Balancing of Cost-Oriented U-Type General Resource-Constrained Assembly Line: New Constraint Programming Models

Hacı Mehmet Alakaş (hmalagas@gmail.com)
Kirikkale University, Faculty of Engineering and Architecture, Industrial Engineering Dept.
https://orcid.org/0000-0002-9874-7588
Mehmet Pınarbaşı
OSTIM Technical University, Faculty of Engineering

Research Article

Keywords: Assembly line balancing, U-type, general resource constraints, conjunctive constraints, constraint programming

Posted Date: March 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-844843/v1

Balancing of Cost-Oriented U-type General Resource-Constrained Assembly Line: New Constraint Programming Models

Hacı Mehmet Alakaş[a,*], Mehmet Pınarbaşı[b]

[a] Kırıkkale University, Faculty of Engineering and Architecture, Dept. of Industrial Engineering, Address: Ankara Yolu 7. Km, 71451, Yahşihan/Kırıkkale, Turkey, hmalagas@kku.edu.tr, Tel: +903183574242/1045, ORCID ID: 0000-0002-9874-7588

[b] OSTIM Technical University, Faculty of Engineering, Dept. of Industrial Engineering, OSTIM OSB, 06374, Yenimahalle/Ankara, Turkey, mehmetpinarbasi71@hotmail.com, ORCID ID: 0000-0003-3424-2967

[*]: Corresponding Author

Abstract

In simple assembly line balancing problems, it is assumed that the resources required to perform the tasks are available at the relevant station for task assignment. However, each task may need different resource types depending on the difficulty, complexity and technical requirements of the tasks in real life. For this reason, tasks and resources belonging to these tasks must be assigned to the relevant station while balancing the line. In this study, the resource-constrained U-shaped assembly line balancing problem(U-GRCALBP) is discussed. According to the literature research, there is no study dealing with U-GRCALBP. Two different constraint programming (CP) models that define the resource constraints as "and/or" constraint types with concurrent resource types have been developed. In these models, the sum of resource usage costs and station opening cost minimization is aimed. The models are explained with an illustrative example and the efficiency of the models was tested by deriving new resource constraints on sample data sets. The number of stations obtained, the total number of resources used, total station opening and resource usage costs, and CPU time were used as performance criteria. According to the numerical results, it has been shown that both models are effective in solving the problem.

Keywords: Assembly line balancing, U-type, general resource constraints, conjunctive constraints, constraint programming

1. Introduction

The products had to be produced in high quantities but economically due to the increasing demand. One of the production systems established to produce products more economically is assembly lines. The assembly line consists of sequential workstations connected to each other by a transport system in a specified area. The assigned tasks are fulfilled at these stations and the main product is obtained by assembling the sub-parts of the product. The assembly lines, which were first established by Henry Ford, are now used to produce many products such as automobiles, white goods, furniture, etc.

Different types of assembly lines have emerged due to the changes in their features, such as layout type, number of models, and task time variability. Assembly lines are classified as simple or straight lines and U-lines according to the layout type. While the stations are arranged in a row on straight lines, in U-type lines, the stations have a layout arrangement in the form of the letter U. U-type lines have been preferred in practice due to the following situations: (i) Failure to establish a straight line due to space constraints, (ii) to reduce cycle time (i.e., to increase line efficiency), (iii) to reduce performance degradation due to monotony in employees, (iv) Significantly increasing the production output rate by the Just in Time (JIT) principle. The contribution of U-type lines to JIT systems can be given as increasing production flexibility, improving communication between stations, and facilitating operator mobility between stations. Since the assembly line is formed in a U shape, the beginning and end of the line are aligned. The fact that the tasks to be performed will be performed on either side of the U-type line allows more task assignment alternatives to be evaluated and more efficient lines to be established(Eghtesadifard et al. 2020).

Assembly line balancing problems (ALBP) are the problem of assigning the tasks to the station where the tasks will be performed under the desired constraints in all types of assembly lines. In ALBP, a combinatorial NP-hard problem, while trying to optimize one or more objectives, priority relations, task assignment, and station cycle time constraints, should also be provided. However, in practice, besides these basic constraints, some specific constraints arising from the sectorial conditions and the technical characteristics of the tasks to be performed at the stations can also be taken into account. In the literature, these constraints are classified as assignment constraints, resource constraints, and distance constraints(Pinarbasi et al. 2019). Assembly lines, resources are defined as human and/or machine. The number or cost of these resources is also taken into account in ALBP. Such problems are called resource-constrained assembly line balancing problem (RCALBP). RCALBP was first addressed by Ağpak and Gökçen(2005), and a mathematical model for the problem was proposed. It is ensured that the tasks that need the same resources in their work are assigned together. In this way, it is aimed to balance the line and to use fewer resources.

In the literature, it is seen that exact and heuristic solution approaches are frequently used for the solution of ALBP (Sivasankaran and Shahabudeen 2014). Exact solution methods are methods by which the optimum solution of the problem can be obtained. Examples of these methods are mathematical modeling, constraint programming, and branch-bound algorithm. Examples of heuristic solution approaches are genetic algorithm, taboo search, ant colony, annealing simulation, and problem-specific heuristics. In the problem addressed in this study, alternative and different numbers of resources are needed. In this case, it requires logical constraints in the modeling of the problem. The constraint programming method was preferred in the solution of the problem because it provides exact solutions and allows the constraints in different structures such as mathematical, logical, linguistic, etc., to be defined easily. Constraint programming, which is a preferred method in solving combinatorial problems, has been widely used in ALBP solutions in recent years. It is seen as a result of the studies that it is an effective method in ALBP solution.

In this study, the U-type general resource-constrained assembly line balancing problem (U-GRCALBP) is discussed. The aim of the study is to minimize the resource cost due to the resource used and the stations opening cost. The constraint programming approach is used as a solution method. Two different constraint programming models have been proposed to solve the problem. New resource constraints are generated for problem sets in the literature to test the proposed models' performance. The results showed that constraint programming is effective in solving U-GRCALBP.

The main contributions of this study can be listed as follows: 1) The U-type resource-constrained problem is addressed for the first time, 2) The constraint programming solution approach, which is an exact method, has been used for the solution of the problem, 3) The tasks are performed with different types and numbers of alternative resources in this problem type. 4) Resource constraints are defined as general resource constraints in the "and/or" structure. 5) It has been shown that CP is an effective solution method in solving U-GRCALBP.

This paper is organized as follows. Section 1 is related to the introduction. In Section 2, studies using both resource-constrained ALBP and constraint programming in the ALBP solution are given in comparative tables. In the third chapter, the RCALBP problem is defined, and the assumptions of the problem discussed are also included. In the fourth chapter, two CP models proposed for the solution of the problem are given. In the fifth chapter, an illustrative example solution is explained with the suggested models. In the sixth chapter, the numerical experiments performed to test the models' performance are included and the comparative results of the experiments are presented. In the seventh section, the results of the study and recommendations for future studies are given.

2. Literature review

Since the ALBP literature is quite extensive, the literature to be given in the study has been compiled to include RCALBP studies. There are not many studies about RCALBP in the literature. Studies classified according to the results of the literature review are summarized in Table 1. In the classification of the studies, the following features of ALBP are taken into consideration: line shape, resource type, objective function, resource constraint type, and solution method. The literature will be explained in order of this classification.

The studies have been handled in three different ways in terms of line shape: straight, U-shape, and two-side. When the results given in Table 1 are examined, it is seen that most work (i.e., 12 pieces of work) is done for straight assembly lines. There is only one study for each of U-shape (Ogan and Azizoglu 2015) and two-sided (Mete and Agpak 2013) RCALBP. All the results of these assessments, studies taking into account U-type RCALBP are still insufficient.

Studies can be classified into two groups as simple and general cases according to resource types. There are many studies (i.e., 11 pieces) for the simple case RCALBP in the literature. There are only four studies (Corominas et al. 2011; Mete and Agpak 2013; Pereira 2018; Alakaş 2021) for general case RCALBP(Table 1). According to the literature research, there is still a gap in the literature for the solution of ALBP with general resource constraints.

When the literature studies in Table 1 are examined in terms of objective function, it is determined that the studies use eight different objective functions. It is seen that the literature studies mostly consider the minimization of the number of stations (Ns) and the minimization of the number of resources (Nr) as objective functions. In addition to these, in the literature, total equipment cost (Ogan and Azizoglu 2015), line efficiency maximization (Jusop and Ab Rashid 2017), total workforce balancing (Pereira 2018; Chen et al. 2018), resource cost minimization (Pereira 2018), cycle time minimization(Chen et al. 2012; Alakas et al. 2020) have been considered as objective functions. A result seen in Table 1 is that the problem is mostly considered as a multi-objective in the studies. Usually, the minimization of the number of stations and the number of resources simultaneously is considered together. In addition to these, there are also studies that consider total workforce and resource cost together (Pereira 2018) and the number of resources and cycle time minimization together (Alakas et al. 2020). According to Table 1, there is a study (Mete and Agpak 2013) considering the total cost function expressed as the sum of the resource cost used and the station openning cost.

Studies in the literature are classified into three different groups regarding resource constraint type as dedicated, alternative, and concurrent. According to the studies summarized in Table 1, the most considered resource constraint is the dedicated type (i.e., 11 works). In only two studies (Ağpak and Gökçen 2005; Alakas et al. 2020), the alternative resource type has been considered together with the dedicated resource type. Three studies (Corominas et al. 2011; Mete and Agpak 2013; Alakaş 2021) can be given as an example of the concurrent resource type. As a result, studies considering alternative and concurrent resource types are still insufficient in the literature.

When RCALBP studies are evaluated in terms of solution method, it is seen that exact (e.g., MP, CP), heuristic (e.g., SRA), and meta-heuristic (e.g., GA) solution methods exist. According to the results specified in Table 1, the most used solution method in the RCALBP solution is the genetic algorithm (Chen et al., 2018; Jusop & Ab Rashid, 2017; Kamarudin & Ab. Rashid, 2017; Khalib et al., 2019; Quyen et al., 2017). (Ağpak and Gökçen 2005) is the first study to use the exact method in solving the problem. Also, the studies using the mathematical model as a solution method can be given as an example(Corominas et al. 2011; Mete and Agpak 2013; Ogan and Azizoglu 2015; Pereira 2018). It is possible to list the heuristic algorithms used in the solution of the problem as follows: shortest route algorithm (Kao et al. 2010); rank positional weight-based heuristic (Kao et al. 2011). The only two studies that uses constraint programming to solve the problem (Alakas et al. 2020; Alakaş 2021). When the results of the literature research given in Table 1 are examined, it can be easily seen that heuristic methods are often used to solve the RCALBP. Although exact methods such as mathematical programming have been used to solve the problem, the study using exact methods such as constraint programming, is still insufficient.

According to the literature research results mentioned above, this study focused on RCALBP with the following features: U-shape line, general case resource type, minimization of total cost, concurrent resource constraint type, and constraint programming solution method.

Table 1 is about here

Studies in the literature where CP is used in the ALBP solution are given in Table 2. The first column of Table 2 shows the study, the line shape between the 2nd and 4th columns, the model type between the 5-6th columns, the objective function between the 7-12, and the last column show the additional characteristics. The literature will be examined in terms of this classification.

In the solution of ALBP, CP is mostly used for straight line types. There are three studies that about U-shape (Bukchin and Raviv 2018; Pınarbaşı 2021a, b) and one two-sided (Kizilay and Çil 2020) ALBP study using CP as a solution method. There is a great need in the literature for studies to be carried out outside of straight assembly lines.

According to the number of models considered in the ALBP problem in the literature, there are two types of studies, single and mixed models. When the studies in Table 2 are examined, single-model ALBP was considered in most studies, and mixed model studies are relatively few. Nevertheless, considering the different ALBP characteristics, it can be said that the use of CP in the single model ALBP solution is still insufficient.

When the literature is evaluated in terms of the objective functions, it is seen that the number of stations and cycle time minimization were taken into account in most of the studies. In addition, total worker minimization (Çil and Kizilay 2020; Kizilay and Çil 2020), number of resource minimization (Alakas et al. 2020), and total completion time minimization (Öztürk et al. 2013, 2015) were used as objective functions. However, there is no study that takes into account the minimization of the station opening cost and the cost of the resource usage simultaneously.

When CP studies in the literature are evaluated, there is no study dealing with the total cost minimization of single model U-shape ALBP. For this reason, U-type ALBP was taken into consideration in this study. Two new CP models using different methods for modeling resource constraints are proposed.

Table 2 is about here

3. Problem definition and assumptions

The labor or machine required to perform a task is assumed as the resource in assembly line literature. Along with a task, the resources that achieve the task must also be assigned to the station. The problem in which the assignment of tasks and resources to the station is handled together is defined as the RCALBP (Ağpak and Gökçen 2005). In the early versions of RCALBP, it was assumed that only one resource was used to accomplish a task i. In the developed version of the problem, alternative resource groups are used to perform a task. This problem type is defined as GRCALBP (Corominas et al. 2011). In this problem, it is assumed that different numbers of A, B, and C type resources are used to perform a task i. For example, suppose task i can be performed using combinations of 3A, 2A and 2B or 3B and C resources. If task i is assigned to a station, one of these resource combinations must also be assigned to the related station. Otherwise, the task cannot be performed and the assembly process cannot be done

The assumptions for RCALBP discussed in the study are assumed as follows:

- Each task is not divisible and should be assigned to a station.
- There is a precedence relationship restrictions between task pairs and these constraints should be satisfied in problem solving.
- Task times are predetermined deterministic constants.
- The cycle time is known and fixed.
- U-shaped line layout is considered.
- Numbers of different and alternative resource types have been taken into consideration.

- The number and types of references have been previously determined.
- Resources costs and station opening cost is known.
- Once a task is assigned to a station, the resources associated with that task must be assigned to that station.

4. Constraint programming models for U-GRCALBP

4.1. Overview of CP

This study is about the development of CP models in the solution of GRCALBP. CP is an effective problem solving method used in the solution of many optimization problems such as scheduling, timetabling etc. in the literature (Gür et al. 2019).

The CP search algorithm produces solutions by using the decision variables, the domains of these variables and the constraints between these decision variables. In short, the search algorithm works as follows. First, the search algorithm determines the domain of each variable before it starts searching for a solution. It then propagates the constraints, reducing the domains of the variables that do not satisfy the constraints. This mechanism is called the domain reduction. After this mechanism, a value is assigned to each decision variable. If there is no variable that does not assign a value, all variables have taken values and the search is terminated. If there is at least one variable that does not assign a value, any variable is selected according to the variable selection rules. A value from the domain of the variable is selected for that variable and the selected value is assigned to the variable. After this stage, the domains of the variables are rearranged using the constraint propagation algorithm. Then it is checked whether there is a decision variable to be assigned value again. Similarly, the search process continues until there are no decision variables for which no value is assigned. After these stages, if there is empty domain of any decision variables, backtracking mechanism is carried out to change the value of that variable. On the contrary, if there is no variable whose domain is empty, the search continues repeatedly. This search algorithm is continued until all decision variables are assigned feasible values and the search is terminated after the solution is reported.

4.2. CP-1 model

Two different proposed constraint programming models are given in this section. In CP models, "or" logical operator is referred to define the alternative resource restrictions. Similarly, “and” operator is used to assign the resources together. These operators could be easily modeled in CP model by using generic form syntax. Likewise, the ability to easily model logical constraints is an important modeling advantage of CP. This advantage has provided great convenience in modeling alternative resource constraints in CP software. Necessary notations for models are given in Table 3. The proposed CP-1 model for U-GRCALBP is as follows:

Table 3 is about here

Objective function:

$$Min \sum_{r=1}^{R} \sum_{j=1}^{m} CR_r * v_{rj} + CS * m \tag{1}$$

Subject to:

$$\sum_{i=1}^{N} (x_i = j \ \text{v} \ x_i = 2m - j) t_i \le ct \qquad \forall j \tag{2}$$

$$x_i \leq x_h \qquad \forall(i,h) \in Pr \tag{3}$$

$$\Big(\big(\alpha_{r1i} * (x_i = j) \leq v_{rj}\big) \vee \big(\alpha_{q1i} * (x_i = j) \leq v_{qj}\big)\Big) \wedge \Big(\big(\alpha_{r2i} * (x_i = j) \leq v_{rj}\big) \vee \big(\alpha_{q2i} * (x_i = j) \leq v_{qj}\big)\Big) \ldots \wedge \Big(\big(\alpha_{rpi} * (x_i = j) \leq v_{rj}\big) \vee \big(\alpha_{qpi} * (x_i = j) \leq v_{qj}\big)\Big) \qquad \forall i,j,r,q \tag{4}$$

$$x_i \in (1, \ldots, s) \qquad \forall i \tag{5}$$

$$v_{rj} \in \big(0, \ldots, \max(\alpha_{rpi})\big) \qquad \forall i,p,r \tag{6}$$

The objective function is the sum of the costs of the resources assigned to the stations and the station opening cost (Eq.1). Eq.2 ensures that the total duration of the tasks assigned to the station does not exceed the cycle time. Eq. 3 guarantees priority relationships. It should be written for each priority relationship. Eq. 4 takes into account resource constraints. If a task i is assigned to station j, it ensures that the resources needed to perform the task i are assigned to the relevant station in desired amounts. With Eq. 5, the domain is defined for the decision variable that holds the station numbers to which the task i can be assigned. The decision variables' domains that keep the number of resources assigned to the station is defined by Eq. 6.

In CP models, decision variables can be desired to be equal to or greater than or less than a value. This logical query takes the value one if the expression is true and zero if not. Equality or inequalities are used in parentheses for logical inquiry in CP. For example, if xi takes the value of j for the expression $(x_i = j)$ in the CP model, it returns one, in other cases, zero. Similarly, a similar query can be made by enclosing the smaller than or greater than expressions in parentheses.

4.3. CP-2 model

In the CP-2 model, resource constraints are changed and the model is formed as follows:

Objective function:

$$Min\sum_{r=1}^{R}\sum_{j=1}^{m} CR_r * v_{rj} + CS * m \tag{7}$$

Subject to:

$$\sum_{i=1}^{N}(x_i = j \ \vee \ x_i = 2m - j)t_i \leq ct \qquad \forall j \tag{8}$$

$$x_i \leq x_h \qquad \forall (i,h) \in Pr \tag{9}$$

$$\begin{aligned}&\Big(\big((x_i = j) \leq (v_{rj} \geq \alpha_{r1i})\big) \vee \big((x_i = j) \leq (v_{qj} \geq \alpha_{q1i})\big)\Big) \wedge \Big(\big((x_i = j) \leq (v_{rj} \\ &\qquad \geq \alpha_{r2i})\big) \vee \big((x_i = j) \leq (v_{qj} \geq \alpha_{q2i})\big)\Big) \ldots \wedge \Big(\big((x_i = j) \leq (v_{rj} \\ &\qquad \geq \alpha_{rpi})\big) \vee \big((x_i = j) \leq (v_{qj} \geq \alpha_{qpi})\big)\Big) \qquad \forall i,j,r,q\end{aligned} \tag{10}$$

$$x_i \in (1, \ldots, s) \qquad \forall i \tag{11}$$

$$v_{rj} \in \big(0, \ldots, \max(\alpha_{rpi})\big) \qquad \forall i,p,r \tag{12}$$

As explained in the CP-1 model, the objective function is the minimization of the sum of resource usage and station installation costs (Eq. 7). Eq 8 is the cycle time constraint and Eq 9 is the constraint of priority relationships. In Eq 11 and 12, the range of values that the decision variables can take is defined. With Eq 10, if a task i is assigned to station j, it ensures that sufficient resources required by the task are assigned to the relevant station.

5. Illustrative example

In this section, the solution of the proposed model is explained through an illustrative example. The problem in the literature (Mete and Agpak 2013) was used as an illustrative example. The researchers solved this example as two-sided in that study. In this study, it was solved by assuming that it is a one-sided and U-type problem. Resource constraints are taken the same as in the study. Data on precedence relations, task processing times, and tasks' resource needs of the problem are given in Table 4. Also precedence diagram of the illustrative example is given in Figure 1. Resource costs were accepted as 12, 8, and 5 units for A, B, and C resources. Station opening cost was taken as 10 units. The cycle time is taken as 5.

Table 4 is about here

Figure 1 is about here

Considering the data in Table 4, resource constraints related to tasks 1, 3, and 6 can be modeled as given in Eq 8, 9, and 10, respectively for CP-1 model.

$$\Big(\big(2 * (x_1 = j) \leq v_{Aj}\big) \vee \big(1 * (x_1 = j) \leq v_{Bj}\big)\Big) \wedge \Big(\big(2 * (x_1 = j) \leq v_{Aj}\big) \vee \big(2(x_1 = j) \leq v_{Cj}\big)\Big) \tag{13}$$

$$\big(5 * (x_3 = j) \leq v_{Aj}\big) \wedge \Big(\big(2 * (x_1 = j) \leq v_{Bj}\big) \vee \big(4 * (x_1 = j) \leq v_{Cj}\big)\Big) \tag{14}$$

$$\Big(\big(4 * (x_1 = j) \leq v_{Aj}\big) \vee \big(4 * (x_6 = j) \leq v_{Bj}\big)\Big) \wedge \Big(\big(5 * (x_6 = j) \leq v_{Cj}\big)\Big) \tag{15}$$

Unlike the CP 1 model, resource constraints for the above tasks are given as in equations 11, 12 and 13 respectively for the CP-2 model.

$$\left(\left((x_1 = j) \leq (v_{Aj} \geq 2\right) \text{ v } \left((x_1 = j) \leq (v_{Bj} \geq 1)\right)\right) \wedge \left(\left((x_1 = j) \leq (v_{Aj} \geq 2)\right) \text{ v } \left((x_1 = j) \leq (v_{Cj} \geq 2)\right)\right) \tag{16}$$

$$\left((x_3 = j) \leq (v_{Aj} \geq 5)\right) \wedge \left(\left((x_1 = j) \leq (v_{Bj} \geq 2)\right) \text{ v } \left((x_1 = j) \leq (v_{Cj} \geq 4)\right)\right) \tag{17}$$

$$\left(\left((x_1 = j) \leq (v_{Aj} \geq 4)\right) \text{ v } \left((x_6 = j) \leq (v_{Bj} \geq 4)\right)\right) \wedge \left(\left((x_6 = j) \leq (v_{Cj} \geq 5)\right)\right) \tag{18}$$

An example problem is solved with two CP models called as CP-1 and CP-2. The result of the CP-1 model and the CP-2 model are given in Table 5. In both models, the station number is five, and the total number of resources used by all stations is 29. The theoretical minimum station number is five, and the number of the station is obtained five with the two models. When the results are evaluated in terms of the number of stations, the optimum result has been reached with two CP models.

Table 5 is about here

6. Numerical experiment

Data sets from Alakas, Pinarbasi, and Yuzukirmizi (2020) were used to show the effectiveness of the proposed models in the experimental studies. While task times and priority relationships are considered to be the same as data sets, resource constraints have been generated by the authors for U-GRCALBP in a way that does not prevent the problem from reaching a feasible solution. It is assumed that there are 2 or 4 resource types in the data sets, and the numbers of tasks are range from 21 to 70 in the data set. Resource costs are taken as 12, 8, 5, and 10, and station cost is taken as 10. Both CP models are modeled and solved with ILOG Cplex Optimization v12.10. A PC with Intel CoreTM i5, 2.30 GHz processor and 8 GB of RAM are used to perform the model solution. The execution time limit is assumed 3600 seconds for both CP models. Numerical results reported according to the number of two and four sources are given in Table 6 and Table 7. Since the problem addressed in the study was not discussed in another study in the literature, numerical results were given by comparing the proposed models. While making comparisons in the tables, performance criteria are highlighted as shown below. If one of the models is superior to the other in terms of the relevant criterion, that criterion is emphasized as stated.

Italic	Minimum number of stations
Bold	Minimum number of resources
Highlighted	Minimum total cost

Table 6 is about here

Table 7 is about here

Table 6 contains the comparative results of problem sets with two resource types. When the models' results are evaluated in terms of the minimum number of stations, it can be said that the CP-1 model has relatively better results than the CP-2 model. However, in most of the problem sets, both models have reached the same station number value. When the number of resources obtained is evaluated, it can be said that the CP-2 model obtains the minimum number of resources in more problem sets. However, in general, solutions with the same number of resources have been produced in both models. It cannot be said that both models establish a clear superiority to each other in total cost results.

The results of the problem sets with four resource types are reported in Table 7. When the results were evaluated according to the criteria of the number of stations, both models showed similar performance. Similarly, considering minimizing the number of resources used, neither model could outperform each other. However, considering the total cost criteria, it is seen that the CP-1 model is superior to the CP-2 model.

Table 8 shows the number of problems where both models provide superiority to each other according to each performance criterion. It is easily seen in the results that the more the number of sources, the more obvious the superiority between the models. While both models show similar performance in problems with two resource types, it is seen that the CP-1 model performs better than the criterion of minimizing the total cost in problems with four resource types.

Table 8 is about here

Some remarkable results were also obtained in numerical results. Considering the Tongue c = 502 problem in Table 6, while the results of the CP-1 model are m = 8, the total number of used resources is 49, and the total cost is 556, for the CP-2 model, these values are 9, 48, and respectively. It was reported as 550. More stations were opened with the CP-2, and fewer resources were used, but it achieved a low-cost result. We see the reverse of this situation in the Kilbridge c = 69 problem in Table 7. Although the CP-2 model obtained more station numbers and fewer resource numbers, the CP-1 model obtained a lower cost value with less station number and more resource number. When we consider the Sawyer c = 31 problem, the CP-1 model resulted in a lower cost, although the CP-1 model obtained more station numbers and more resources than the CP-2 model. This situation shows a trade-off between the number of opened stations and the number of resources to reach the minimum cost. However, it depends on the given station opening and resource usage costs, and resource constraints.

Another interesting result obtained from the results is as follows. When the Mitchell c = 25 problem is examined in Table 7, it is seen that both models reach the same number of stations and resources. However, the total cost values obtained by both models are 302 and 306, respectively. In other words, the CP-1 model has achieved a lower cost result. This situation is related to the types of resources required to perform the tasks considered in the problem and these resources' costs.

When the proposed CP models are evaluated in terms of solution time, the results obtained are given in Table 9. It can be said that the CP-1 model is relatively better in terms of average CPU time for problem sets with two resources. When the maximum, minimum, and standard deviation values are examined, it can be said that both models require a long solution time for solving the problem sets with two resources. Both models could not obtain the optimum solution in less than the solution time limit for all problem sets with four resources.

Table 9 is about here

When the results are evaluated in general, it can be said that the proposed CP models are effective solutions to the problem. The CP-1 model performs better in solving the problem, especially with the increase in the number of resource types.

The comparative results of the theoretical number of stations obtained on a U-type line without resource constraints with the results given in Table 6 and Table 7 are given in Table 10. The purpose of this comparison is to evaluate whether resource constraints cause a significant increase in the number of stations. The gap value reported in the table is given in Equation 19. According to the results reported in table 8, the biggest gap value Kilbridge c = 110 was obtained in four resource-constrained problems. The smallest gap was obtained as 0.00 in problem sets where the theoretical cycle time and the number of stations obtained in many problems are the same. Mean gap values were obtained as 7.06% and 8.91% respectively for the problems with two and four resource types for the CP-1 model, while 9.59% and 8.45% for the CP-2 model. These values show that the number of stations obtained with both proposed models is acceptable according to the theoretical number of stations obtained without resource constraints.

$$\%gap = \frac{(Result\ of\ proposed\ model - Therotical\ \#station)}{Result\ of\ proposed\ model} \times 100 \tag{19}$$

Table 10 is about here

Considering all these results, managers should make the resources used for the tasks as common as possible to establish the line more effectively. Thus, the assembly line will be established by evaluating more assignment alternatives. At the same time, the number of resource types needed can be reduced by collecting resources. Thus, the solution of the problem will be easier.

7. Conclusion and future research directions

In this study, two innovative and effective exact modeling approaches are presented for the resource-constrained U-shaped assembly line balancing problem. General and concurrent resource constraints are considered as resource constraint types. The proposed models aim to minimize the sum of the station opening cost and the costs of the resources used. Two different

CP models based on different modeling of resource constraints are proposed to solve the problem. Although constraint programming is a new modeling technique for the solution to the ALBP in the literature, it is a frequently used method for solving combinatorial problems such as scheduling, assignment problems, and car sequencing.

The performance of the proposed models was tested with 20 literature problems. While task times and priority relations are preserved in the literature problems, resource constraints have been produced by the authors with different reference numbers. In addition to the total cost, the number of stations obtained, the number of resources, and CPU time were also considered as performance criteria. According to the numerical experimental results, the proposed models showed a remarkable performance in solving the problem. Although it can be said that both models show similar performance in general, it can be said that the CP-1 model is more superior in terms of obtaining the minimum total cost with the increase in the number of resources. The models performed similarly in terms of CPU time. However, with the increase in the number of resources, the time needed to solve the problem also increases. Besides these results, it has been shown that model solutions cause interesting trade-offs among performance criteria. Both models showed notable performance in comparison with the theoretical number of stations for a U-shape ALBP without resource constraints.

Future study suggestions can be given as follows: ALBP with different line configurations (e.g., parallel, two-sided) CP models can be developed for different source types. ALBP, where task times are stochastic, would also be a notable future study. Moreover, it will be a very innovative future work to develop models in which the total cost will be minimized for the solution of ALBP, which uses Industry 4.0, the internet of things, and augmented reality based resources.

Declarations:
Funding: No funding is received
Conflict of interest: The author declares that he has no conflict of interest.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.
The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Authors Contributions:
Author 1: study conception and design, data collection, establishment of constraint programming model, analysis and interpretation of results, and manuscript preparation.
Author 2: study conception and design, literature review, establishment of constraint programming model, analysis and interpretation of results, and manuscript preparation.

References

Ağpak K, Gökçen H (2005) Assembly line balancing: Two resource constrained cases. Int J Prod Econ 96:129–140. https://doi.org/10.1016/j.ijpe.2004.03.008

Alağaş HM, Pınarbaşı M, Yüzükırmızı M, Toklu B (2016) A constraint programming model for mixed model type 2 assembly line balancing problem. Pamukkale Univ J Eng Sci 22:340–348. https://doi.org/10.5505/pajes.2016.93276

Alağaş HM, Yüzükırmızı M, Türker AK (2013) Balancing stochastic assembly lines using constraint programming and closed queueing networks. J Fac Eng Arch Gazi Univ 28:231–240

Alakaş HM (2021) General resource-constrained assembly line balancing problem: conjunction normal form based constraint programming models. Soft Comput 25:6101–6111

Alakas HM, Pinarbasi M, Yuzukirmizi M (2020) Constraint programming model for resource-constrained assembly line balancing problem. Soft Comput 24:5367–5375

Alakas HM, Toklu B (2020) Problem Specific Variable Selection Rules for Constraint Programming: A Type II Mixed Model Assembly Line Balancing Problem Case. Appl Artif Intell 34:564–584

Bockmayr A, Pisaruk N (2001) Solving assembly line balancing problems by combining IP and CP. In: In: Sixth Annual Workshop of the ERCIM Working Group on Constraints,

Bukchin Y, Raviv T (2018) Constraint programming for solving various assembly line balancing problems. Omega 78:57–68. https://doi.org/10.1016/j.omega.2017.06.008

Chen JC, Chen C-C, Su L-H, et al (2012) Assembly line balancing in garment industry. Expert Syst Appl 39:10073–10081

Chen Y-Y, Cheng C-Y, Li J-Y (2018) Resource-constrained assembly line balancing problems with multi-manned workstations. J Manuf Syst 48:107–119. https://doi.org/10.1016/j.jmsy.2018.07.001

Çil ZA, Kizilay D (2020) Constraint programming model for multi-manned assembly line balancing problem. Comput Oper Res 124:105069

Corominas A, Ferrer L, Pastor R (2011) Assembly line balancing: general resource-constrained case. Int J Prod Res 49:3527–3542. https://doi.org/10.1080/00207543.2010.481294

Del Valle C, Márquez AA, Gasca RM, Toro M (2003) On selecting and scheduling assembly plans using constraint programming. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, pp 1329–1336

Eghtesadifard M, Khalifeh M, Khorram M (2020) A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990–2017. Comput Ind Eng. https://doi.org/10.1016/j.cie.2019.106182

Gür Ş, Eren T, Alakaş HM (2019) Surgical Operation Scheduling with Goal Programming and Constraint Programming: A Case Study. Mathematics 7:251. https://doi.org/10.3390/math7030251

Jusop M, Ab Rashid MFF (2017) Optimization of assembly line balancing with resource constraint using NSGA-II: a case study. Int J Appl Eng Res 12:1421–1426

Kamarudin NH, Ab. Rashid MFF (2017) Assembly line balancing with resource constraints using new rank-based crossovers. J Phys Conf Ser 908:012059. https://doi.org/10.1088/1742-6596/908/1/012059

Kao H-H, Yeh D-H, Wang Y-H (2011) Resource constrained assembly line balancing problem solved with ranked positional weight rule. Rev Econ Financ 5:71–80

Kao H-H, Yeh D-H, Wang Y-H, Hung J-C (2010) An optimal algorithm for type-I assembly line balancing problem with resource constraint. Afr J Bus Manag 4:2051–2058

Khalib KH, Kamarudin NH, Rashid MFFA (2019) Evaluation of rank-based crossovers to optimize real-life assembly line balancing with resource constraint. In: IOP Conference Series: Materials Science and Engineering. p 12014

Kizilay D, Çil ZA (2020) Constraint programming approach for multi-objective two-sided assembly line balancing problem with multi-operator stations. Eng Optim 1–16

Mete S, Agpak K (2013) Multi objective generalized resource constrained two sided assembly line balancing problem and computational analysis. J Fac Eng Arch Gazi Univ 28:567–576

Ogan D, Azizoglu M (2015) A branch and bound method for the line balancing problem in U-shaped assembly lines with equipment requirements. J Manuf Syst 36:46–54. https://doi.org/10.1016/j.jmsy.2015.02.007

Öztürk C, Tunalı S, Hnich B, Örnek A (2015) Cyclic scheduling of flexible mixed model assembly lines with parallel stations. J Manuf Syst 36:147–158. https://doi.org/10.1016/j.jmsy.2015.05.004

Öztürk C, Tunalı S, Hnich B, Örnek MA (2013) Balancing and scheduling of flexible mixed model assembly lines. Constraints 18:434–469. https://doi.org/10.1007/s10601-013-9142-6

Pastor R, Ferrer L, García A (2007) Evaluating optimization models to solve SALBP. In: In: Gervasi O, Gavrilova ML (eds) Proceeding of international conference on computational science and its applications, ICCSA, Lecture notes in computer science. Springer. Berlin, pp 791–803

Pereira J (2018) Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem. Int J Prod Res 56:3994–4016. https://doi.org/10.1080/00207543.2018.1427899

Pinarbasi M, Alakas HM, Yuzukirmizi M (2019) A constraint programming approach to type-2 assembly line balancing problem with assignment restrictions. Assem Autom 39:813–826. https://doi.org/10.1108/AA-12-2018-0262

Pınarbaşı M (2021a) New chance-constrained models for U-type stochastic assembly line balancing problem. Soft Comput 25:9559–9573. https://doi.org/10.1007/s00500-021-05921-z

Pınarbaşı M (2021b) New mathematical and constraint programming models for U-type assembly line balancing problems with assignment restrictions. Eng Optim 1–16. https://doi.org/10.1080/0305215X.2021.1921171

Pınarbaşı M, Alakaş HM (2020) Balancing stochastic type-II assembly lines: chance-constrained mixed integer and constraint programming models. Eng Optim 52:2146–2163. https://doi.org/10.1080/0305215X.2020.1716746

Pınarbaşı M, Yüzükırmızı M, Toklu B (2016) Variability modelling and balancing of stochastic assembly lines. Int J Prod Res 54:5761–5782. https://doi.org/10.1080/00207543.2016.1177236

Quyen NTP, Chen JC, Yang C-L (2017) Hybrid genetic algorithm to solve resource constrained assembly line balancing problem in footwear manufacturing. Soft Comput 21:6279–6295. https://doi.org/10.1007/s00500-016-2181-3

Sivasankaran P, Shahabudeen P (2014) Literature review of assembly line balancing problems. Int J Adv Manuf Technol 73:1665–1694

Topaloglu S, Salum L, Supciller AA (2012) Rule-based modeling and constraint programming based solution of the assembly line balancing problem. Expert Syst Appl 39:3484–3493. https://doi.org/10.1016/j.eswa.2011.09.038

Figures

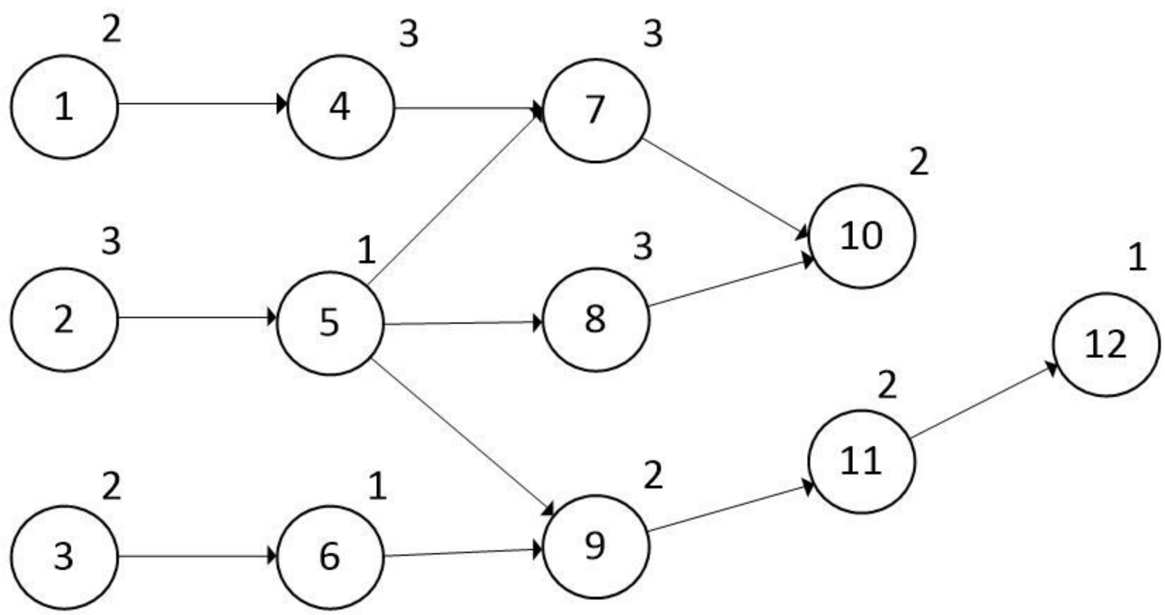

Figure 1

Precedence diagram of the illustrative example

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Table3.jpg