeStart(\alpha_{sj}, \alpha_{wr}), \forall\, j \in P(r), s, w \in S \tag{30}$$

$$\mathrm{x}_j \le \mathrm{x}_r \cap \left(endOf(\alpha_{sj}) \le startOf(\alpha_{sr})\right), \forall\, j \in P(r) \tag{31}$$

$$(\mathrm{x}_j \le \mathrm{x}_r \cap (endOf(\alpha_{sj}) \le startOf(\alpha_{sr}))) \cup (\mathrm{x}_h \le \mathrm{x}_r \cap (endOf(\alpha_{sh}) \le startOf(\alpha_{sr}))), \forall\, j \in P(r) \text{ or } h \in P(r), s \in S \tag{32}$$

$$(\mathrm{x}_j \le \mathrm{x}_r \cap (endOf(\alpha_{sj}) \le startOf(\alpha_{sr}))) \cup (\mathrm{x}_h \le \mathrm{x}_r \cap (endOf(\alpha_{sh}) \le startOf(\alpha_{sr}))), \forall\, j \in P(r) \text{ and } h \in P(r), s \in S \tag{33}$$

$$\begin{gathered}\left(x_j \le x_r \cap \left(endOf(\alpha_{sj}) \le startOf(\alpha_{sr})\right)\right) \cup \\ (\left(x_h \le x_r \cap \left(endOf(\alpha_{sh}) \le startOf(\alpha_{sr})\right)\right) \cap \\ \left(x_k \le x_r \cap \left(endOf(\alpha_{sk}) \le startOf(\alpha_{sr})\right)\right)) \\ , \forall\, j \in P(r) \text{ or } \left(h \in P(r) \text{ and } k \in P(r)\right), s \in S\end{gathered} \tag{34}$$

$$(\left(x_j \leq x_r \cap \left(endOf(\alpha_{sj}) \leq startOf(\alpha_{sr})\right)\right) \cap$$
$$\left(x_h \leq x_r \cap \left(endOf(\alpha_{sh}) \leq startOf(\alpha_{sr})\right)\right)) \cup$$
$$(\left(x_g \leq x_r \cap \left(endOf(\alpha_{sg}) \leq startOf(\alpha_{sr})\right)\right) \cap$$
$$\left(x_k \leq x_r \cap \left(endOf(\alpha_{sk}) \leq startOf(\alpha_{sr})\right)\right))$$
$$, \forall \left(j \in P(r) \ and \ h \in P(r)\right) \ or \ \left(g \in P(r) \ and \ k \in P(r)\right), s \in S \tag{35}$$

$$\sum_{t \in T(t)} a_{st} y_{st} \leq b_s, \forall \ s \in S \tag{36}$$

$$y_{st} = 0, \forall \ t \in T, s \notin S(t) \tag{37}$$

$$span\left(\beta_{sp}, all(j \in J : j.product = p)\alpha_{sj}\right), \forall \ p \in P, s \in S \tag{38}$$

$$endBeforeStart\left(\beta_{s,p}, \beta_{wp}\right), \forall \ p \in P, s, w \in S | s < w \tag{39}$$

$$noOverlap\left(\beta_{sp}\right), \forall p \ \in P, s \in S \tag{40}$$

$$x_j = s \Rightarrow y_{x_j, j.task} = 1, \forall \ j \in J, s \in S \tag{41}$$

$$endOf\left(\beta_{sp}\right) \leq C_{max}, \forall \ p \in P, s \in S \tag{42}$$

$$y_{st} \in \{0,1\}, \forall \ t \in T, s \in S \tag{43}$$

Objective function (27) minimizes the latest completion time. Constraint (28) indicates that a job has a processing time when it is assigned to a station, which is $\left(x_j = \mathrm{s}\right) \Rightarrow sizeOf\left(\alpha_{sj}\right) = d_{sj}, \forall \ j \in J, s \in S(j)$. Constraint (29) guarantees that different tasks on the same station are independent. Constraint (30) states the precedence relationships between different tasks. Constraints (31) – (35) denotes different precedence relationships with If-then rules, respectively. In detail, Constraint (31) corresponds to If $J_j$ then $J_r$. Constraint (32) corresponds to If $J_j$ OR $J_h$ then $J_r$. Constraint (33) corresponds to If $J_j$ AND $J_h$ then $J_r$.

Constraints (34) and (35) correspond to If $J_j$ OR ($J_h$ AND $J_k$) then $J_r$ and If ($J_j$ AND $J_h$) OR($J_g$ AND $J_k$) then $J_r$, respectively. Constraint (36) ensures that the cumulative production capacity requirements for tasks on the station within the total production capacity. Constraint (37) prevents the task from being assigned to a noneligible station. Constraint (38) states that product when it is present spans over all tasks. Constraint (39) means that the product must remain at the current station before the next station is opened. Constraint (40) prevents the different products from overlapping on the same station, and Constraint (41) means that the task can be assigned to the station only when one or more tasks corresponding to the process are assigned to the station. Constraint (42) defines the scope of the scheduled completion time, and Constraint (43) gives the definitions of decision variables.

## 5. Illustrative example

Take the assembly of Giant for the CitySpeed FCR, Expedition 1 Travel Mountain Bike and CityStorm City Commuter (referred to as Product 1, Product 2, and Product 3) as an example, and Table 2 gives the details of tasks. The Precedence relationship indicates the processing order constraint corresponding to the task. For example, task 1 and task 2 are not subject to any constraint, the task 6 can be processed under the conditions that the tasks 2, 4, and 5 are completed, and the tasks 3 and 5 include two alternative predecessor activities respectively. Time means the processing time of the task. Production capacity refers to the production space required to process a task, such as the completion of process 3 requires 2 units of production space. In Table 3, $\delta$ indicates that processing is required. For example, the assembly of product 1 needs to complete the processes 1, 2, 3, 4, and 6. Besides, '-' means no processing is required. For example, product 2 does not require task 3 as the accessories of product 2 do not contain fender. A process consists of a task, such as a task '5, 2' indicating the fifth operation of product 2. Due to limitations of resources such as technicians, tools, production capacity, etc., multiple repair stations need to be coordinated to complete the tasks shown in Table 3. That is, each station on the assembly line can only be equipped with a certain number of tasks and the details of production capacity are shown in Table 4. The total production capacities of stations 1, 2, and 3 in Table 4 are 8, respectively, which means that each station provides up to 8 units of production space. Moreover, '-' means that the task does not need to be processed at the corresponding workstation.

*<<<<<Insert Table 2 about here>>>>>*

*<<<<<Insert Table 3 about here>>>>>*

*<<<<<Insert Table 4 about here>>>>>*

In the CP model, the precedence relationship of the task is mapped to the If-then rules, so the number of rules in the model is consistent with the number of tasks with constraint relationships. The If-then rules obtained by mapping from Table 2 and 3 are shown in Table 5. It can be seen that the CP model contain 10 rules. Combined with Table 2 and 3, four tasks with optional predecessor activity can be used to derive $2^4$=16 precedence graphs, that is, a single precedence graph cannot fully describe the compatibility relationship between assembly tasks. The AOG derived from Table 2 and 3 is shown in Figure 2. Lines 3 and 5 of Table 2 is optional coexistence relationships. The AOG can be represented by two sets of sub-graphs with OR relationships.

For the example problem, ILOG software is used to solve the CP and MIP models. The operation results are shown in Table 6 and the details of scheduling are shown in Figure 3. It can be seen that, in the case of satisfying the optimal scheduling completion time of 22 minutes, the CP model is significantly faster than MIP model as the calculation time of the CP model and the MIP model are 1.36 seconds and 42.75 seconds, respectively. The number of variables included in the model and the constraint of the CP model and MIP model in Table 6 is (94, 354) and (7985, 26023), respectively. Thus, in terms of the model size, the number of variables included in the model and the constraint of the CP model is significantly less than the MIP model, indicating that the CP model is smaller. A comparison of the performance of the CP and MIP models is analysed in the following sections.

*<<<<<Insert Table 5 about here>>>>>*

*<<<<<Insert Table 6 about here>>>>>*

*<<<<<Insert Figure 2 about here>>>>>*

*<<<<<Insert Figure 3 about here>>>>>*

## 6. Comparison and discussion

The scale of the number of jobs is the main factor affecting the scale of the scheduling problem, and the number of tasks and products determines the scale of the number of jobs, so the consideration of the number of tasks and products is the first, and the consideration of the number of workstations is the second. In order to compare the performance of CP and MIP models, we design some test cases based on the above factors. The number of tasks in the cases may be 5, 10 or 15. The number of products may be 3, 4 or 5. And the number of the station may be 3, 4, or 5. Therefore, the number of cases is 3 × 3 × 3 = 27. In this paper, the examples are divided into three categories based on the number of processes: small scale (5 tasks), medium scale (10 tasks), and large scale (15 tasks). The cases are in the appendix.

This article uses ILOG software to solve the CP and MIP models. All programs and data are encoded in OPL language and run on a computer configured as Intel® Xeon Platinum 8268 processor, 2.90GHz, 96GB RAM. The cases retain the original precedence relationship of the tasks and add three type optional relationships which called 1OR, 2ORS, and 3ORS. 1OR means that one task of a product has two optional precedence relationships, 2ORS means that two products each contain one task with two optional precedence relationship, and 3ORS means that each of the three products contains a task with two optional precedence relationships. Table 7 and 8 lists the calculation results of the studies in the case of maximum operation time are 3600 seconds. In Tables 7 and 8, the first column is the type of cases. The second, third, and fourth columns correspond to the number of tasks, the number of products, and the number of stations in the cases. The fifth column is the number of tasks which are generated by the combination of the process and product. Columns 6 to 9 are the number of variables, the number of constraints, the target value, and the operation time of solution results and evaluation indicators of the original problem. Columns 10 through 13, columns 14 through 17, and columns 18 through 21 represent the results of adding optional relationships for the 1OR, 2ORS, and 3ORS types, respectively.

Regarding the solution effect of the model, the data in Table 7 shows that the cases with changes have achieved the best results as the number of selectable relationships increases. For example, new cases which combine 1OR, 2ORS, and 3ORS with the 15-3-3 case in Table 7 have achieved the same optimization results as the original problem. In addition, new cases with increased optional relationships can even yield better results than the original case. For example, the optimal target value of the 5-5-3 original problem in Table 7 is 28 minutes and is reduced to 24 minutes after adding the 2ORS and 3ORS type optional relationships. The situation has been verified in Table 8.

Regarding the efficiency of the model, it can be found that the average operation time of the CP model is lower than the MIP model regardless of whether the task has an optional priority relationship, through comparing the average operation time of the two models in Table 7 and Table 8 on various problems. For example, to solve the small-scale original problem, the average operation time of the CP model is 1.24 seconds, while the MIP model is 3.25 seconds. As the scale of the problem increases, the advantage of the CP model in computing efficiency becomes more apparent. Unlike the CP model, the operation time of the MIP model explodes exponentially, and it is impossible to find the optimal solution within the specified time. For example, the CP model solves the 10-5-5 original problem with an operation time of 13.70 seconds, while the MIP model uses 3504.26 seconds. In addition, when solving the 15-4-5 case with the 1OR, the operation time of the CP model is 46.88 seconds, while the MIP model cannot find the optimal solution within the specified time.

Regarding the solution scale of the model, compare the number of variables and the number of constraints included in the operations of two models in Table 7 and 8. The size of the CP model is smaller than the MIP model. In terms of absolute size, when solving problems, the CP model is less than the MIP model, regardless of the number of variables or the number of constraints. For example, when solving the 5-5-5 original problem, the number of variables and the number of constraints to be searched by the CP model are 193 and 705, respectively, while the MIP model is 1750 and 5920. In addition, when solving the 15-3-5 problem with the addition of the 3ORS, the MIP model cannot even find the scheduling time because the solution size is too large and exceeds the computer memory, and is represented by '-'. In terms of scale change, as the scale of the problem increases, the growth rate of the CP model in terms of the number of variables and the number of constraints is smaller than the MIP model, and the number of variables grows significantly faster than the number of constraints; For example, when the original problem extended from 10-3-3 to 10-5-5, the number of variables and the number of constraints of the CP model increased from 134, 387 to 314 and 1214 which has increased by 2.34 and 3.14 times, while the MIP model has increased by 4.24 and 4.42 times.

<<<<<*Insert Table 7 about here*>>>>>

<<<<<*Insert Table 8 about here*>>>>>

## 7. Conclusion and further research

Today's competitive market requires more flexible production systems to respond quickly to changes in market conditions. Therefore, the balancing of assembly line is regarded as a tactical problem and become an optimal problem. As stated in the statement, the workload of the station is affected by the sequencing of mixed model assembly line. Therefore, in order to carry out effective production line management, task assignment and model sequence must be considered simultaneously. And the alternative precedence as a constraint often encountered in reality, although it can effectively improve the flexibility of the assembly line, but also increases the complexity of the model. As mentioned earlier, in the survey of related literature, there are a scarce number of papers dealing with this topical problem.

This paper has examined balancing and sequencing simultaneously in mixed model assembly line with alternative precedence to minimize the last completion time. One of the most important contributions of this paper is considering alternative precedence diagrams. Although all aspects of the studied problem are completely contained in a MIP model, it is found that the applicability of the model in real-world problems is computationally inefficient. Therefore, to solve this problem, we build a CP model which has proved their strength compare to mixed integer programming for solving simultaneous balancing and sequencing of flexible mixed model assembly lines with alternative precedence. Furthermore, we evaluate the performance of the proposed CP model and computational experiments indicate that the CP model outperforms MIP model over all sizes of test cases.

In a future research, we can consider extending the alternative precedence to other problems in the field of assembly lines, such as two-sided assembly lines or parallel assembly lines. We can also solve more problems together, such as the allocation of robots, workstation planning. Or, minimize the last completion time can be modified to other goals or expanded to the multi-objective problem. Although the CP model shows better performance than the MIP model, it is still difficult to meet the demand when solving larger-scale problems, such as more complicated alternative precedence. Therefore, designing more effective constraints or combining neighborhood search algorithms may also improve the performance of the solution.

<<<<<*Insert Appendix after references*>>>>>

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**References**

Battini, D., M. Faccio, A., Persona, and F. Sgarbossa. 2009. “Balancing–sequencing procedure for a mixed model assembly system in case of finite buffer capacity.” *The International Journal of Advanced Manufacturing Technology* 44 (3-4): 345-359. doi:10.1007/s00170-008-1823-8.

Baybars, I. 1986. "A survey of exact algorithms for the simple assembly line balancing problem." *Management science* 32 (8): 909-932. doi:10.1287/mnsc.32.8.909.

Becker, C., and A. Scholl. 2006. "A survey on problems and methods in generalized assembly line balancing." *European journal of operational research* 168 (3): 694-715. doi:10.1016/j.ejor.2004.07.023.

Bockmayr, A., and N. Pisaruk. 2001. "Solving assembly line balancing problems by combining IP and CP." *arXiv preprint* Article Number: cs/0106002.

Bukchin, Y., and T. Raviv. 2018. "Constraint programming for solving various assembly line balancing problems." *Omega* 78: 57-68. doi:10.1016/j.omega.2017.06.008.

Capacho, L., and R. Pastor. 2006. "The ASALB problem with processing alternatives involving different tasks: Definition, formalization and resolution." *Lecture Notes in Computer Science* 3982: 554–563. doi:10.1007/11751595_59.

Capacho, L., and R. Pastor. 2008. "ASALBP: The alternative subgraphs assembly line balancing problem." *International Journal of Production Research* 46 (13): 3503-3516. doi:10.1080/00207540701197010.

Capacho, L., R., Pastor, A., Dolgui, and O. Gunshinskaya. 2009. "An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem." *Journal of Heuristics* 15: 109–132. doi:10.1007/s10732-007-9063-x.

Capacho, L., and R., Pastor, 2011. "A Metaheuristic Approach to Solve the Alternative Subgraphs Assembly Line Balancing Problem." *Assembly Line: Theory and Practice* 37.

Defersha, F. M., and F. Mohebalizadehgashti. 2018. "Simultaneous balancing, sequencing, and workstation planning for a mixed model manual assembly line using hybrid genetic algorithm." *Computers & Industrial Engineering* 119: 370-387. doi:10.1016/j.cie.2018.04.014.

De Mello, Luiz S. Homem, and A. C. Sanderson. 1990. "AND/OR graph representation of assembly plans.". *IEEE Transactions on Robotics & Automation* 6 (2): 188-199. doi:10.1109/70.54734.

Faccio, M., M. Gamberi, and M. Bortolini. 2016. "Hierarchical approach for paced mixed-model assembly line balancing and sequencing with jolly operators." *International journal of production research* 54 (3): 761-777. doi:10.1080/00207543.2015.1059965.

Goekcen, H., K., Agpak, and R. Benzer. 2006. "Balancing of parallel assembly lines." *International Journal of Production Economics* 103 (2): 600-609. doi:10.1016/j.ijpe.2005.12.001.

Hamzadayi, A., and G. Yildiz. 2013. “A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines.” *Computers & Industrial Engineering* 66 (4): 1070-1084. doi:10.1016/j.cie.2013.08.008.

Haq, A. N., Rengarajan, K., and J. Jayaprakash. 2006. “A hybrid genetic algorithm approach to mixed-model assembly line balancing.” *The International Journal of Advanced Manufacturing Technology* 28 (3-4): 337-341. doi:10.1007/s00170-004-2373-3.

Huang, W. P., Z., Zhang, and Z. L. Zhou. 2010. “A Beam Search Approach to Stochastic Mixed-Model Assembly Line Balancing Problem.” *Operations Research and Management Science* 19 (6): 20-26.

Hillier, F. S., and S. R. W. Boling. 1993. “Toward characterizing the optimal allocation of storage space in production line systems with variable processing times.” *Management Science* 39 (1): 126-133. doi:10.5555/157303.157315.

Hwang, R., and H. Katayama. 2010. “Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach.” *International Journal of Production Research* 48 (21): 6417-6441. doi:10.1080/00207540903289755.

Kara, Y., U., Ozcan. and A. Peker. 2007. “Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives.” *Applied Mathematics and Computation* 184 (2): 566-588. doi:10.1016/j.amc.2006.05.185.

Kim, Y. K., S. J. K., and J. Y. Kim. 2000a. “Balancing and sequencing mixed-model U-lines with a co-evolutionary algorithm.” *Production Planning & Control* 11 (8): 754-764. doi:10.1080/095372800750038355.

Kim, Y. K., J. Y., Kim, and Y. Kim. 2000b. “A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines.” *Applied intelligence* 13 (3): 247-258. doi:10.1023/A:1026568011013.

Koc, A., I., Sabuncuoglu, and E. Erel. 2009. “Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph.” *Iie Transactions* 41 (10): 866-881. doi:10.1080/07408170802510390.

Kucukkoc, I., and D. Z. Zhang. 2014. “Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines.” *International Journal of Production Economics* 158: 314-333. doi:10.1016/j.ijpe.2014.08.010.

Laborie, P., J., Rogerie, P., Shaw, and P. Vilim. 2018. “IBM ILOG CP optimizer for scheduling.” *Constraints* 23 (2): 210-250. doi:10.1007/s10601-018-9281-x.

Li, Z., Janardhanan, M. N., Q., Tang, and P. Nielsen. 2018. "Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line." *Engineering Optimization* 50 (5): 877-893. doi:10.1080/0305215X.2017.1351963.

Liu, Q., Wang, W. X., Zhu, K. R., C. Y., Zhang, and Y. Q. Rao. 2014. "Advanced scatter search approach and its application in a sequencing problem of mixed-model assembly lines in a case company." *Engineering Optimization* 46 (11): 1485-1500. doi:10.1080/0305215X.2013.846334.

Mosadegh, H., S. F., Ghomi, and M. Zandieh. 2012. "Simultaneous solving of balancing and sequencing problems in mixed-model assembly line systems." *International Journal of Production Research* 50 (18): 4994-5016. doi:10.1016/j.asoc.2011.11.027.

Monteiro F., B., I., Albuquerque, and F. L. Neto. 2017. "Simultaneously solving mixed model assembly line balancing and sequencing problems with FSS algorithm." *arXiv preprint* Article Number: 1707.06185.

Nilakantan, J. M., Z., Li, Q., Tang, and P. Nielsen. 2017. "MILP models and metaheuristic for balancing and sequencing of mixed-model two-sided assembly lines." *European Journal of Industrial Engineering* 11 (3): 353-379. doi:10.1504/EJIE.2017.084880.

Öztürk, C., S., B., T., Hnich, and M. A. Örnek. 2013. "Balancing and scheduling of flexible mixed model assembly lines." *Constraints* 18 (3): 434-469. doi:10.1007/s10601-013-9142-6.

Özcan, U., H., Çerçioğlu, H., Gökçen, and B. Toklu. 2010. "Balancing and sequencing of parallel mixed-model assembly lines." *International Journal of Production Research* 48 (17): 5089-5113. doi:10.1080/00207540903055735.

Salum, L., and A. Supciller. 2008. "Rule-Based modeling of assembly constraints for line balancing." *In International Conference on Intelligent Computing* 783-789. doi:10.1007/978-3-540-85984-0_94.

Saif, U., Guan, Z., W., B., Liu, Wang, and C. Zhang. 2014. "Multi-objective artificial bee colony algorithm for simultaneous sequencing and balancing of mixed model assembly line." *The International Journal of Advanced Manufacturing Technology* 75 (9-12): 1809-1827. doi:10.1007/s00170-014-6153-4.

Sawik, T., Tadeusz. 2002. "Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line." *European Journal of Operational Research* 143 (1): 115-124. doi:10.1016/S0377-2217(01)00328-9.

Scholl, A., N., Boysen, and M. Fliedner. 2008. “The sequence-dependent assembly line balancing problem.” *OR Spectrum* 30 (3): 579-609. doi:10.1007/s00291-006-0070-3.

Scholl, A., N., Boysen, and M. Fliedner. 2009. “Optimally solving the alternative subgraphs assembly line balancing problem.” *Annals of Operations Research* 172 (1): 243. doi:10.1007/s10479-009-0578-4.

Senin, N., R., Groppetti, and D. R. Wallac. 2000. “Concurrent assembly planning with genetic algorithms.” *Robotics and Computer-Integrated Manufacturing* 16 (1): 65-72. doi:10.1016/S0736-5845(99)00058-7.

Schaus, P., and Y. Deville. 2008. “A Global Constraint for Bin-Packing with Precedences: Application to the Assembly Line Balancing Problem.” *AAAI*. doi:http://hdl.handle.net/2078.1/85268.

Schaus, P. 2009. “Solving balancing and bin-packing problems with constraint programming.” PhD, thesis, Roman Catholic University.

Sparling, D., and J. Miltenburg. 1998. “The mixed-model U-line balancing problem.” *International Journal of Production Research* 36 (2): 485-501. doi:10.1080/002075498193859.

Suwannarongsri, S., and D. Puangdownreong. 2008. “Optimal assembly line balancing using tabu search with partial random permutation technique.” *International Journal of Management Science and Engineering Management* 3 (1): 3-18. doi:10.1080/17509653.2008.10671032.

Topaloglu, S., L., Salum, and A. A. Supciller. 2012. “Rule-based modeling and constraint programming based solution of the assembly line balancing problem.” *Expert Systems with Applications* 39 (3): 3484-3493. doi:10.1016/j.eswa.2011.09.038.

Uddin, M. K., M. C., Soto, and J. L. M. Lastra. 2010. “An integrated approach to mixed‐model assembly line balancing and sequencing.*”* *Assembly Automation* 30 (2): 164172. doi:10.1108/01445151011029808.

Wang, Y., and D. Tian. 2016. “A weighted assembly precedence graph for assembly sequence planning.” *The International Journal of Advanced Manufacturing Technology* 83 (1-4): 99-115. doi:10.1007/s00170-015-7565-5