Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2007, Article ID 10106, 12 pages
doi:10.1155/2007/10106

*Research Article*

# A Network Model for Parallel Line Balancing Problem

Recep Benzer, Hadi Gökçen, Tahsin Çetinyokuş, and Hakan Çerçioglu

Received 30 August 2006; Revised 25 December 2006; Accepted 19 July 2007

Recommended by Stanley B. Gershwin

Gökçen et al. (2006) have proposed several procedures and a mathematical model on single-model (product) assembly line balancing (ALB) problem with parallel lines. In parallel ALB problem, the goal is to balance more than one assembly line together. In this paper, a network model for parallel ALB problem has been proposed and illustrated on a numerical example. This model is a new approach for parallel ALB and it provides a different point of view for interested researchers.

## 1. Introduction

Assembly lines are flow-oriented production systems which still are typical in the industrial production of high-quality standardized commodities and even gain importance in low-volume production of customized products. Among the decision problems which arise in managing such systems, ALB problems are important tasks in medium-term production planning [1].

ALB relates to a finite set of tasks, each having a task time and a set of precedence relations, which specify the permissible orderings of the tasks. One of the problems in organizing mass production is performing way of group task on workstations so as to achieve the desired level of performance. Line balancing is an attempt to allocate equal amounts of work to the various workstations along the line. The fundamental line balancing problem is how to assign a set of tasks to an ordered set of workstations, such that the precedence relations and some performance measures (minimizing the number of workstation, cycle time, idle time, etc.) are satisfied [2].

The studies related to the assembly line can be classified into general groups: traditional assembly lines (with single and multi/mixed products) and U-type assembly lines (with single and multi/mixed products). The ALB problem has been widely studied since the first analytical statement of the ALB problem was published in mathematical form by Salveson [3]. Over the last five decades, many heuristics and optimal procedures have been proposed for the solution of ALB problem. For the studies on traditional assembly line balancing, the review papers of Baybars [4], Ghosh and Gagnon [2], Erel and Sarin [5], Scholl and Becker [6], and Becker and Scholl [1] can be seen. In addition, the papers of Miltenburg and Wijngaard [7], Urban [8], Scholl and Klein [9], Ohno and Nakade [10], Miltenburg [11], Sparling and Miltenburg [12], Miltenburg [13], and Guerriero and Miltenburg [14] may also be investigated for U-type line balancing. Although literature on traditional and U-type ALB is rather extensive, the studies on parallel lines are quite little. In designing the parallel assembly lines, Süer and Dagli [15] have suggested heuristic procedures and algorithms to dynamically determine the number of lines and the line configuration. Also, Süer [16] has studied alternative assembly line design strategies for a single product. Other researches involving parallel workstation have focused on the simple assembly line balancing problem [17] and mixed-model production line balancing problem [18–20]. These studies on parallel lines are logically different from the approach of Gökçen et al. [21]. The new problem presented by Gökçen et al. [21] has been derived from the traditional and U-type ALB problem where more than one assembly line is balanced with common resources.

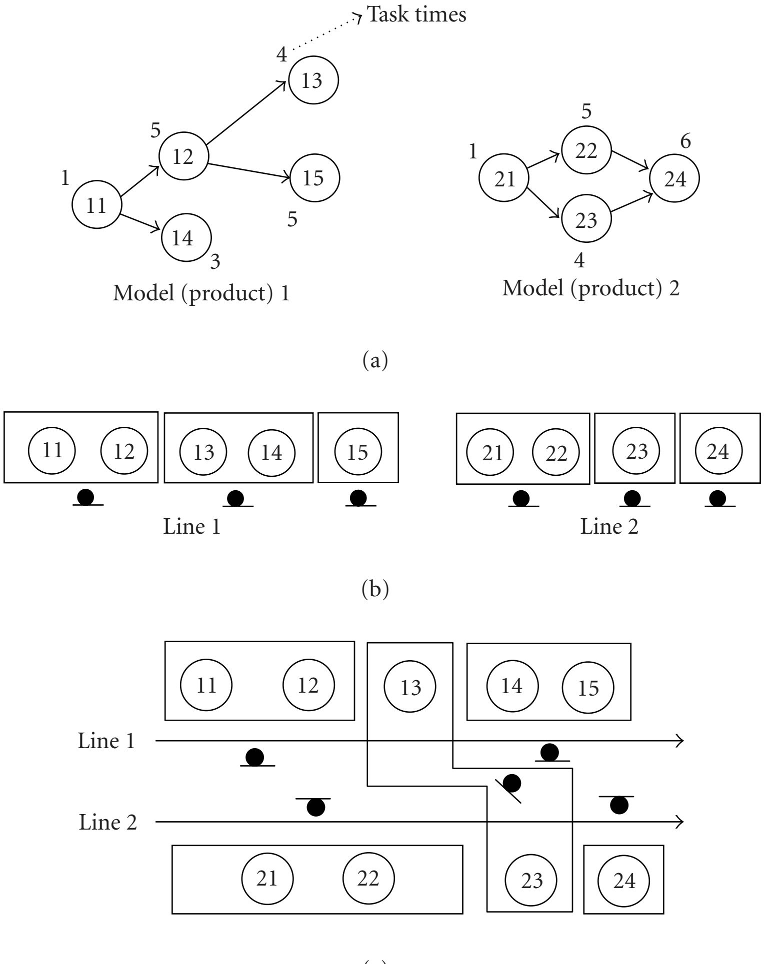
In this paper, a network model is developed for this new problem called parallel ALB.

## 2. Balancing of the parallel lines

It is the most common case in industry that more than one line (especially two or three lines) produce the same or different types of product at the same time independently. Working of the lines simultaneously with a common resource is very important in terms of resource minimization [21]. The goal of the problem (balancing of the parallel lines) presented by Gökçen et al. [21] is to balance more than one assembly line together. That is, it will be possible to assign task(s) from each line to a multiskilled operator. As a result, it is inevitable to minimize the total idle times of the lines. For this purpose, Gökçen et al. [21] have developed two procedures and a mathematical model and tested the models on well-known problems in the line balancing literature.

In Figure 2.1(a), precedence diagrams for two different products (two assembly lines) and the line balancing results of traditional and parallel lines are given. The numbers within the nodes represent tasks, and the arrow (or arcs) connecting the nodes specifies the precedence relations. The numbers next to the nodes represent task performance times. When each product in the problem is balanced with a cycle time of 8, it can be seen that all tasks are performed at 6 workstations in the traditional assembly line (Figure 2.1(b)), whereas all tasks are performed at 5 workstations in parallel assembly line (Figure 2.1(c)).

As seen in Figure 2.1(c), workstations can include tasks located on different parts of the two production lines. For example, one of the workstation consists of tasks 13 and 23, where 13 is located in line 1 while 23 is located in line 2.



(a)

(b)

(c)

Figure 2.1. (a) Precedence diagrams for product 1 and product 2, line balancing results of (b) traditional and (c) parallel.

## 3. Shortest-route formulation

First shortest-route model of the traditional single-model ALB problem has been presented by Klein [22]. The network had directed arcs representing possible assignments of tasks to workstations, and each path from source to sink represented a possible line design. Then, Gutjahr and Nemhauser [23] have developed an algorithm to solve the single-model version of the problem based on finding the shortest route in a finite-directed network. Mansoor [24] has suggested an adjustment to the Gutjahr and Nemhauser [23] algorithm to obtain the optimal solution after considering only a fraction of the shortest-route calculations. Roberts and Villa [25] have extended the Gutjahr and Nemhauser [23] algorithm to solve the mixed-model version of the problem. Chakravarty and Shtub [26] have presented a shortest-route formulation for mixed model, unpaced line in which setup, inventory holding, and labor costs were considered [27]. Erel and Gökçen [27] have developed a shortest-route formulation for mixed-model assembly lines based on the Gutjahr and Nemhauser's [23] algorithm. The model was considerably superior to the model of Roberts and Villa [25]. Finally, Gökçen et al. [28] have presented a shortest-route formulation for the U-type ALB problem.

The network model proposed for the parallel ALB problem in this paper is based on the Gutjahr and Nemhauser's [23] algorithm. In the network model, we assumed that the task performance times are known constant, precedence relations of tasks are known, operators worked in each workstation of each line are multiskilled, only one product is produced on each assembly line, and operators can work on each side of the line.

The mathematical model developed by Gökçen et al. [21] for balancing of the parallel assembly lines is given below.

$$\text{Objective function:}\quad \text{Min}\sum_{k=1}^{K_{\max}} z_k. \tag{3.1}$$

Constraints:

$$\sum_{k=1}^{K_{\max}} x_{hik} = 1 \quad \text{for } i = 1,\dots,n_h,\ h = 1,\dots,H, \tag{3.2}$$

$$\sum_{i=1}^{n_h} t_{hi}x_{hik} + \sum_{i=1}^{n_{h+1}} t_{(h+1)i}x_{(h+1)ik} \le Cz_k \quad \text{for } k = 1,\dots,K_{\max},\ h = 1,\dots,H-1, \tag{3.3}$$

$$\sum_{i=1}^{n_h} x_{hik} - \|M_{hk}\|_{hk}^{U} \le 0 \quad \text{for } h = 1,\dots,H,\ k = 1,\dots,K_{\max}, \tag{3.4}$$

$$U_{hk} + U_{(h+a)k} = 1 \quad \text{for } h = 1,\dots,H-2,\ a = 2,\dots,H-h,\ k = 1,\dots,K_{\max}, \tag{3.5}$$

$$\begin{aligned} &\sum_{k=1}^{K_{\max}} \left(K_{\max} - k + 1\right)\left(x_{hrk} - x_{hsk}\right) \ge 0 \quad \forall (r,s) \in P_h \\ &x_{hik}, z_k, U_{hk} \in \{0,1\} \quad \text{for } h,i,k. \end{aligned} \tag{3.6}$$

Constraint (3.2) ensures that all tasks are assigned to a station and each task is assigned only once. Constraint (3.3) ensures that the work content of any station does not exceed the cycle time. Constraints (3.4) and (3.5) ensure that an operator worked at station $k$ and line $h$ can do task(s) from only one adjacent line (i.e., operator in line $h$ can do the tasks in line $h+1$ or $h-1$). Constraint (3.6) ensures that the precedence constraints are not violated on the line h precedence diagrams. As a result of objective function, the number of workstations will be minimized [21].

**3.1. Network model.** The network model proposed here is based on the shortest-route model developed by Gutjahr and Nemhauser [23] for single-model ALB problem. The network model includes developing a finite-directed network for which the arcs represent

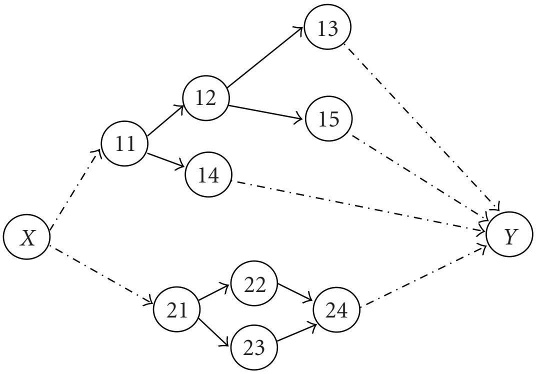


Figure 3.1. Combined precedence diagrams which are tied with null nodes.

workstations in the assembly line and the nodes correspond to possible first workstation assignments of tasks. The arc lengths are the idle times of workstations. Thus, the optimization procedure is to find the shortest path in the network or to find the minimum number of arcs. Node generation, arc construction, and finding the shortest path are given below in detail.

*3.1.1. Generation of nodes.* Combining more than one precedence diagram into one diagram will ease the state generation. For this reason, precedence diagrams of products should be connected with null starting and ending nodes as given in Figure 3.1. The null nodes $(X, Y)$ will not be considered in state generation. Each node in the network model (shortest-route model) represents a state. A state is a collection of tasks that can be processed without prior completion of any other tasks in any order that satisfies the precedence relations.

In state or node generation process, the following properties should be satisfied.

(i) No state elements can be generated as duplicate.
(ii) All sets generated are states.
(iii) Every state is generated [23].

The node generation process used in the developed model is similar to the Gutjahr and Nemhauser's [23] node generation process. The node generation process for the parallel ALB problem can be defined as follows. The empty set is considered as the first state generated. The tasks that are available for assignment (i.e., the task without any predecessor except the null nodes in the combined precedence diagram) are unmarked immediate followers. These tasks are placed on stage 1 and are considered as marked tasks. For the initial stage, all sets of task combination related to marked/unmarked tasks are generated. Each set is defined as a state (S). The unmarked immediate followers of a state are augmented to the current stage to construct the states of the next stage. The augmentation of states and the corresponding unmarked immediate followers are performed in stages. For any state S of stage $s$, unmarked immediate followers are placed in a list called $F(S)$. Let $T \subset F(S)$, the $S \cup T$ is a state for stage $s+1$. For each state of stage $s$, the unmarked immediate followers are determined and placed as marked tasks for stage $s+1$. When all tasks are marked or $F(S)$ is empty for the current stage, the node generation process is completed. All possible feasible states can be generated with this node generation process. The final node/state in the network consists of all tasks.

After constructing the state generation table, some states should be removed. This is necessary due to nature of the parallel ALB problem. When there are more than two lines, it should not be allowed for the operator that works on line 1 to work on line 3 because there is another line, line 2, exists between line 1 and line 3. In other words, tasks of line 1 and line 3 should not be assigned to the same operator. When states are being produced, these unsuitable states are also possible to be produced. So before setting the shortest route, these unsuitable states should be deleted. The following conditions should be attained in order to do this.

(i) Assignment of tasks from neighbor line is acceptable: $\{(h,i),(h+1,i)\}$.

(ii) Assignment of tasks from lines that are not neighbors is not acceptable: $\{(h,i),(E,i)\}$, $E \in \{h+2,h+3,\dots,H\}$.

$(h,i)$: $h$ line, $i$ task.

*3.1.2. Constructing the arcs and finding the shortest path.* Each state generated in the state generation process corresponds to the nodes in the directed network. Let $G_i$, $i = 1,\dots,r$, represent the set of tasks in node $i$ ($r$ is the total number of nodes). $G_0 = 0$ and $G_r$ is the set of all tasks in the combined precedence diagram. Note that, no arc enters node 0 and no arc leaves node $r$. Also, $T(G_i) = \sum_{j\in G_i} t_j$ is the total task time in $G_i$. The path(s) from node 0 to node $r$ in the network can be constructed as follows: the aim is to find the shortest path(s) of the network. Finding the shortest path from node 0 to node $r$ can be achieved by finding any path(s) from node 0 to node $r$ with the least number of arcs. A path with the least number of arcs from node 0 to node $r$ is the optimal solution of the problem.

Constructing the network begins with the initial node (node 0). All arcs are branched from node 0. If $T(G_i) \le C$, there is an arc from node 0 to node $i$. These nodes are called the first nodes. If $G_i \subset G_j$ and $T(G_j) - T(G_i) \le C$ for node $i$ among the first nodes, an arc to node $j$ is constructed. These nodes are also called the second nodes. Network construction is repeated until node $r$ is reached. Each directed arc $(ij)$ from node $i$ to node $j$ in the network has a length of $[C - T(G_j) + T(G_i)]$. This length denotes the idle time of each workstation. After the network construction is completed, a shortest route (or path) from node 0 to node $r$ is obtained by considering the arc lengths (or idle time of workstation). Each arc in the final network corresponds to the workstation in the assembly line. If the shortest route from node 0 to node $r$ is $(0,i,j,k,r)$, the task assignments to the workstations can be determined as $[(G_r - G_k),(G_k - G_j),(G_j - G_i),(G_i - G_0)]$. Each set represents the tasks that will be assigned to the workstations.

## 4. Illustrative example

Precedence diagrams given in Figure 2.1(a) will be used. Models 1 and 2 have 5 and 4 tasks, respectively. Combined diagram which is obtained by adding the null starting and ending nodes is also depicted in Figure 3.1. The operation of combining the precedence diagrams with null nodes should be done before the node generation process. The cycle

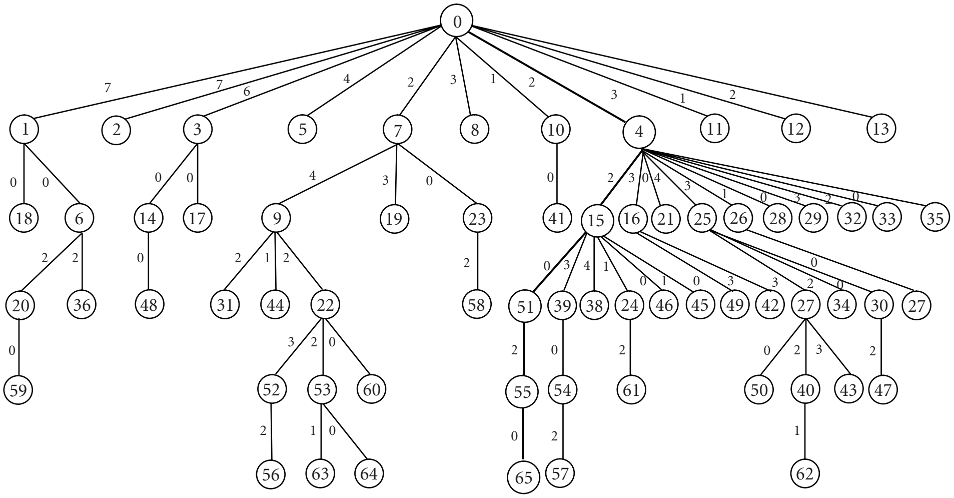


FIGURE 4.1. Resulting network of the example problem.

time of each model is 8. The state generation is shown in Table 4.1. Initially, there are two tasks which are available for assignments in the combined diagram. These tasks are 11 and 21, and are placed as unmarked immediate followers in stage 0. Task 11 and task 21 are considered as marked and they are placed in stage 1. All possible combinations of task 11 and task 21 are determined and placed as state elements for stage 1. Unmarked immediate followers of task 11 are tasks 12 and 14; the unmarked immediate followers of task 21 are tasks 22 and 23, and the unmarked immediate followers of tasks 11 and 21 are tasks 12, 14, 22, and 23. They are placed in list $F(S)$. $F(S)$ list is union of the all unmarked immediate followers subset in stage 1, that is, $\{12,14\} \cup \{22,23\} \cup \{12,14,22,23\} = \{12,14,22,23\}$. These tasks of $F(S)$ are placed in stage 2 as marked tasks. Task 11 is augmented to all subsets of the list $F(S)$ that include tasks 12, 14; task 21 is augmented to all subsets of the list $F(S)$ that include tasks 22, 23, and tasks 11 and 21 are also augmented to all subset of the list $F(S)$ that include tasks 12, 14, 22, 23, to form the states of stage 2. For example, in stage 1, unmarked immediate followers of task 11 are tasks 12 and 14. The generated states are $\{11,12\}$, $\{11,14\}$, and $\{11,12,14\}$. These states correspond to 4th, 5th, and 6th states of stage 2. For each state in stage 2, $F(S)$ list is determined. Because $F(S)$ is empty for all states of stage 3, the procedure is terminated in this stage. The total number of states generated is 65.

The network constructed with the constraint of $T(G_j) - T(G_i) \leq C$ is shown in Figure 4.1. Nodes 1, 2, 3, 4, 5, 7, 8, 10, 11, 12, and 13 define the first nodes, because the constraint of $T(G_i) \leq C$ is satisfied for these nodes. From node 1, two new nodes, from node 3, two new nodes, from node 7, three new nodes, from node 10, one new node, and lastly from node 4, ten new nodes can be constructed with $G_1 \subset G_j$ and $T(G_j) - T(G_1) \leq C$. These new nodes are called as second nodes. For example, state 18 includes tasks of 11, 21, 14, 23 and state time is 9. Because $G_1 \subset G_{18}$ ($\{11\} \subset \{11,21,14,23\}$) and $T(G_{18}) - T(G_1) \leq 8(9 - 1 \leq 8\}$ constraints are satisfied, an arc from node 1 to node 18

Table 4.1. State generation.

| Stage | Marked tasks | State no. | State elements (S) | State times | Unmarked immediate followers |
|---|---|---|---|---|---|
| 0 | | 0 | Ø | 0 | 11 21 |
| 1 | 11 21 | 1 | 11 | 1 | 12 14 |
| 1 | | 2 | 21 | 1 | 22 23 |
| 1 | | 3 | 11 21 | 2 | 12 14 22 23 |
| 2 | 12 14 22 23 | 4 | 11 12 | 6 | 13 15 |
| 2 | | 5 | 11 14 | 4 | — |
| 2 | | 6 | 11 12 14 | 9 | 13 15 |
| 2 | | 7 | 21 22 | 6 | — |
| 2 | | 8 | 21 23 | 5 | — |
| 2 | | 9 | 21 22 23 | 10 | 24 |
| 2 | | 10 | 11 21 12 | 7 | 13 15 |
| 2 | | 11 | 11 21 14 | 5 | — |
| 2 | | 12 | 11 21 22 | 7 | — |
| 2 | | 13 | 11 21 23 | 6 | — |
| 2 | | 14 | 11 21 12 14 | 10 | 13 15 |
| 2 | | 15 | 11 21 12 22 | 12 | 13 15 |
| 2 | | 16 | 11 21 12 23 | 11 | 13 15 |
| 2 | | 17 | 11 21 14 22 | 10 | — |
| 2 | | 18 | 11 21 14 23 | 9 | — |
| 2 | | 19 | 11 21 22 23 | 11 | 24 |
| 2 | | 20 | 11 21 12 14 22 | 15 | 13 15 |
| 2 | | 21 | 11 21 12 14 23 | 14 | 13 15 |
| 2 | | 22 | 11 21 12 22 23 | 16 | 13 15 24 |
| 2 | | 23 | 11 21 14 22 23 | 14 | 24 |
| 2 | | 24 | 11 21 12 14 22 23 | 19 | 13 15 24 |
| 3 | 13 15 24 | 25 | 11 12 13 | 10 | — |
| 3 | | 26 | 11 12 15 | 11 | — |
| 3 | | 27 | 11 12 13 15 | 15 | — |
| 3 | | 28 | 11 12 14 13 | 13 | — |
| 3 | | 29 | 11 12 14 15 | 14 | — |
| 3 | | 30 | 11 12 14 13 15 | 18 | — |
| 3 | | 31 | 21 22 23 24 | 16 | — |
| 3 | | 32 | 11 21 12 13 | 11 | — |

Table 4.1. Continued.

| Stage | Marked tasks | State no. | State elements (S) | State times | Unmarked immediate followers |
|---|---|---|---|---|---|
| 3 | | 33 | 11 21 12 15 | 12 | — |
| 3 | | 34 | 11 21 12 13 15 | 16 | — |
| 3 | | 35 | 11 21 12 14 13 | 14 | — |
| 3 | | 36 | 11 21 12 14 15 | 15 | — |
| 3 | | 37 | 11 21 12 14 13 15 | 19 | — |
| 3 | | 38 | 11 21 12 22 13 | 16 | — |
| 3 | | 39 | 11 21 12 22 15 | 17 | — |
| 3 | | 40 | 11 21 12 22 13 15 | 21 | — |
| 3 | | 41 | 11 21 12 23 13 | 15 | — |
| 3 | | 42 | 11 21 12 23 15 | 16 | — |
| 3 | | 43 | 11 21 12 23 13 15 | 20 | — |
| 3 | | 44 | 11 21 22 23 24 | 17 | — |
| 3 | | 45 | 11 21 12 14 22 13 | 19 | — |
| 3 | | 46 | 11 21 12 14 22 15 | 20 | — |
| 3 | | 47 | 11 21 12 14 22 13 15 | 24 | — |
| 3 | | 48 | 11 21 12 14 23 13 | 18 | — |
| 3 | | 49 | 11 21 12 14 23 15 | 19 | — |
| 3 | | 50 | 11 21 12 14 23 13 15 | 23 | — |
| 3 | | 51 | 11 21 12 22 23 13 | 20 | — |
| 3 | | 52 | 11 21 12 22 23 15 | 21 | — |
| 3 | | 53 | 11 21 12 22 23 24 | 22 | — |
| 3 | | 54 | 11 21 12 22 23 13 15 | 25 | — |
| 3 | | 55 | 11 21 12 22 23 13 24 | 26 | — |
| | | 56 | 11 21 12 22 23 15 24 | 27 | |
| 3 | | 57 | 11 21 12 22 23 13 15 24 | 31 | — |
| 3 | | 58 | 11 21 14 22 23 24 | 20 | — |
| 3 | | 59 | 11 21 12 14 22 23 13 | 23 | — |
| 3 | | 60 | 11 21 12 14 22 23 15 | 24 | — |
| 3 | | 61 | 11 21 12 14 22 23 24 | 25 | — |
| 3 | | 62 | 11 21 12 14 22 23 13 15 | 28 | — |
| 3 | | 63 | 11 21 12 14 22 23 13 24 | 29 | — |
| 3 | | 64 | 11 21 12 14 22 23 15 24 | 30 | — |
| 3 | | 65 | 11 21 12 14 22 23 13 15 24 | 34 | — |

can be constructed. All the tasks in the problem are represented by node 65. This node is first obtained with 5 arcs. As seen from the Figure 4.1, optimal route (or path) is determined as 0–4–15–51–55–65. There are 5 arcs in the path, that is, the optimal solution has 5 workstations in the parallel assembly line. Tasks in $G_{65} - G_{55}$ $\{14, 15\}$ constitute a workstation assignment. Similarly, $G_{55} - G_{51}$ $\{24\}$, $G_{51} - G_{15}$ $\{13, 23\}$, $G_{15} - G_4$ $\{21, 22\}$, and $G_4 - G_0$ $\{11, 12\}$ tasks are the optimal workstation assignments.

The numbers next to the arcs in Figure 4.1 represent the station idle times. Workstation assignments of this problem were given in Figure 2.1(c).

## 5. Concluding remarks

Parallel ALB problem suggested by Gökçen et al. [21] is a new research area of the line balancing literature. In the parallel assembly line balancing, the goal is to balance more than one assembly line together. That is, it will be possible to assign task(s) from each line to a multiskilled operator. As a result, it is inevitable to minimize the total idle times of the lines.In this paper, a network model for parallel assembly line balancing (ALB) problem is proposed and illustrated on a numerical example. The model is based on the Gutjahr and Nemhauser's [14] algorithm developed for single-model ALB problem. The shortest-route model developed here for parallel ALB problem is a new approach and it provides a different perspective for interested ALB researchers. Furthermore, model can also be used as a framework to develop effective heuristic procedures to solve the parallel line balancing problem.

*Notations*

(i) $C$: cycle time,
(ii) $h$: line number, $h = 1, \ldots, H$,
(iii) $k$: station number, $k = 1, \ldots, K_{\max}$,
(iv) $\|M_{hk}\|$: total number of tasks (that can be) assigned to station $k$ in line $h$,
(v) $n_h$: number of tasks in line $h$,
(vi) $t_{hi}$: performance time of task $i$ in line $h$,
(vii) $K_{\max}$: maximum number of station,
(viii) $P_h$: set of precedence relationships in precedence diagram of line $h$,
(ix) $x_{hik}$: 1, if task $i$ in line $h$ is assigned to station $k$; 0, otherwise,
(x) $U_{hk}$: 1, if station $k$ is utilized in line $h$; 0, otherwise,
(xi) $z_k$: 1, if station $k$ is utilized; 0, otherwise.

## References

[1] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *European Journal of Operational Research*, vol. 168, no. 3, pp. 694–715, 2006.

[2] S. Ghosh and R. J. Gagnon, "A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems," *International Journal of Production Research*, vol. 27, no. 4, pp. 637–670, 1989.

[3] M. E. Salveson, "The assembly line balancing problem," *Journal of Industrial Engineering*, vol. 6, no. 3, pp. 18–25, 1955.

[4] İ. Baybars, “A survey of exact algorithms for the simple assembly line balancing problem,” *Management Science*, vol. 32, no. 8, pp. 909–932, 1986.

[5] E. Erel and S. C. Sarin, “A survey of the assembly line balancing procedures,” *Production Planning and Control*, vol. 9, no. 5, pp. 414–434, 1998.

[6] A. Scholl and C. Becker, “State-of-the-art exact and heuristic solution procedures for simple assembly line balancing,” *European Journal of Operational Research*, vol. 168, no. 3, pp. 666–693, 2006.

[7] G. J. Miltenburg and J. Wijngaard, “U-line line balancing problem,” *Management Science*, vol. 40, no. 10, pp. 1378–1388, 1994.

[8] T. L. Urban, “Note: optimal balancing of U-shaped assembly lines,” *Management Science*, vol. 44, no. 5, pp. 738–741, 1998.

[9] A. Scholl and R. Klein, “ULINO: optimally balancing U-shaped JIT assembly lines,” *International Journal of Production Research*, vol. 37, no. 4, pp. 721–736, 1999.

[10] K. Ohno and K. Nakade, “Analysis and optimization of a U-shaped production line,” *Journal of the Operations Research Society of Japan*, vol. 40, no. 1, pp. 90–104, 1997.

[11] J. Miltenburg, “Balancing U-lines in a multiple U-line facility,” *European Journal of Operational Research*, vol. 109, no. 1, pp. 1–23, 1998.

[12] D. Sparling and J. Miltenburg, “The mixed-model U-line balancing problem,” *International Journal of Production Research*, vol. 36, no. 2, pp. 485–501, 1998.

[13] J. Miltenburg, “U-shaped production lines: a review of theory and practice,” *International Journal of Production Economics*, vol. 70, no. 3, pp. 201–214, 2001.

[14] F. Guerriero and J. Miltenburg, “The stochastic U-line balancing problem,” *Naval Research Logistics*, vol. 50, no. 1, pp. 31–57, 2003.

[15] G. A. Süer and C. Dagli, “A knowledge-based system for selection of resource allocation rules and algorithms,” in *Handbook of Expert System Applications in Manufacturing; Structures and Rules*, A. Mital and S. Anand, Eds., pp. 108–147, Chapman and Hall, New York, NY, USA, 1994.

[16] G. A. Süer, “Designing parallel assembly lines,” *Computers and Industrial Engineering*, vol. 35, no. 3-4, pp. 467–470, 1998.

[17] A. S. Simaria and P. M. Vilarinho, “The simple assembly line balancing problem with parallel workstations—a simulated annealing approach,” *International Journal of Industrial Engineering*, vol. 8, no. 3, pp. 230–240, 2001.

[18] R. G. Askin and M. Zhou, “A parallel station heuristic for the mixed-model production line balancing problem,” *International Journal of Production Research*, vol. 35, no. 11, pp. 3095–3105, 1997.

[19] P. R. McMullena and G. V. Frazier, “A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations,” *International Journal of Production Economics*, vol. 51, no. 3, pp. 177–190, 1997.

[20] P. M. Vilarinho and A. S. Simaria, “A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations,” *International Journal of Production Research*, vol. 40, no. 6, pp. 1405–1420, 2002.

[21] H. Gökçen, K. Ağpak, and R. Benzer, “Balancing of parallel assembly lines,” *International Journal of Production Economics*, vol. 103, no. 2, pp. 600–609, 2006.

[22] M. Klein, “On assembly line balancing,” *Operations Research*, vol. 11, no. 2, pp. 274–281, 1963.

[23] A. L. Gutjahr and G. L. Nemhauser, “An algorithm for the line balancing problem,” *Management Science*, vol. 11, no. 2, pp. 308–315, 1964.

[24] E. M. Mansoor, “Improvement on Gutjahr and Nemhauser’s algorithm for the line balancing problem,” *Management Science*, vol. 14, no. 3, pp. 250–254, 1967.

[25] S. D. Roberts and C. D. Villa, “On a multiproduct assembly line balancing problem,” *AIIE Transactions*, vol. 2, no. 4, pp. 361–364, 1970.

[26] A. K. Chakravarty and A. Shtub, "Balancing mixed model lines with in-process inventories," *Management Science*, vol. 31, no. 9, pp. 1161–1174, 1985.

[27] E. Erel and H. Gökçen, "Shortest-route formulation of mixed-model assembly line balancing problem," *European Journal of Operational Research*, vol. 116, no. 1, pp. 194–204, 1999.

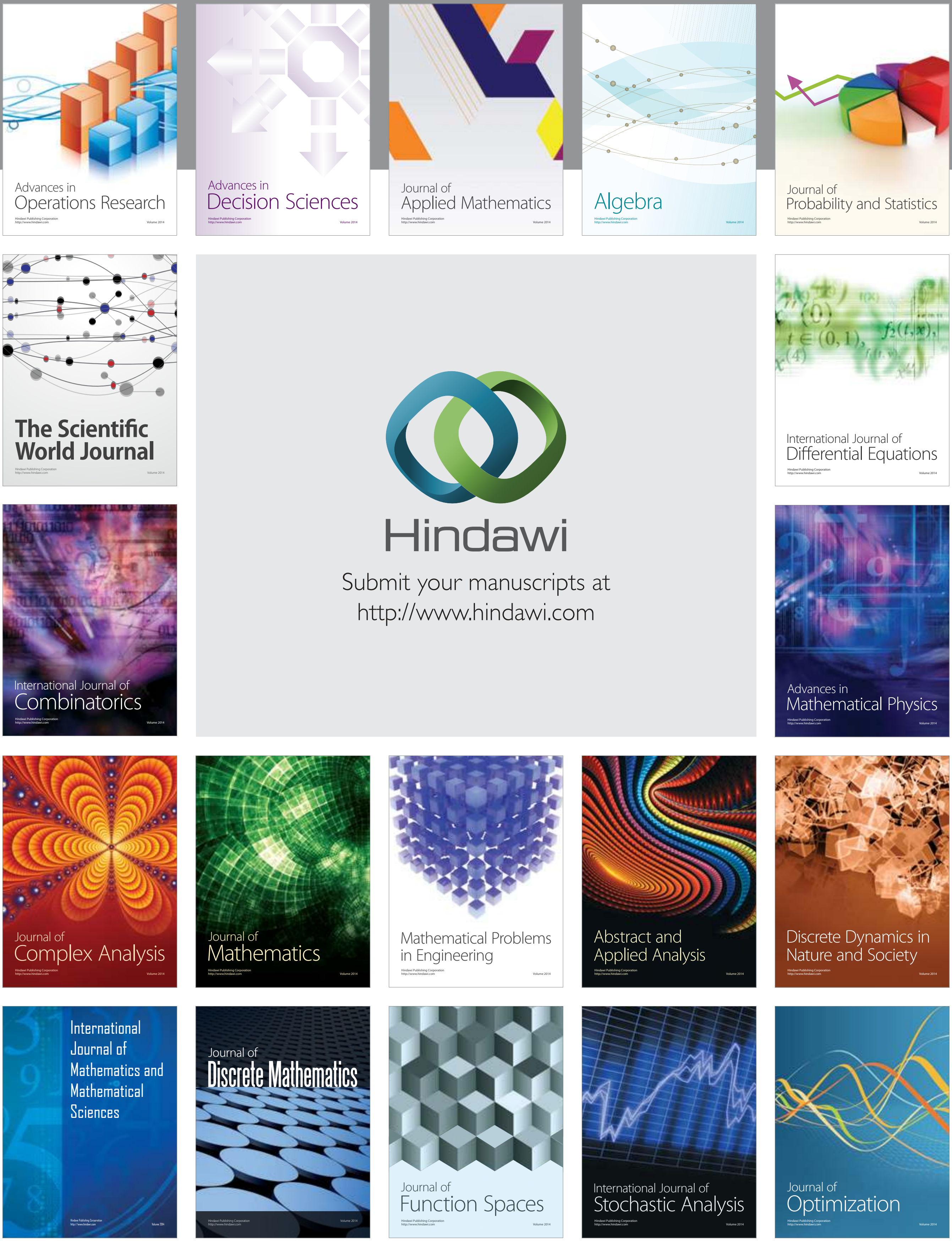
[28] H. Gökçen, K. Ağpak, C. Gencer, and E. Kizilkaya, "A shortest route formulation of simple U-type assembly line balancing problem," *Applied Mathematical Modelling*, vol. 29, no. 4, pp. 373–380, 2005.

Recep Benzer: Turkish General Staff, Communication and Information System, Support Command,
Balgat, 06100 Ankara, Turkey
*Email address*: rbenzer@tsk.mil.tr

Hadi Gökçen: Department of Industrial Engineering, Faculty of Engineering and Architecture,
Gazi University, Maltepe, 06570 Ankara, Turkey
*Email address*: hgokcen@gazi.edu.tr

Tahsin Çetinyokuş: Department of Industrial Engineering, Faculty of Engineering and Architecture,
Gazi University, Maltepe, 06570 Ankara, Turkey
*Email address*: tahsinc@gazi.edu.tr

Hakan Çerçioglu: Department of Industrial Engineering, Faculty of Engineering and Architecture,
Gazi University, Maltepe, 06570 Ankara, Turkey
*Email address*: cercioglu@gazi.edu.tr


Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization