RAIRO - Operations Research

A Constraint Programming Approach To Type-2 Assembly Line Balancing Problem With Assignment Restrictions

--Manuscript Draft--

Manuscript Number:	ro170139
Article Type:	Research article
Full Title:	A Constraint Programming Approach To Type-2 Assembly Line Balancing Problem With Assignment Restrictions
Short Title:	Constraint Programming Approach to Assembly Line Balancing
Section/Category:	
Corresponding Author:	Mehmet PINARBAŞI, Ph.D. Independent Researcher Çorum, TURKEY
Order of Authors:	Mehmet PINARBAŞI, Ph.D.
	Hacı Mehmet ALAĞAŞ, M.Sc.
	Mustafa Yüzükırmızı, Ph.D.
Manuscript Region of Origin:	TURKEY
Keywords:	Assembly line balancing; Type-2 problem; Assignment restrictions; Constraint programming; Mixed integer programming
Abstract:	Main constraints for assembly line balancing problem (ALBP) are cycle time/number of stations and task precedence relations. But due to the technological and organizational limitations, several other limitations, such as; task assignment, station, resource and distance restrictions can be encountered in productions systems. In this study, we evaluate the effect of these assignment restrictions on ALBP. A Constraint Programming (CP) model is proposed and compared to Mixed-Integer Programming (MIP) as a benchmark. The objective is minimizing the cycle time for a given number of stations. We provide a more explicit analogy of the effects of assignment restrictions on line efficiency, the solution quality and computation time. Furthermore, the proposed approach is verified with a real-life problem from a defence manufacturing industry. Computational experiments show that, despite additional assignment restrictions are problematic in mathematical solutions, CP is a versatile exact solution alternative in modelling and solution quality.

RAIRO Operations Research
Will be set by the publisher

A CONSTRAINT PROGRAMMING APPROACH TO TYPE-2 ASSEMBLY LINE BALANCING PROBLEM WITH ASSIGNMENT RESTRICTIONS

Mehmet Pinarbasi[1], Haci Mehmet Alagas[2] and Mustafa Yuzukirmizi[3]

Abstract. Main constraints for assembly line balancing problem (ALBP) are cycle time/number of stations and task precedence relations. But due to the technological and organizational limitations, several other limitations, such as; task assignment, station, resource and distance restrictions can be encountered in productions systems. In this study, we evaluate the effect of these assignment restrictions on ALBP. A Constraint Programming (CP) model is proposed and compared to Mixed-Integer Programming (MIP) as a benchmark. The objective is minimizing the cycle time for a given number of stations. We provide a more explicit analogy of the effects of assignment restrictions on line efficiency, the solution quality and computation time. Furthermore, the proposed approach is verified with a real-life problem from a defence manufacturing industry. Computational experiments show that, despite additional assignment restrictions are problematic in mathematical solutions, CP is a versatile exact solution alternative in modelling and solution quality.

Keywords: Assembly line balancing, Type-2 problem, Assignment restrictions, Constraint programming, Mixed integer programming

The date should precede \maketitle in AMS documentclasses; reported.

[1] Independent Researcher, Ulukavak Mahallesi, Mavral Sokak Yılmaz Apt., 19030 Çorum, Turkey; e-mail: mehmetpinarbasi71@hotmail.com

[2] Kırıkkale University, Department of Industrial Engineering, Faculty of Engineering, 71450 Yahşihan, Kırıkkale, Turkey;e-mail: hmalagas@kku.edu.tr

[3] Independent Researcher, Bahcelievler Mah., Tepebaglar Cad., No:51/26 38280 Talas, Kayseri, Turkey;e-mail: myuzukirmizi@hotmail.com

1. Introduction

Assembly line (AL) is a serial production system in which stations are organized according to task precedence relations, task completion times and cycle time constraints. The decision problem of finding the optimal task assignment of stations with a set of precedence relations is defined as Assembly Line Balancing Problem (ALBP). ALBP was first mathematically formulated by Salveson [42], upon then called simple ALBP. As a solution approach, the author proposed a linear programming model to minimize the total idle time at stations. Later, 0-1 integer programming model was developed by Bowman [8]. Hence forth, ALBP studies became one of the major topics in manufacturing research. ALBP has different classifications in terms of objective functions, task times, layout, product model type, etc. The main classification of ALBP is according to their objectives functions, and is as follows [17, 44]

- Type 1: Minimization of the number of stations for a given cycle time.
- Type 2: Minimization of the cycle time for a given number of stations.
- Type E (Effectiveness): Minimization of both number of workstation and cycle time.
- Type F (Feasible): Obtaining a feasible solution for a given number of workstation and cycle time.

While extremizing above objectives, the main consideration in ALBP studies are precedence relations of tasks with general assumption of all stations are equally equipped respect to machines and workers. However, various assignment restrictions can be encountered in redesigning a line (i.e. Type-2 problem) due to the technological, operational and location decisions. Also these assignment restrictions arise by the time the line needs to be rebalanced. For example, tasks must be assigned to different stations which are performed at the interactive two machines (like press and precision machining). In addition, certain assignment restrictions of tasks need to be considered depending on the production conditions, operator's skills, and space of the workstations and requirements of equipment. These restrictions are necessary for astute design or redesign of the assembly lines. However, the literature for ALBP Type-2 problem only deals with the basic constraint such as precedence constraints. Also, the models for reconfiguration of assembly lines are insufficient in terms of including assignment restrictions [37].

In this study, the problem of balancing a line which has assignment restrictions is considered. Principally, Type-2 ALBP with zoning, station, resource and distance assignment restrictions is studied. The applicability of contemporary method of Constraint Programming which performs well to solve many combinatorial problems is examined as a solution approach and innovative models are developed. While, CP has been widely applied to solve the several combinatorial problems (e.g. scheduling, sequencing, assignment problem etc.) [50], the applications to ALBP are rare. Comparison to Mixed Integer Programming (MIP) as mathematical model is also carried out as a benchmark method. After validating the performance of the proposed CP model from literature examples, assembly line balancing problem of a defence industry firm in Turkey is solved for various assignment restrictions. Results reveal that the proposed CP model is very effective to obtain the optimal balance for assignment restricted ALBP.

This paper is organized as follows. In the following section general assignment restrictions for ALBP is described and literature review is presented. An overview of the constraint programming is given in Section 3. In Section 4, Proposed CP model for AR-ALBP is presented. Mathematical model for AR-ALBP are given in Section 5. A comparison of CP and MIP is presented in Section 6. Section 7 provides the numerical results case by case for each assignment restriction. Discussion of the results is given in Section 8. Conclusion and future researches are addressed in the last section.

2. Assignment Restrictions in The Assembly Line Balancing Problem

Mainly, to model mathematically, simple ALBP includes constraints such as cycle time or number of stations constraints, the precedence constraints and occurrence constraints. In addition to these basic model requirements, there can be additional assignment constraints which restrict the assignment of tasks to the stations [10]. These are:

Task restrictions (zoning restrictions): There are two types of task assignment restrictions: linked tasks and incompatible tasks restrictions. In linked tasks restrictions, a set of tasks has to be assigned to the same stations due to the resource requirement. In contrast, incompatible tasks require different equipment and should not be assigned to the same stations [37, 52].

Resource (attribute) restrictions: When the required machine at a station needs a special space to place in line, the resource restrictions limit the cumulative assignments of the tasks [9, 25].

Station restrictions: The station constraints restrict the possible assignment tasks to stations. There are two station restrictions: tasks must be assigned to a certain station, such as requiring certain equipment, and tasks cannot be assigned to certain stations [16, 22, 52].

Distance restrictions: Due to the production process needs, the tasks should be assigned to stations to observe minimum or maximum distance between tasks. For minimum distance example, a task succeeding the colouring task may have to be performed after the color on the work pieces dry. Maximum distance can be observed in the case where melted metal must be prevented from cooling task [34, 37].

Adapting from the renowned study of Scholl et al. [45] and by adding recent studies, literature review about assignment restricted ALBP research is listed in Table 1. To signify milestone studies, a comprehensive study for type-2 ALBP without restrictions was presented by Klein and Scholl [20]. They also proposed a branch and bound procedure which consisted of a local lower bound method and a new enumeration technique. In a similar study, Uğurdağ et al. [53] provided a two-stage heuristic. They generated an initial solution in stage 1 and improved it using a Simplex method like in stage 2. A bidirectional heuristic procedure was proposed for stochastic type-2 ALBP [26]. Nearchou [30] improved a heuristic using differential evolution method for large size assembly line. Also a particle swarm optimization was proposed by Nearchou [31] and an ant colony optimization was developed by Zheng et al. [58] to solve the type-2 ALBP. Unlike the traditional solution methods to solve the ALBP, Kilincci [18] presented a heuristic method integrating with forward, backward and bidirectional procedures based on the Petri-nets by using reachability analysis.

As it is seen in Table 1, even the task, resource and station restrictions have widely considered in the literature, distance restrictions have not attracted considerable attention. However several industrial firms (e.g. auto mobile, electronics, machine construction etc.) have implemented all restriction types for balancing their assembly lines [45]. The solution methods on AR-ALBP are mainly mathematical models and heuristic or meta-heuristic techniques [29]. Mathematical models are used to enumeration procedures such as branch and bound to seek optimum solution. Constructive or greedy procedures which utilize a priority rule to assign the tasks are two main approaches for heuristic procedures. Although heuristics procedures reach the best solution in reasonable computing time, the optimal solution is not assured. There is a lack of research that focuses on the new exact solution methods for ALBP in the literature. Accordingly, CP solution procedure as an exact method which considers several types of assignment restrictions to solve the assembly line balancing problem has to be addressed. Thus this study is also novel in its solution method using constraint programming and establishing its advantages on AR-ALBP for the case of simultaneously considering several types of assignment restrictions.

2.1. Model assumptions and notations

Basic assumptions of the considered AR-ALBP in this study are as follows:

(1) Each task must be assigned to a station
(2) Serial assembly line layout
(3) Task durations are deterministic and known
(4) Tasks are not divisible
(5) Total number of stations is fixed and known.
(6) A single model assembly line without buffers
(7) Precedence constraints are known and must be stable on task-station assignments
(8) At least one task must be assigned to each station
(9) Incompatible and linked tasks are considered between some pair of tasks
(10) Resource and station restrictions may exist which limits assignments
(11) Distance restrictions between tasks are considered

Accordingly, the following notations listed in Table 2 are used in the description of models for AR-ALBP-2:

Table 1. Literature review and solution methods for ARALBP

Restriction	Paper	Solution Method
Linked	Pastor and Corominas [34]	M,H
	Lapierre and Ruiz [21]	H
	Rekiek et al. [38, 39]	H
	Miralles [28]	M
	Vilarinho and Simaria [54, 55]	M, H
	Boysen and Fliedner [9]	M
	Purnomo et al. [37]	H
	Tuncel and Topaloğlu [52]	M
Incompatibles	Bautista et al. [5]	H
	Pastor and Corominas [34]	M,H
	Rekiek et al. [39]	H
	Bautista and Pereira [3]	H
	Boysen and Fliedner [9]	M
	Lapierre and Ruiz [21]	H
	Vilarinho and Simaria [54, 55]	M, H
	Purnomo et al. [37]	H
	Tuncel and Topaloğlu [52]	M
Resource	Carnahan et al. [12]	H
	Liu and Chen [25]	M,H
	Pastor et al. [33]	H
	Sawik [43]	M
	Wilhelm and Gadidov [56]	M
	Miralles [28]	M
	Ağpak and Gökçen [2]	M
	Bautista and Pereira [4]	H
	Boysen and Fliedner [9]	M
	Corominas et al. [13]	M
Station	Kim et al. [19]	H
	Gadidov and Wilhelm [16]	M
	Lee et al. [23]	H
	Pastor et al. [33]	H
	Rekiek et al. [38, 39]	H
	Lapierre and Ruiz [21]	H
	Vilarinho and Simaria [55]	M, H
	Lapierre et al. [22]	H
	Purnomo et al. [37]	H
	Tuncel and Topaloğlu [52]	M
Distance	Deckro [15]	M
	Pastor and Corominas [34]	H,M
	Purnomo et al. [37]	H

M: Mathematical Model, H: Heuristic

3. Constraint Programming

The main aim of this research is to utilize problem specific approaches for ARALBP. Constraint programming (CP) is an alternative programming technique to MIP, generated by combining effectiveness in achieving the optimal solution of linear programming and easy definition property of logical expressions of computer programming methods. CP is *an exact method* among combinatorial optimization solving methods with single or multi-objective functions as branch and bound, dynamic programming etc. [48]. Main concept of CP is constraints. Domain is a set of values that can be assigned to variables. Each constraint is defined as relation between some variables related to their domains. The constraints may be of various different types: linear, non-linear, logical or global constraints.

Table 2. Notations of the model

Symbol	Meaning
c	Cycle time
W	Set of workstations
$\|W\|$	Number of elements of set W
T	Set of tasks
$\|T\|$	Number of elements of set T
t_i	Operation time of task i
Pr	The set of precedence relations between task pairs
P_i^*	The set of all predecessors of task i
F_i^*	The set of all followers of the task i
d_{ij}^-	Minimum distance between task i and j
d_{ij}^+	Maximum distance between task i and j
IT	Set of all incompatible task pairs
LT	Set of all linked task pairs
LD	Set of task pairs with minimum (lower bound) distance
UD	Set of task pairs with maximum (upper bound) distance
x_{ik}	$= 1$ if task i assigned to station k
	$= 0$ otherwise

CP has been widely used to solve some NP-hard combinatorial problems. Some recent studies by subjects can be counted as following: scheduling problems [32,49, 51,57]in supply chain configuration problems [24], timetabling [1], the car sequencing problem [46]. Brailsford et al. [11] can be examined in detail as a literature review for other applications of CP.

ALBP is firstly formulated as a CP model by Bockmayr and Pisaruk [7] and a combined method with integer programming and CP is offered. Pastor et al. [35] presented a comparative study for performance of CP and MIP for simple ALBP type 1 and 2 problems. Although the authors stated that these methods cannot be compared exactly in terms of the efficiency, CP model was better and have a faster formulation than MIP even for large size problems. Also, Topaloğlu et al. [50] has proposed a solution procedure with rule-based CP modelling to solve ALBP. They presented a comparative result of CP and integer programming in terms of modelling capacity, solution quality and CPU time. These studies did not consider the assignment restrictions. Since the assignment restrictions reduce the domain intervals for the decision variables in our CP model, CP could reach a solution quickly.

3.1. Constraint programming methodology

A CP model is generally expressed by Constraint Satisfaction Problems (CSP). CSP are defined as problems with the decision variables, the constraint sets and no objective function in the model. The formal definition of CSP is as follows:

Definition 1. *A CSP is notated as a triple such that (X,D,C), where;*

X is a finite set of n variables $X = \{x_1, ..., x_n\}$

D which is called a domain is the set of possible values that may be assigned to each variables, $v_i \in D(x_i)$ $i = 1, ..., n$.

$C = \{C_1, ..., C_m\}$ is a finite set of constraints. C is a relation between some variables of X and $var(C_j) = \{x_1, ..., x_k\}$ is referred this subset of variables S_i; $C_j \subseteq D(x_1) \times \cdots D(x_k)$,$S_i \subseteq X$.

A solution of a CSP consists of assigned values to the variables when all constraints are satisfied. An assignment is a set of variable/values couples such that $A = \{(x_1, v_1),\ (x_2, v_2), ..., (x_r, v_r)\}$. If an assignment satisfies all constraint, it is called consistent; otherwise if it violates one or more constraints it is inconsistent. If all the variables take a value from their domain then an assignment is *complete*, otherwise is *partial* [47].

Briefly, a CSP is formulated as [27]:

$$C_i(x_1, x_2, ..., x_n)\ =\ 1, 1 \leq i \leq m$$

$$x_j \in D_j, 1 \leq j \leq n$$

In this model, if $C_i(x_1, x_2, ..., x_n)$ is satisfied, i.e. is equal to 1, a consistent solution has been achieved. An algorithm which solves to CSP consists of two main situations: variable selection and value selection, when all the constraints are satisfied.

CP solvers have two fundamentals concepts in general: search tree, propagation and domain reduction. In search tree, a decision variable is declared as a node and a possible assignment shows a branch related to variables. The search starts with an empty assignment and proceeds until there are no variables that can be assigned a value. If the search could not reach a possible solution, backtracking mechanism is executed to try some other branches. And then the search tree has been constructed in this manner iteratively. Many search strategies can be used to guide the backtracking for obtaining an assignment: depth first strategy, multi-point strategy, restart strategy and automatic [40]. In this study, since we concentrated on the modelling of AR-ALBP and achieving a solution, automatic search strategy is preferred as default setting in OPL to improve the search performance and to guide the current solution towards the optimal solution.

The propagation (or consistency) is used to filter the variable domains by eliminating the inconsistent values. As soon as a variables' domain is changed, constraints related to that variable are propagated. Domain reduction is a process which removes the non-assigned variable values that do not satisfy the constraints. So we obtain a consistent assignment at every iteration of the backtracking. Let an illustrative example on the point of ALBP to explain the concepts of the constraint propagation. Assume that, tasks 2 and 5 are two selected tasks with their domain interval $D_2 = [4, 7]$ and $D_5 = [2, 5]$ respectively and the variables defined as the station number assigned to the tasks are x_2 and x_5. And there is precedence relations between these tasks, task 2 should be preceded the task 5 ($x_2 \leq x_5$). Since the constraint states that x_5 should be greater than or equal tox_2, $x_5 = 2$ and $x_5 = 3$ is removed by reducing domain of the task 5. So, D_5 becomes $[4, 5]$. This process should be applied for all of the precedence relations between tasks, so all of the variable domains will be modified. The overall solution procedure of the CSP can be defined branch and propagate which is similar to branch-and-bound used to solve the combinatorial optimization problem. For more detailed information about fundamental concepts of CP, reader may review [14, 36]. Algorithm 1 is summarized the overview of the CP solution framework.

Algorithm 1: CP solution procedure adapted for AR-ALBP

Step 1. (Task selection)
Select a task for partial assignment
Assign a value for selected task
Generate the set of constraints
If there is no task for assignment
Then Backtrack (go to the beginning of the Step1)
Otherwise go to Step 2
Step 2. (Propagation and Domain Reduction)
Apply propagation and domain reduction procedure to the partial assignment
Adjust decision variables and objective function value
Step 3. (Feasibility)
If any feasible assignment does not satisfy at least one of the constraints
Then Backtrack
Otherwise go to Step 4
Step 4. (Termination)
There exists any unassigned task to the station then go to Step 1(Bactracking)
Otherwise Stop and return the result.

In the above algorithm, a search tree which is the main idea is built. While there are still tasks remaining to be assigned to any stations, one of these jobs is selected with the partial assignment. After a value is assigned to the selected task, the set of constraints are generated to check the feasibility. There are two ways in this case. First, propagation and domain reduction procedure is applied to the current partial assignment and then other steps are carried out. Second, if the set of assignable task is empty, the algorithm must backtrack. Feasibility checking is performed by using generated constraint sets on the partial assignment. If feasibility checking is not verified, the algorithm backtracks. If there is any unassigned task to the station, the algorithm returns the Step 1 (i.e. backtracking), in other cases the algorithm terminates and the result is reported.

Whereas CSP is used to find a feasible solution, constraint optimization problem (COP) allows the use of a predefined objective function. The objective f is a function which is related to an assignment launched by the backtracking. Objective function can be minimized or maximized by the assignment A.

4. Proposed Constraint Programming Model for AR-ALBP

Our CP model is based on the model which was highlighted by [35]. His model consists of Equations 1, 3 and 5 as mentioned below. We propose a new CP model to generate all feasible task assignments by adding problem specific assignment restrictions.

Considering that Definition 1 above, ALBP can be modelled as a CSP without any objective function as follows.

- X is a decision variable set. $x_i \in X; \forall i \in T$: The value of this variable gives the assigned number of station of task i.
- D is domain for each decision variable. Each task can be assigned to a station among station interval$D(x_i) = \{1, ..., \|W\|\}$.
- The set of constraints C for ALBP:
  - The all precedence constraints between tasks can be satisfied:

$$x_i \leq x_j \forall (i, j) \in \mathrm{Pr}$$

For example, an assignment A depended on this CSP model for ALBP with 7 tasks and 3 stations corresponds to following:

$$A = \{(x_1, 1), (x_2, 1), (x_3, 3), (x_4, 1), (x_5, 2), (x_6, 2), (x_7, 3)\}.$$

Our proposed constraint optimization programming models are as follows:

Objective Function: The objective function in CP model is minimization of the cycle time.

$$Minimize \quad c \tag{1}$$

Decision variable:

$$StationNumber\,[Task_i] \in W \qquad \forall i \in T \tag{2}$$

Differing from a mathematical model, decision variable is stated as the assigned number of station for task and an integer value between $1, ..., \|W\|$. We can easily satisfy all the restrictions by using this decision variable in constraint programming model.

Precedence relations: The precedence relations are satisfied as following statement:

$$StationNumber\,[Task_i] \leq StationNumber\,[Task_h] \qquad \forall\,(i, h) \in Pr \tag{3}$$

Occurrence and station restrictions: In an assembly line, for a station to be formed a task must be assigned to that station. These constraints express the requirement of at least 1 task is assigned to a station. If a task i must be assigned to a certain station$j$, the restriction is stated as $StationNumber[Task_i] = j$.

$$\underset{i \in T}{\exists} StationNumber\left[Task_i\right] : \left(StationNumber\left[Task_i\right] = j\right) \qquad \forall j \in W \quad (4)$$

To satisfy the occurrence restrictions, the statement $count(StationNumber\left[Task_i; \forall i \in T\right]; j) \geq 1 \forall j \in W$ is used to script where $''count''$ as a function is a special construct in IBM ILOG Software. Using this form, at least one of variable $StationNumber\left[Task_i\right]$ has taken the station number j.

Cycle time restrictions:

$$\sum_i t_i \leq c \qquad \forall i \in T | \forall j \in W \wedge \left(StationNumber\left[Task_i\right] = j\right) \quad (5)$$

The station time is sum of the task times which are assigned to the station. This time must not be reached the cycle time. Equation 5 ensures that these restrictions are satisfied.

Task assignment restrictions:

We can easily establish the task assignment restrictions with the above decision variable as follows:

$$StationNumber\left[Task_i\right] = StationNumber\left[Task_j\right] \qquad \forall (i,j) \in LT \quad (6)$$

$$StationNumber\left[Task_i\right] \neq StationNumber\left[Task_j\right] \qquad \forall\left(i,j\right) \in IT \quad (7)$$

In this way, all of the task assignment constraints are satisfied clearly. For example $StationNumber\left[Task_1\right] = StationNumber\left[Task_4\right]$ ensure that task 1 and task 4 must be assigned into the same station. Accordingly, for incompatible task $StationNumber\left[Task_6\right] \neq StationNumber\left[Task_7\right]$ statement provides that task 6 and task 7 cannot be assigned to the same station.

Distance restrictions: Station intervals between tasks can also be easily defined by using our decision variable in the CP model. It is formulated as follows:

For minimum distance restrictions;

$$\left|StationNumber\left[Task_i\right] - StationNumber\left[Task_j\right]\right| \geq d_{ij}^- \qquad \forall\left(i,j\right) \in LD \quad (8)$$

For maximum distance restrictions;

$$\left|StationNumber\left[Task_i\right] - StationNumber\left[Task_j\right]\right| \leq d_{ij}^+ \qquad \forall\left(i,j\right) \in UD \quad (9)$$

There aren't any solutions for task assignment restricted assembly line balancing for type-2 using constraint programming in the literature. This procedure can generate all possible task assignments for relevant data set and compute the objective functions for a given number of stations.

Cycle time and the line efficiency are performance measures to evaluate the assignments. The procedure finds task assignments that have the minimum cycle time; hence the maximum line efficiency.

5. Mixed Integer Programming Model for AR-ALBP Type-2

The proposed mathematical program for Type-2 AR-ALBP by adapting formulations for Type-1 AR-ALBP from [45] is as follows:

Objective Function: The aim is also the minimization of the cycle time for a given number of stations.

$$Minimize \quad c \tag{10}$$

Variable definition: The decision variables can be defined as follows (if task i. is assigned to station k,$x_{ik} = 1$; otherwise$x_{ik} = 0$)

$$x_{ik} \in \{0, 1\} \qquad \forall (i, k) \in (T \times W) \tag{11}$$

Occurrence and station restrictions: Each task must be assigned to exactly one station.

$$\sum_{k=1}^{\|W\|} x_{ik} = 1 \quad \forall i \in T \quad and \qquad x_{ik} = 1, k \quad \text{predefined station for task } i \tag{12}$$

Precedence restrictions: A task can be assigned to a station only if all its predecessors have been assigned to that station or earlier stations:

$$\sum_{k=1}^{\|W\|} k \cdot x_{ik} \leq \sum_{k=1}^{\|W\|} k \cdot x_{jk} \qquad \forall (i, j) \in Pr \tag{13}$$

Cycle time restrictions: Any station time must be less than the cycle time:

$$c \geq \sum_{i=1}^{\|T\|} t_i \cdot x_{ik} \qquad \forall k \in W \tag{14}$$

Incompatible tasks restrictions: Task i and j, $(i,j) \in IT$, must not be assigned to the same station k.

$$x_{ik} + x_{jk} \leq 1 \qquad \forall\, (i,j) \in IT \quad (15)$$

Linked tasks restrictions: Task i and j, $(i,j) \in LT$, must be assigned to the same station, if the task pair has at least one common station within the possible assignable station sets for each tasks [45].

$$\sum_{k=1}^{\|W\|} k \cdot x_{ik} = \sum_{k=1}^{\|W\|} k \cdot x_{jk} \qquad \forall (i,j) \in LT \quad (16)$$

Minimum distances: We can state minimum distance restriction for task i and j,$(i,j) \in LD$, as follows:

$$\left| \sum_{k=1}^{\|W\|} k \cdot x_{jk} - \sum_{k=1}^{\|W\|} k \cdot x_{ik} \right| \geq d_{ij}^{-} \qquad \forall (i,j) \in LD \quad (17)$$

Maximum distances: This restriction for task i and j, $(i,j) \in UD$, can also be expressed as follows:

$$\left| \sum_{k=1}^{\|W\|} k \cdot x_{jk} - \sum_{k=1}^{\|W\|} k \cdot x_{ik} \right| \leq d_{ij}^{+} \qquad \forall (i,j) \in UD \quad (18)$$

Capacity utilization of the line which is used as another performance parameter in this study is measured in Equation 19. While the cycle time is minimized by reducing the idle time in the assembly line, the line efficiency increases.

$$Line\, eff. = \frac{\sum_{i \in \|T\|} t_i}{\|W\| \cdot c} \qquad (19)$$

6. Constraint Programming versus Mixed Integer Programming

Mathematical models have important characteristics: variables, constraints and solution space. The number of variables and types (e.g. integer, binary) in the model are directly related to problem complexity. Solution space consists of all feasible solution to be obtained by enumeration procedure. Even if the number of constraints could restrict the solution space, it may complicate the solution. These characteristics may lead to exponential growing solution time while increasing problem size. In order to cope with those problems, new logic programming techniques have been developed by researchers. One of these techniques is CP and is now used widely for solving several combinatorial problems. To solve the ALBP efficiently, a new integer decision variable instead of the integer variable used in MIP model are defined in the CP model to reduce the number of variables. Furthermore, one of the innovative properties of the proposed model is that the other problem types AR-ALBP (i.e. type-1, type-E and type-F) can be solved by modifying proposed CP model.

Puget and Lustig [36] compared CP and MIP in terms of three dimensions: modelling ability, node processing and search strategy. Modelling concepts of two approaches consist of general elements: decision variables, constraints, objective function and search tree. Even theory of search strategies is very similar, two approaches has different approach with regard to node processing.

Lustig and Puget [27] stated that CP differs from MIP in two fundamental manners. Firstly, while MIP divide the search space with non-fractional value of decision variables, CP splits search space by choosing a point that is generated by using any set of constraints. To eliminate the suboptimal solution, MIP computes a lower bound for every node of current search. But CP concerns the eliminating of infeasible solutions. In the second manner, variable selection step of branch and bound can be developed by CP framework. CP allows the users to determine his own branching strategy with regard to formulation of the problem. Combining of CP and LP can be very attractive research area that the researcher can build problem-specific search strategies for branch and bound.

CP has some advantages over the Mixed-Integer Programming (MIP). One of the strong features of CP is representation language. CP has specialized constraints, logical constraints and non-linear cost function or constraint are easily defined in such a natural and compact way. In contrast, MIP models support only linearised logical constraints or quadratic convex constraints.

The disadvantages of CP can be counted as memory usage and variable definition by comparison to MIP. CP needs more memory usage since each variable is defined as a domain. CP supports only discrete variables, and continuous variables are not supported [41]. However, CP can be an alternative to MIP for allocation problems that have a slow convergence.

Overall, modelling the problem as a constraint programming is very simple because of the practical language of CP script. Moreover, CP can find the feasible solution quickly by using specific search and propagation algorithm. Also fewer nodes are explored with CP by means of domain reduction method that eliminates some variable values from the solution tree. Considering these advantages, CP can be preferable and efficient method to solve the assembly line balancing problem.

7. Numerical Results

In order to assess the proposed procedure, 9 test problems are solved varying the number of tasks between a minimum of 25 and a maximum of 148 tasks. These test problems are from well-known benchmark data set in the literature and can be downloaded from http://www.alb.mansci.de. We emphasize that these experimented problems are noted as moderate to large size problems. Nevertheless, these data sets included only precedence constraints. For AR-ALBP-2 evaluations, appropriate task assignments, station and distance restrictions are added by the authors for each test problem. These restrictions are listed in Table 3 and Table 4. For example, in Warnecke problem (P58), linked restriction between task 43 and 45, and incompatible restriction between task 34 and 41 exist. For station restrictions -in 10 station problem-, tasks 3, 6 and 22 must be assigned to the station 3, tasks 36 and 41 to station 6, and vice versa. Between tasks 16 and 22, a minimum two-station distance must be kept. Tasks 44 and 55 should have maximum one-station distance in between.

MIP models for AR-ALBP-2 problem are solved using IBM ILOG CPLEX version 12.6 and constraint programming models are solved by IBM ILOG CP Optimizer version 12.6. The time limit for each run is 10,000 seconds for MIP and CP. Cycle time, line efficiency and CPU time are reported as the performance measurements. All test problems are run by using a PC with Intel Core i5-2410M, 2.30 GHz processor and 5 GB memory.

Table 3. List of linked and incompatibles restrictions

Author	Problem	Linked restrictions	Incompatibles restrictions
Roszieg	PP25	(9,13), (23,25)	(1,8), (8,24), (15,21), (19,25)
Gunther	P35	(2,17), (30,34)	(3,19), (8,13), (17,25), (26,34)
Kilbridge	P45	(1,15), (22,33), (38,45)	(7,31), (18,42), (39,43)
Warnecke	P58	(43,45), (46,47)	(13,27), (34,41), (49,52)
Wee-Mag	P75	(28,56), (41,44), (64,74)	(3,24), (6,51), (11,25), (24,41), (39,51), (52,73), (59,75)
Lutz	P89	(8,15), (19,20), (32,34), (44,63)	(10,33), (20,34), (45,80), (83,85)
Mukherje	P94	(22,26), (29,59), (52,78), (81,82), (88,90)	(5,33), (12,17), (23,54), (40,56), (57,65), (64,71), (72,90), (82,88)
Arcus	P111	(14,22), (47,56), (92,93), (107,109)	(10,25), (17,52), (26,33), (45,64), (71,104), (105,108)
Bartholdi	P148	(4,8), (8, 36), (46,47), (57,65)	(4,140), (17,33), (39,123), (81,118), (127,134), (139,142)

TABLE 4. List of stations and distance restrictions

Problem	#stations	Station restrictions{task, station}	Distance restrictions{(task, task),distance}	
			Minimum	Maximum
P25	4	{11,3}	{(4, 22),1}	{(15,24),1}
	8	{16,5}	{(18,22),1}	{(6,7),1}
P35	9	{4,3}, {(20,21),5}	{(10,16),2}	{(3,25);4}
	14	{(11,25,26),11}, {18,5}	{(16,20),2}	{(25,26),2}
P45	4	{5,3}, {9,4}, {23,2}	{(21,24),1}, {(25,28),1}	{(3,10),2}, {(18,21),1}
	10	{3,7}, {14,2}, {(37,43),4}	{(2,10),2}, {(25,28),1}, {(33,41),3}	{(5,9),1}, {(22,33),1}, {(29,37),4}
P58	10	{(3,6,22),3}, {(36,41),6}, {50,9}	{(16,22),2}, {(14,17),4}	{(18,22),1}, {(44,55),1}
	17	{(2,22),4}, {(13,20),5}, {35,12}	{(2,3),3}, {(14,23),4}	{(7,16), 3}, {(43,48),1}, {(52,56),1}, {(53,57,1)}
P75	15	{(13,22),4}, {29,13}, {44,11}, {46,4}, {72,10}, {56,11}	{(24,64),2}, {(4,16),2}, {(50,68),1}, {(24,63),3}	{(4,42),3}, {(23,69),2}, {(68,75),4}
	22	{(3,7,),5}, {14,9}, {20,7}, {24,6}, {(29,36,40),8}, {37,16}, {(56,75),12}, {70,20}	{(4,16),2}, {(58,63),2}	{(28,41),1}, {(31,40),1}, {(55,59),1}, {(69,71),4}
P89	19	{(8,16),3}, {(26,31),6}, {49,9}, {59,12}, {66,13}	{(1,5),1}, {(23,31),2}, {(40,50),2}, {(43,60),3}	{(74,82),2}, {(20,30),2}, {(49,55),4}
	28	{6,1}, {16,10}, {37,15}, {(60,62,65),19}, {(83,84,87),27}	{(1,5),1}, {(23,31),2}, {(40,50),2}, {(43,60),3}	{(74,82),2}, {(20,30),2}, {(49,55),4}
P94	16	{2,2}, {12,4}, {35,7}, {(42,51,61,71),9}, {53,6}, {77,10}, {11,4}	{(14,21),1}, {(53,57),4}, {(76,77),2}	{(5,12),2}, {(35,37),2}, {(90,93),1}
	26	{12,8}, {(14,18,21), 5}, {(47,57), 16}, {85,12}	{(39,49),3}, {(54,61),2}, {(68,74),2}, {(74,77),4}	{(10,14),1}, {(20,31),4}, {(55,79),2}
P111	13	{(8,12,14,18,30),4}, {(95,100,105),12}	{(8,9),2}, {(34,42),2}, {(76,77),2}, {(91,95),1}	{(15,21),2}, {(54,62),1}, {(103,106),1}
	27	{8,10}, {12,2}, {14,3}, {(18,23,42),9}, {(95,100,105),26}	{(8,9),2}, {(34,42),2}, {(76,77),2}, {(91,95),1}	{(15,21),2}, {(54,62),1}, {(103,106),1}
P148	10	{(14,21,34,79), 4}, {(127,138),6}	{(5,34),2}, {(75,79),2}, {(130,140),1}	{(7,15),1}, {(53,71),2}, {(112,144),2}
	15	{9,4}, {20,12}, {(62,63,64,65),9}, {145,11}	{(5,34),2}, {(75,79),2}, {(130,140),1}	{(7,159,1}, {(53,71),1}, {(112,144),2}

Four experimental cases are designed to evaluate the effects of assignment restrictions on the line balance which are defined as follows:

- Case 1: linked and incompatible task assignment restrictions
- Case 2: station and distance restrictions
- Case 3: all assignment restrictions
- Case 4: a real-life problem from a defence manufacturing industry

In all of the experiments, precedence and cycle time constraints are default restrictions. In the following sections we present the numerical results case by case.

7.1. Case 1: AR-ALBP with task assignment restrictions

In this first case, in addition to cycle time and precedence constraints, we consider the linked and incompatible task assignment restrictions as listed in Table 3. CPU time and cycle time are used as the metric for comparison of the solutions. Each test problem is also solved for different station numbers.

As it can be seen in Table 5, MIP and CP found the optimal cycle time for test problems between P25 and P45 (small-size problems). For these problems, CPU times of MIP and CP models are very close. If the medium-size problems (P58-P89) are considered; CP has reached all the optimal solutions except in P75 with 15 stations, while MIP has attained the optimal for all problems. Computational CPU times for these problems for CP models are higher than MIP models. For large-size problems (P94-P148), the optimal solutions are obtained for both approaches. However, CP is significantly faster than MIP for large-size problems. To signify solution characteristics, line efficiencies for all test problems are also reported in Table 5. Substantial line efficiency, on the average 0.93, is achieved except in a few test problems (e.g. P111 with 27 numbers of stations and P148 with 15 numbers of stations).

7.2. Case 2: AR-ALBP with station and distance assignment restrictions

In this second case, the station and distance restrictions as listed in Table 4 are considered as well as the precedence and cycle time constraints. As it can be seen in Table 6, MIP has reached the optimal solutions for all problems. But, CP has failed to reach the optimal solution in P75 with 15 stations. In terms of the CPU time, both models show almost same performance. The reason is the station and distance restrictions have limited the solution space and the models could converge to the optimal solution quickly. Thus, it cannot be inferred that one model is superior to other model.

It is worth mentioning that the line efficiencies decrease due to the effect of assignment restrictions. At the same time, optimum cycle time of assignment restrictions. For instance, in the P89 problem, while line efficiencies were 0.95 and 0.91 in Case 1, 0.88 and 0.82 are obtained in Case 2. The optimum cycle times in Case 1 were 26 and 19 time units and in Case 2, 29 and 21 time units, respectively.

Table 5. Test results of Case 1

Problem	#stations	MIP		CP		Line eff.
		c	CPU	c	CPU	
P25	4	32	0.32	32	0.13	0.98
	8	17	0.49	17	0.21	0.92
P35	9	55	1.72	55	2.23	0.98
	14	40	0.85	40	0.56	0.86
P45	4	138	0.26	138	0.29	1.00
	10	56	0.45	56	0.40	0.99
P58	10	155	3.11	155	1.58	1.00
	17	92	240.60	92	227.00	0.98
P75	15	100	90.80	101	10000.00	0.98
	22	69	5.46	69	123.67	0.97
P89	19	26	10.79	26	33.39	0.95
	28	19	86.47	19	1.58	0.91
P94	16	268	1126.00	268	682.79	0.98
	26	171	13.63	171	1.47	0.95
P111	13	11573	10000.00	11573	22.65	1.00
	27	9210	25.86	9210	1.93	0.60
P148	10	564	4.79	564	4.67	1.00
	15	494	9.11	494	0.85	0.76

Overall, assignment restrictions led the line efficiency to decrease, while cycle time to increase.

7.3. Case 3: AR-ALBP with all task assignment restrictions

In this case, all the task assignment restrictions listed in Table 3 and Table 4 are considered together and respective test problems are solved. The experimental results are reported in Table 7. The optimal solutions for all problems are obtained except in P75 for both models. However, CP has obtained a feasible solution (i.e. 104) for this instance within 10,000 seconds and line efficiency is 0.96. Either approach is not superior with respect to the CPU time. As in previous case, solution spaces of the problems decrease as more assignment restrictions are added. Hence, both models obtain a solution quickly.

As expected, similar situations of Case 1 and Case 2 arises in respect of the line efficiency and optimal cycle time. As assignment restrictions force several task assignment rules, it can be conjectured that the line efficiency will decrease and cycle time will increase.

Table 6. Test results of Case 2

Problem	#stations	MIP		CP		Line eff.
		c	CPU	c	CPU	
P25	4	32	0.28	32	0.20	0.98
	8	16	0.21	16	0.14	0.98
P35	9	56	0.21	56	0.16	0.96
	14	40	0.44	40	0.18	0.86
P45	4	138	0.19	138	0.36	1.00
	10	56	0.36	56	0.33	0.99
P58	10	160	0.68	160	0.32	0.97
	17	95	2.28	95	1.65	0.96
P75	15	100	12.41	101	10000.00	0.98
	22	72	0.50	72	0.39	0.95
P89	19	29	0.67	29	0.67	0.88
	28	21	3.22	21	4.67	0.82
P94	16	284	1.59	284	4.63	0.93
	26	171	2.50	171	0.50	0.95
P111	13	13715	121.00	13715	27.21	0.84
	27	11188	0.69	11188	0.28	0.50
P148	10	564	1.75	564	5.06	1.00
	15	501	2.74	501	5.55	0.75

7.4. Case 4: Application of the proposed model to a real-life case

In this section, an experimental case study is carried out to apply the procedure in a real production firm. Motivated by a balancing problem in its assembly line, the company consulted authors for a solution. The company is one of the main defence firms in Kırıkkale, Turkey which produces weaponry for Turkish Army. Thorough and fast production has crucial importance for the company as in global defence industry production.

After the establishment, rebalancing problems have arisen due to some changes in operational and technical conditions. These conditions are related to new product models, new machine updates, technological developments, labor needs, changing the production method and fluctuations in customer demand. Hence, company's operations managers wanted to minimize cycle time of the line by assigning the appropriate tasks into the stations.

The proposed solution procedure is applied to rebalance the new weapon assembly line. The weapon production process has 14 tasks. Many of these tasks require sensitive production techniques. In this context, the numbers of stations are given as 5. Also, due to the modifications mentioned above and sensitive

Table 7. Test results of Case 3

Problem	#stations	MIP		CP		Line eff.
		c	CPU	c	CPU	
P25	4	36	1.12	36	0.10	0.87
	8	18	0.22	18	0.14	0.87
P35	9	56	0.23	56	0.20	0.96
	14	42	0.23	42	0.19	0.82
P45	4	138	0.24	138	0.39	1.00
	10	58	0.27	58	0.15	0.95
P58	10	160	0.64	160	0.38	0.97
	17	96	6.85	96	1.71	0.95
P75	15	N/A	N/A	104	10000.00	0.96
	22	75	0.53	75	0.89	0.91
P89	19	32	0.83	32	0.32	0.80
	28	21	2.32	21	4.68	0.82
P94	16	284	2.44	284	6.77	0.93
	26	171	5.88	171	36.85	0.94
P111	13	13715	1834.00	13715	1785.46	0.84
	27	11188	0.77	11188	0.29	0.50
P148	10	564	1.87	564	4.72	1.00
	15	581	0.76	581	5.43	0.65

production techniques, several assignment restrictions have raised. Weapons production tasks and required task times are shown in Table 8. Moreover, Figure 1 shows the precedence diagram of weapon process.

Essential assignment restrictions are reported in Table 9. For example, linked assignment restriction is needed between task 9 and 10, since these jobs are very similar jobs and require similar equipment to assembly. Task 3 and 6 should have minimum 2 stations distance restriction in between. While task 3 needs a heavy machine (e.g. press) to produce, task 6 needs precision manufacturing techniques. Hence, these two tasks must be assigned far away from each other.

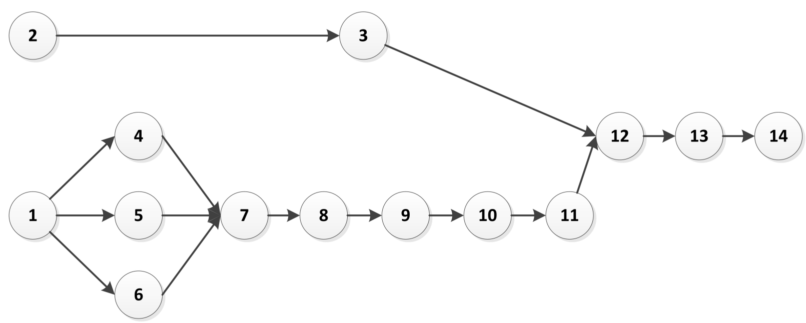
The comparison results for MIP and CP models in terms of the cycle time are reported in Table 10. The CP and MIP models have reached the same optimal solution for first three assignment restrictions cases mentioned in Section 7. Furthermore, assigned tasks to the stations are reported in Table 11. These results state that the proposed CP approach can be easily applied to solve the real life problems.

8. Discussion

As noted in above experimental cases, assignment restrictions have effects on the convergence of the solution and the computing time. When the assignment

Table 8. Weapons production tasks and task times for Case 4

Number	Operation	Time (sec)
1	Assemble the body	90
2	Assemble the firing device	180
3	Assemble the barrel	150
4	Plug the clip switch	96
5	Pick up the screw thread	60
6	Plug the main spring	120
7	Plug the striker	96
8	Assemble the fitting spring	60
9	Pick up the grip safety	30
10	Assemble the safety lock	60
11	Assemble the plastic grip	90
12	Assemble with main body and barrel	30
13	Assemble the slide pin	60
14	Assemble the recoil spring	96



1.jpg

Figure 1. Precedence diagram of weapons production

Table 9. List of assignment restrictions of the weapons production for Case 4

Types	Restrictions
Incompatible	(1,2)
Linked	(5,8), (9,10)
Maximum distance	{(6,7),1)}
Minimum distance	{(3,6),2}
Stations	(12,4)

Table 10. Comparison of MIP and CP model in terms of the cycle time for Case 4

Case	MIP	CP
Without assignment restrictions	270	270
1	276	276
2	270	270
3	306	306

Table 11. Assigned tasks to the stations for each case with proposed model

Case	Assigned tasks to the stations				
	1	2	3	4	5
Without assignment restrictions	1,2	3,6	4,5,7	8,9,10,11,12	13,14
1	1,6	2,4	5,7,8	3,9,10	11,12,13,14
2	1,5,6	4,7,8	2,9,10	3,11,12	13,14
3	1,4,6	5,7,8,9,10	2,11	3,12,13	14

restrictions are added to simple ALBP, possible task assignment combinations are reduced. They influence the problem complexity. Optimal solution and optimum task assignment can change by adding these restrictions into the problem. Addition of these assignment restrictions also reduces the problem constraints and variables [44]. A well-known property of mathematical programming is that reduction of the problem variables can make it easier to solve the problem [6]. CP uses this advantage with fewer variables than their MIP counterparts for each test problem. Furthermore, CP has more constraints than MIP model even if the problem has no additional (assignments) constraints. In terms of the objective functions, it can be claimed that the model without any assignment restrictions is a lower bound for the model with assignment restrictions. These results are shown at Table 12.

In general, both MIP and CP have demonstrated very good performance in achieving the optimal solutions for test problems. Line efficiency has decreased as we add more assignment restrictions within the model. The optimal cycle times also increase. The reason is that with assignment restrictions may force some tasks to be assigned to different stations than the model without restrictions. This situation arises from Eq. (19).

Both models performed the same performance for small-size problems with regard to CPU time. For other problems, we can generally say that CPU time decreased by adding assignment restrictions comparing Table 5, 6 and 7 to 8. Except from some large-size problems, CPU time has declined.

Another property is CP models have less variables due to the domain interval for each decision variables. CP models work with far less number of variables than

Table 12. Comparison of size and performance of the CP and MP model without assignment restrictions

Problem	#stations	MIP				CP			
		#variables	#constraints	c	CPU	#variables	#constraints	c	CPU
P25	4	102	65	32	0.31	26	190	32	0.24
	8	202	73	16	0.55	25	298	16	0.27
P35	9	317	98	54	1.22	36	448	54	0.85
	14	492	108	40	2.00	36	633	40	0.40
P45	4	182	115	138	0.49	46	340	138	0.26
	10	452	127	56	0.45	46	622	56	0.41
P58	10	582	148	155	8.75	59	786	155	0.36
	17	988	162	92	1384.26	59	1206	92	1675.87
P75	15	1127	192	100	4.00	76	1392	100	8250.35
	22	1652	206	69	4.01	76	1931	69	13.36
P89	19	1693	145	26	28.43	90	2025	26	6.76
	28	2494	263	18	13.26	90	2844	18	7.54
P94	16	1506	307	268	123.52	95	1905	268	975.28
	26	2446	327	171	16.72	95	2865	171	2.41
P111	13	1445	313	11573	2382.03	112	1867	11573	37.82
	27	2999	341	5689	23.12	112	3449	5689	8.39
P148	10	1482	343	564	4.56	149	1971	564	3.72
	15	2222	353	383	3.97	149	2721	383	1.05

MIP problems. However, CP models have more number of constraints than MIP models which is an advantage for CP to converge quickly.

A comparison of whether MIP or CP is quicker in terms of CPU times is displayed in Table 13. The numbers of instances for each case are counted for experiments. Especially in small-size problems, both models exhibit almost equal CPU time behaviour. For other problems particularly in Case 1, CP is quicker in solution time of a problem. CP gives a good performance for other cases as well. In general, depending on the problem complexity and size, CP is more efficient for solving the AR-ALBPs.

Table 13. The numbers of instances for each case in terms of the CPU time comparison

	Case1	Case2	Case3
Almost equal	8	11	9
CP faster than MIP	7	2	4
MIP faster than CP	3	5	5

9. Conclusion and Future Researches

In this paper, uncomplicated modelling and efficient solution procedures for assembly lines with assignment restrictions are developed. The considered assignment restrictions are: attribute restrictions, linked and incompatible tasks, station restrictions and distance restrictions. The objective of the problem is to minimize the cycle time for a given number of stations and to maximize the line efficiency.

To solve the problem, a constraint programming approach has been developed. This method is an effective technique to solve several combinatorial problems such as scheduling problems, but merely applied to assembly line balancing field. In the solution procedure, CP generates all possible task assignments as feasible solutions while satisfying all restrictions. Then, considering the cycle time and the line efficiency as the performance measures, the best feasible or optimal assignment is obtained. The performance of the proposed procedure is tested and compared to MIP on 9 literature data sets with 18 instances for three cases. The proposed model is also practiced to solve a line balancing problem in a defence industry firm in Kırıkkale, Turkey.

According to the numerical results, CP models give notable performance compared to MIP models especially for large-size problems. In addition, CP modelling is very practical and user-friendly than MIP modelling in terms of scripting language convenience. Consequently proposed solution procedure may be a very effective alternative modelling method to MIP in balancing assembly lines with several task assignment restrictions.

For future studies, the procedure can be improved with other ALBP types such as u-lines, parallel lines and two-sided lines. Another area is to consider several other objective functions such as minimization of the number of stations and cost of line balancing. And also, stochastic and dynamic task time problems can be solved by using the proposed model.

References

[1] Slim Abdennadher and Michael Marte. University course timetabling using constraint handling rules. *Applied Artificial Intelligence*, 14(4):311–325, 2000.

[2] Kürşad Ağpak and Hadi Gökçen. Assembly line balancing: Two resource constrained cases. *International Journal of Production Economics*, 96(1):129–140, 2005.

[3] Joaquín Bautista and Jordi Pereira. Ant algorithms for assembly line balancing. In *International Workshop on Ant Algorithms*, pages 65–75. Springer, 2002.

[4] Joaquin Bautista and Jordi Pereira. Ant algorithms for a time and space constrained assembly line balancing problem. *European journal of operational research*, 177(3):2016–2032, 2007.

[5] Joaquin Bautista, Raúl Suárez, Manuel Mateo, and Ramón Companys. Local search heuristics for the assembly line balancing problem with incompatibilities between tasks. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2404–2409. IEEE, 2000.

[6] Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.

[7] Alexander Bockmayr and Nicolai Pisaruk. Solving assembly line balancing problems by combining ip and cp. In *Proceedings of the 6th Annual Workshop of ERCIM Working Group on Constraints*. ERCIM, 2001.

[8] Edward H Bowman. Assembly-line balancing by linear programming. *Operations Research*, 8(3):385–389, 1960.

[9] Nils Boysen and Malte Fliedner. A versatile algorithm for assembly line balancing. *European Journal of Operational Research*, 184(1):39–56, 2008.

[10] Nils Boysen, Malte Fliedner, and Armin Scholl. A classification of assembly line balancing problems. *European journal of operational research*, 183(2):674–693, 2007.

[11] Sally C Brailsford, Chris N Potts, and Barbara M Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557–581, 1999.
[12] Brian J Carnahan, Bryan A Norman, and Mark S Redfern. Incorporating physical demand criteria into assembly line balancing. *IIE Transactions*, 33(10):875–887, 2001.
[13] Albert Corominas, Laia Ferrer, and Rafael Pastor. Assembly line balancing: general resource-constrained case. *International Journal of Production Research*, 49(12):3527–3542, 2011.
[14] Rina Dechter. *Constraint processing*. Morgan Kaufmann, 2003.
[15] Richard F Deckro. Balancing cycle time and workstations. *IIE transactions*, 21(2):106–111, 1989.
[16] Radu Gadidov and Wilbert Wilhelm. A cutting plane approach for the single-product assembly system design problem. *International Journal of Production Research*, 38(8):1731–1754, 2000.
[17] Michel Gourgand, Nathalie Grangeon, and Sylvie Norre. Metaheuristics based on bin packing for the line balancing problem. *RAIRO-Operations Research*, 41(2):193–211, 2007.
[18] Ozcan Kilincci. A petri net-based heuristic for simple assembly line balancing problem of type 2. *The International Journal of Advanced Manufacturing Technology*, 46(1-4):329–338, 2010.
[19] Yeo Keun Kim, Yeongho Kim, and Yong Ju Kim. Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning & Control*, 11(1):44–53, 2000.
[20] Robert Klein and Armin Scholl. Maximizing the production rate in simple assembly line balancing—a branch and bound procedure. *European Journal of Operational Research*, 91(2):367–385, 1996.
[21] SD Lapierre and AB Ruiz. Balancing assembly lines: an industrial case study. *Journal of the Operational Research Society*, 55(6):589–597, 2004.
[22] Sophie D Lapierre, Angel Ruiz, and Patrick Soriano. Balancing assembly lines with tabu search. *European Journal of Operational Research*, 168(3):826–837, 2006.
[23] Tae Ok Lee, Yeongho Kim, and Yeo Keun Kim. Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers & Industrial Engineering*, 40(3):273–292, 2001.
[24] Haitao Li and Keith Womer. Optimizing the supply chain configuration for make-to-order manufacturing. *European Journal of Operational Research*, 221(1):118–128, 2012.
[25] Chiun-Ming Liu and Chung-Hwa Chen. Multi-section electronic assembly line balancing problems: A case study. *Production Planning & Control*, 13(5):451–461, 2002.
[26] SB Liu, HL Ong, and HC Huang. A bidirectional heuristic for stochastic assembly line balancing type ii problem. *The International Journal of Advanced Manufacturing Technology*, 25(1-2):71–77, 2005.
[27] Irvin Lustig and Jean-Francois Puget. Constraint programming. In S.I. Gass and M.C. Fu, editors, *Encyclopedia of Operations Research and Management Science*, pages 267–273. Springer, 2013.
[28] C. Miralles. Solving procedures for the assembly line worker assignment and balancing problem: Application to sheltered work centres for disabled. In *Proceedings of the XI Escuela Latinoamericana de Verano en Investigacionde Operaciones*, Villa de Leyva, Colombia, 2005.
[29] Koichi Nakade, Akiyasu Ito, and Syed Mithun Ali. U-shaped assembly line balancing with temporary workers. *International Journal of Industrial Engineering*, 21(6):134–146, 2015.
[30] Andreas C Nearchou. Balancing large assembly lines by a new heuristic based on differential evolution method. *The International Journal of Advanced Manufacturing Technology*, 34(9-10):1016–1029, 2007.
[31] Andreas C Nearchou. Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *International Journal of Production Economics*, 129(2):242–250, 2011.

[32] Cemalettin Ozturk and M Arslan Ornek. Optimisation and constraint based heuristic methods for advanced planning and scheduling systems. *International Journal of Industrial Engineering*, 23(1):26–48, 2016.
[33] Rafael Pastor, C Andres, A Duran, and M Perez. Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of the Operational Research Society*, 53(12):1317–1323, 2002.
[34] Rafael Pastor and Albert Corominas. Assembly line balancing with incompatibilities and bounded workstation loads. *Ricerca Operativa*, 30:23–45, 2000.
[35] Rafael Pastor, Laia Ferrer, and Alberto García. Evaluating optimization models to solve salbp. In *International Conference on Computational Science and Its Applications*, pages 791–803. Springer, 2007.
[36] Jean-Francois Puget and Irvin Lustig. Constraint programming and maths programming. *The Knowledge Engineering Review*, 16(01):5–23, 2001.
[37] Hindriyanto Dwi Purnomo, Hui-Ming Wee, and Hsin Rau. Two-sided assembly lines balancing with assignment restrictions. *Mathematical and Computer Modelling*, 57(1):189–199, 2013.
[38] Brahim Rekiek, Pierre De Lit, and Alain Delchambre. Hybrid assembly line design and user's preferences. *International Journal of Production Research*, 40(5):1095–1111, 2002.
[39] Brahim Rekiek, Pierre De Lit, Fabrice Pellichero, Thomas L'eglise, Patrick Fouda, Emanuel Falkenauer, and Alain Delchambre. A multiple objective grouping genetic algorithm for assembly line design. *Journal of intelligent manufacturing*, 12(5-6):467–485, 2001.
[40] ILOG S.A. *ILOG OPL 12.6 Language Manual*. IBM, France, 2013a.
[41] ILOG S.A. *CP Optimizer 12.6 User's Manual*. IBM, France, 2013b.
[42] M.E. Salveson. The assembly line balancing problem. *Journal of Industrial Engineering*, 6:18–25, 1955.
[43] Tadeusz Sawik. Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line. *European Journal of Operational Research*, 143(1):115–124, 2002.
[44] Armin Scholl. *Balancing and sequencing of assembly lines*. Physica-Verlag Heidelberg, 1999.
[45] Armin Scholl, Malte Fliedner, and Nils Boysen. Absalom: Balancing assembly lines with assignment restrictions. *European Journal of Operational Research*, 200(3):688–701, 2010.
[46] Mohamed Siala, Emmanuel Hebrard, and Marie-José Huguet. A study of constraint programming heuristics for the car-sequencing problem. *Engineering Applications of Artificial Intelligence*, 38:34–44, 2015.
[47] Christine Solnon. *Ant colony optimization and constraint programming*. John Wiley & Sons, 2013.
[48] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
[49] Daria Terekhov, Mustafa K Doğru, Ulaş Özen, and J Christopher Beck. Solving two-machine assembly scheduling problems with inventory constraints. *Computers & Industrial Engineering*, 63(1):120–134, 2012.
[50] Seyda Topaloglu, Latif Salum, and Aliye Ayca Supciller. Rule-based modeling and constraint programming based solution of the assembly line balancing problem. *Expert Systems with Applications*, 39(3):3484–3493, 2012.
[51] Mariem Trojet, Fehmi H'Mida, and Pierre Lopez. Project scheduling under resource constraints: Application of the cumulative global constraint in a decision support framework. *Computers & Industrial Engineering*, 61(2):357–363, 2011.
[52] Gonca Tuncel and Seyda Topaloglu. Assembly line balancing with positional constraints, task assignment restrictions and station paralleling: A case in an electronics company. *Computers & Industrial Engineering*, 64(2):602–609, 2013.
[53] H Fatih Uğurdağ, Ram Rachamadugu, and Christos A Papachristou. Designing paced assembly lines with fixed number of stations. *European Journal of Operational Research*, 102(3):488–501, 1997.
[54] Pedro M Vilarinho and Ana Sofia Simaria. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 40(6):1405–1420, 2002.

[55] Pedro M Vilarinho and Ana Sofia Simaria. Antbal: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations. *International journal of production research*, 44(2):291–303, 2006.

[56] Wilbert E Wilhelm and Radu Gadidov. A branch-and-cut approach for a generic multiple-product, assembly-system design problem. *INFORMS Journal on Computing*, 16(1):39–55, 2004.

[57] LJ Zeballos. A constraint programming approach to tool allocation and production scheduling in flexible manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 26(6):725–743, 2010.

[58] Qiaoxian Zheng, Ming Li, Yuanxiang Li, and Qiuhua Tang. Station ant colony optimization for the type 2 assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 66(9-12):1859–1870, 2013.