



ELSEVIER

European Journal of Operational Research 116 (1999) 194–204

EUROPEAN  
JOURNAL  
OF OPERATIONAL  
RESEARCH

Theory and Methodology

## Shortest-route formulation of mixed-model assembly line balancing problem

Erdal Erel <sup>a,\*</sup>, Hadi Gokcen <sup>b</sup>

<sup>a</sup> *Faculty of Business Administration, Bilkent University, 06533 Bilkent, Ankara, Turkey*

<sup>b</sup> *Department of Industrial Engineering, Gazi University, 06570 Maltepe, Ankara, Turkey*

Received 3 October 1997; accepted 3 February 1998

---

### Abstract

A shortest-route formulation of the mixed-model assembly line balancing problem is presented. Common tasks across models are assumed to exist and these tasks are performed in the same stations. The formulation is based on an algorithm which solves the single-model version of the problem. The mixed-model system is transformed into a single-model system with a combined precedence diagram. The model is capable of considering any constraint that can be expressed as a function of task assignments. © 1999 Elsevier Science B.V. All rights reserved.

**Keywords:** Assignment; Assembly line balancing; Network programming

---

### 1. Introduction

The assembly line balancing problem has been a focus of interest to the production/operations management community for the last four decades. Although there are numerous studies published on the various aspects of the problem, the number of studies on mixed-model assembly lines are relatively small. The complex mathematical nature of the problem hinders the attempts to obtain solution procedures.

Single-model lines are used to assemble large numbers of a product whereas multi-model lines

are used to assemble different models of the same general product in batches with large lot sizes. Mixed-model lines are used to assemble different models of a product; the models are launched to the line one after another. This paper deals with the mixed-model assembly lines which are the most frequently encountered type in industry due to the pressure of producing several models to attain higher customer satisfaction. Formally, a mixed-model assembly line balancing problem can be stated as follows: Given  $P$  models, the set of tasks associated with each model, the performance times of the tasks, and the set of precedence relations which specify the permissible orderings of the tasks for each model, the problem is to assign the tasks to an ordered sequence of stations such that the precedence relations are satisfied and some

---

\* Corresponding author. Tel.: +90 312 266 4164; fax: +90 312 266 4958; e-mail: erel@bilkent.edu.tr.

performance measure is optimized. The complex mathematical nature of the problem makes it difficult to solve; for example, with a single model and tasks with no precedence relations, it is easy to reduce the problem into a bin-packing problem which is NP-hard in the strong sense. In this note we present a shortest-route formulation of the mixed-model assembly line balancing problem that leads to the optimal solution. It is assumed that a set of common tasks across the models exists. Although the performance times of these common tasks may vary across models, they are performed in the same stations. The performance measure is the sum of the idle times associated with each model. The framework allows various constraints being considered simultaneously.

The first researcher who constructed a mathematical model of the assembly line balancing problem and suggested a solution procedure was, to the best knowledge of the authors, Salvendy (1955). Later, numerous optimum-seeking algorithms and heuristic procedures that attempt to solve different versions of the problem have been developed; interested reader should see the review papers of Baybars (1986) and Ghosh and Gagnon (1989). The shortest-route formulations of the problem that have appeared in the literature are as follows. Klein (1963) presented the first shortest-route formulation of the single-model assembly line balancing problem. The network had directed arcs representing possible assignments of tasks to stations, and each path from source to sink represented a possible line design. Gutjahr and Nemhauser (1964) developed an algorithm to solve the single-model version of the problem based on finding the shortest-route in a directed network. The model was considerably superior to the model of Klein (1963) in the sense that only a portion of the feasible orderings was generated. Mansoor (1967) presented an adjustment to the Gutjahr and Nemhauser (1964) algorithm to obtain the optimal solution after considering only a fraction of the 'shortest-route' calculations. Roberts and Villa (1970) extended the Gutjahr and Nemhauser (1964) algorithm to solve the mixed-model version of the problem. Nodes were generated similarly for each model in the problem, and the nodes of the network consisted of the Cartesian product of the

nodes of each model. Thus, the size of the tree grows very rapidly with increases in the number of tasks and models. Finally, Chakravarty and Shtub (1985) presented a shortest-route formulation for the mixed-model, unpaced line in which setup, inventory holding and labor costs were considered. Precedence relations of the different models were reflected in a combined precedence diagram which was transformed into a serial system based on the concepts of multi-echelon inventory and holding costs. The serial system was then solved with a shortest-route algorithm.

The model proposed in this note is heavily based on the algorithm of Gutjahr and Nemhauser (1964). The mixed-model version of the problem is transformed into a single-model version with a combined precedence diagram. Common tasks between the models are assumed to exist and these tasks are assigned to the same stations. The resulting network is considerably smaller than the one of the shortest-route formulation of Roberts and Villa (1970); this is partly due to the constraint of assigning the common tasks to the same stations.

The paper is organized as follows: In Section 2, we present the shortest-route formulation of the mixed-model assembly line balancing problem. The model is further clarified on an illustrative example in Section 3. Some concluding remarks are given in Section 4.

## 2. Shortest-route formulation

### 2.1. Assumptions

The assumptions of the model are listed below:

1. Task performance times of each model are known constants.
2. Precedence diagrams for each model are known.
3. No WIP inventory buffer is allowed between stations.
4. Parallel stations are not allowed.
5. Common tasks exist between models which must be assigned to the same stations.

Let  $t_{im}$  denote the performance time of task  $i$  of model  $m$ ,  $i = 1, \dots, N$ ,  $m = 1, \dots, P$ , where  $N$  and

$P$  are the total number of tasks and models in the problem, respectively. The last assumption above is especially valid if the common tasks of different models require the same tools and equipment. However, the performance times of these common tasks can differ for different models. In other words,  $t_{im}$  may not equal to  $t_{in}$  for  $i = 1, \dots, N$  and  $m, n = 1, \dots, P$ ,  $m \neq n$ .

Typically there are several tasks common to the various models manufactured on a mixed-

model assembly line with similar precedence relations among these tasks. Thus, we will utilize the similarity between the precedence relations of different models. Thomopoulos (1970) used the concept of a combined precedence diagram to transform different models into an equivalent single model. The combined diagram is constructed by taking the union of the nodes and the precedence relations of the diagrams of all the models. The construction of the combined dia-

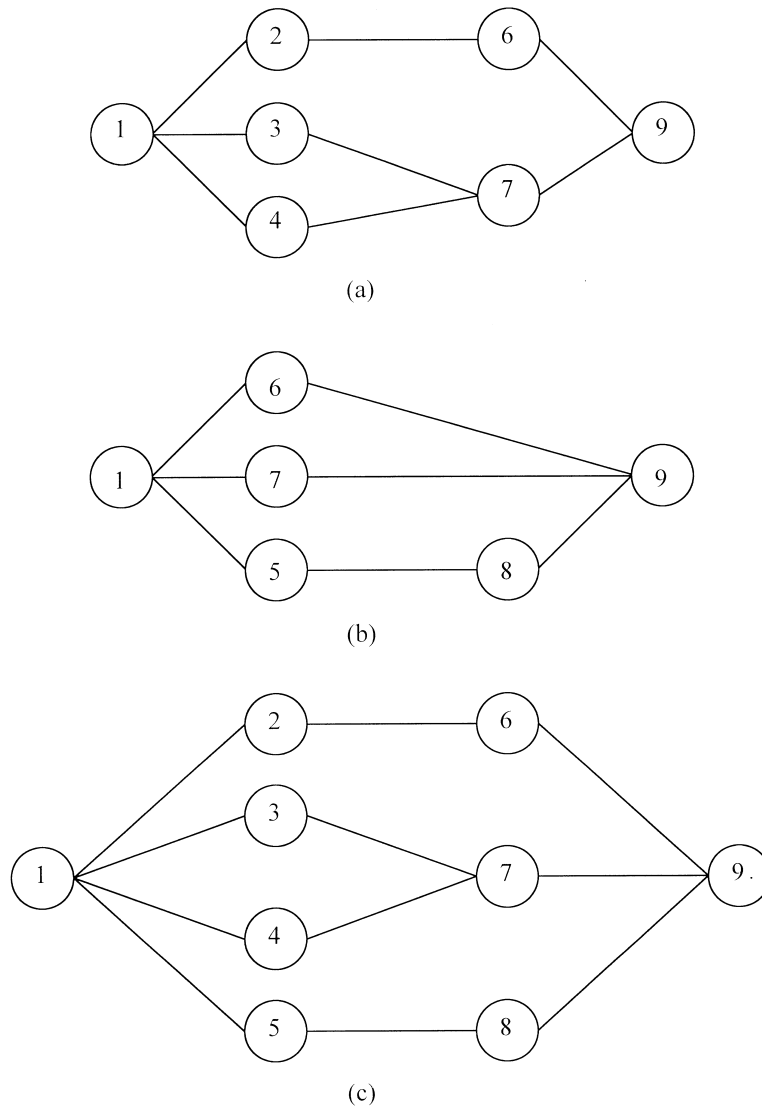


Fig. 1. Precedence diagrams for: (a) model A, (b) model B, (c) the combined diagram.

gram is straightforward with precedence matrices; a precedence matrix is an upper-triangular matrix with the  $ab$ th entry equal to 1 if the processing of task  $b$  requires the completion of task  $a$ . Otherwise the entry is zero. In the precedence matrix of the combined diagram, the  $ab$ th entry is equal to 1 if the  $ab$ th entry of any of the precedence matrices of the models is equal to 1. Furthermore, if there are any implied precedence relations, then the related entries in the combined precedence

matrix should also be 1. Note that there should be no conflict in the precedence relations across the models; for example, if a model requires the completion of task  $a$  before task  $b$ , then no other model should require the completion of task  $b$  before task  $a$ . A simple example is given in Fig. 1 to illustrate the process of constructing a combined diagram. The numbers within the circles represent tasks and the arrows connecting the circles specify the precedence relations. In Fig. 2,

	1	2	3	4	6	7	9
1		1	1	1	1	1	1
2			0	0	1	0	1
3				0	0	1	1
4					0	1	1
6						0	1
7							1
9							

(a)

	1	5	6	7	8	9
1		1	1	1	1	1
5			0	0	1	1
6				0	0	1
7					0	1
8						1
9						

(b)

	1	2	3	4	5	6	7	8	9
1		1	1	1	1	1	1	1	1
2			0	0	0	1	0	0	1
3				0	0	0	1	0	1
4					0	0	1	0	1
5						0	0	1	1
6							0	0	1
7								0	1
8									1
9									

(c)

Fig. 2. Precedence matrices of: (a) model A, (b) model B, (c) the combined diagram.

the associated precedence matrices of the diagrams of Fig. 1 are depicted.

## 2.2. Constraints

The constraints of the model are given below:

1. All tasks must be assigned to a station.
2. No task can be assigned more than once.
3. The work content in any station for any model cannot exceed the cycle time of that model.
4. If task  $a$  in a given model precedes task  $b$ , then task  $b$  cannot be assigned to a station that precedes the one to which task  $a$  is assigned.

Let  $M$  denote the number of stations utilized and  $ST_k$  and  $S$  denote the set of tasks in station  $k$  and in the problem, respectively. Let also  $C_m$  denote the cycle time of model  $m$  and  $x_{imk} = 1$  if task  $i$  of model  $m$  is assigned to station  $k$ , 0 otherwise. The problem can now be stated as follows:

$$\min \sum_{m=1}^P \sum_{k=1}^M \left\{ C_m - \sum_{i=1}^N x_{imk} t_{im} \right\}$$

s.t.

$$\bigcup_{i=1}^M ST_i = S,$$

$$ST_i \cap ST_j = \emptyset \quad \text{for } i, j = 1, \dots, M \text{ and } i \neq j,$$

$$\sum_{i=1}^N x_{imk} \cdot t_{im} \leq C_m \quad \text{for } k = 1, \dots, M \text{ and}$$

$$m = 1, \dots, P.$$

For each task  $b \in S$ , if  $ab$ th entry is 1 in the combined precedence matrix,  $a \in ST_i$ ,  $b \in ST_j$ , then  $i \leq j$  for all  $a \in S$ .

Note that the objective function above can also be expressed as  $\sum_{m=1}^P \{MC_m - \sum_{i \in B_m} t_{im}\}$  where  $B_m$  is the set of tasks of model  $m$ . Minimizing the sum of idle times can now be expressed as the minimization of  $M$ , since  $\sum_{i \in B_m} t_{im}$  and  $C_m$  are constants for  $m = 1, \dots, P$ .

In addition to the traditional constraints above, non-traditional constraints such as an upper bound on the idle time of each station or upper bounds on the work load differences between models can easily be imposed. In fact, any constraint which can be expressed as a function of the task assignment can be considered.

## 2.3. Model

The network model developed here is based on the shortest-route model of Gutjahr and Nemhauser (1964) for the single-model assembly line balancing problem. The nodes of the network are constructed similar to the Gutjahr and Nemhauser's procedure; nodes are the states that are sets of tasks that can be processed without prior completion of any other tasks on the combined precedence diagram. Let  $A_i, i = 1, \dots, r$  denote the set of tasks in node  $i$  where  $r$  is the total number of nodes with  $A_0 = \emptyset$  and  $A_r$  representing the set of all the tasks in the combined diagram. That is,  $A_r = S$ . Also let  $\tau_{A_i, m} = \sum_{j \in A_i} t_{j, m}$  for  $m = 1, \dots, P$ . A directed arc  $(ij)$  connects node  $i$  to node  $j$  if  $A_i \subset A_j$ ,  $\tau_{A_j, m} - \tau_{A_i, m} \leq C_m$  for  $m = 1, \dots, P$ , and any other constraint expressed as a function of the task assignment being satisfied. Each directed arc  $(ij)$  from node  $i$  to node  $j$  is assigned a distance of  $\sum_{m=1}^P C_m - [\tau_{A_j, m} - \tau_{A_i, m}]$ . Note that no arc enters node 0, the node corresponding to  $A_0$ , and no arc leaves node  $r$ , the node corresponding to  $A_r$ .

The length of a path from node 0 to node  $r$  with  $M$  arcs,  $(0i, ij, \dots, pr)$ , is  $\sum_{m=1}^P \{C_m - [\tau_{A_i, m}] + C_m - [\tau_{A_j, m} - \tau_{A_i, m}] + \dots + C_m - [\tau_{A_r, m} - \tau_{A_p, m}]\}$ . This length can be expressed as  $\sum_{m=1}^P \{MC_m - \tau_{A_r, m}\}$ . Note also that  $\tau_{A_r, m} = \sum_{i \in B_m} t_{im}$  for  $m = 1, \dots, P$ . Hence, this length is the sum of the idle times over the  $M$  stations for all the models. Since  $\tau_{A_r, m}$  and  $C_m$  are constants for  $m = 1, \dots, P$ , finding the shortest-path from node 0 to node  $r$  can be achieved by finding any path from node 0 to node  $r$  with the least number of arcs. In other words, a path with the least number of arcs (from node 0 to node  $r$ ) yields the optimal solution of the problem. If the  $k$ th arc on this path connects node  $i$  to node  $j$ , then the set of tasks in station  $k$  associated with model  $m$  is  $B_m \cap (A_j - A_i)$  with a work load of  $\tau_{A_j, m} - \tau_{A_i, m}$  for  $m = 1, \dots, P$ . This proves that the mixed-model line balancing problem can be solved with this shortest-path formulation.

The path with the least number of arcs can be constructed similar to the procedure given in Gutjahr and Nemhauser (1964) as follows:

1. Determine the nodes that are connected to node 0 with an arc. There is an arc from node 0 to node  $i$  if  $\tau_{A_i, m} \leq C_m$  for  $m = 1, \dots, P$ . These

nodes are called the first nodes.

2. For node  $i$  among the first nodes, construct an arc to node  $j$  if  $A_i \subset A_j$ ,  $\tau_{A_j,m} - \tau_{A_i,m} \leq C_m$  for  $m = 1, \dots, P$ . These nodes are called the second nodes.
3. Repeat the above process until node  $r$  is obtained for the first time.

Additional constraints can easily be considered in the above process; for example, if an upper bound on the idle time of model  $m$  in any station,  $I_m$ , is imposed, then the construction of an arc from node  $i$  to node  $j$  requires the additional constraint  $C_m - [\tau_{A_j,m} - \tau_{A_i,m}] \leq I_m$  being satisfied. If a maximum work load difference of  $D$  between models is imposed, then construction of an arc from node  $i$  to node  $j$  requires the additional constraint

$$|[\tau_{A_j,m} - \tau_{A_i,m}] - [\tau_{A_j,n} - \tau_{A_i,n}]| \leq D$$

for  $m, n = 1, \dots, P, m \neq n$ .

### 2.3.1. Node generation

The node generation process developed by Gutjahr and Nemhauser (1964) for the single-model assembly line balancing problem will be used in this paper. Note that there are other procedures to generate the nodes of the network reported in the literature; for example, Schrage and Baker (1978) present an enumeration and addressing scheme to generate the feasible subsets.

The node generation process can be described as follows: the tasks that are available for assignment (i.e., the tasks without any predecessor in the combined precedence diagram) are placed in Stage 1 and are considered marked. An immediate follower of a state  $S$  of a stage is defined as a task that is an immediate follower of at least one of the tasks in  $S$  and is not preceded by any tasks not in  $S$ . The unmarked immediate followers of a state are augmented to the current state to form the states of the next stage. The augmentation of states and the corresponding unmarked immediate followers is done in stages. For any state  $S$  of stage  $k$ , the unmarked immediate followers are placed in a list called  $F(S)$ . Let  $H \subset F(S)$ , the  $S \cup H$  is a state for stage  $k+1$ . For each state of stage  $k$ , the unmarked immediate followers are found and placed

as marked tasks for stage  $k+1$ . When all the tasks are marked or  $F(S)$  gets empty for the current stage, the node generation process is complete.

The node generation process above constructs all possible feasible states. The number of feasible states ranges from  $N$  corresponding to a serially ordered precedence diagram to  $2^N - 1$  corresponding to a problem with no precedence relations. However, precedence relations reduce the total number of feasible states considerably. Sidney and Steiner (1986) state that the number of feasible states is substantially reduced with the precedence relations. They report that the total number of feasible states is bounded from above by a polynomial in  $N$  for fixed-width networks. They show that the bound on the number of feasible states is  $(1 + k_1)(1 + k_2) \dots (1 + k_w)$  where the combined precedence diagram is partitioned into  $w$  chains having  $k_1, k_2, \dots, k_w$  nodes, respectively. Here,  $w$  is the minimum number of chains needed to partition the nodes of the precedence diagram.

## 3. Illustrative example

Suppose that two models A and B whose precedence diagrams are depicted in Fig. 1 are assembled on the same line. The combined diagram is also depicted in Fig. 1. Task performance times for each model are given in Table 1. The cycle times of models A and B are 6 and 5, respectively.

The generation of the nodes is shown in Table 2. Initially, task 1 is placed in stage 1 and

Table 1  
Task performance times of models A and B

Tasks	Performance times	
	Model A	Model B
1	2	2
2	3	—
3	2	—
4	1	—
5	—	1
6	2	3
7	3	4
8	—	3
9	2	1

Table 2  
Generation of states of nodes

Stage	Marked tasks	State number	State elements	State times for		Unmarked immediate followers
				Model A	Model B	
0		0	$\emptyset$	0	0	1
1	1	1	1	2	2	2,3,4,5
2	2,3,4,5	2	1,2	5	2	6
		3	1,3	4	2	—
		4	1,4	3	2	—
		5	1,5	2	3	8
		6	1,2,3	7	2	6
		7	1,2,4	6	2	6
		8	1,2,5	5	3	6,8
		9	1,3,4	5	2	7
		10	1,3,5	4	3	8
		11	1,4,5	3	3	8
		12	1,2,3,4	8	2	6,7
		13	1,2,3,5	7	3	6,8
		14	1,2,4,5	6	3	6,8
		15	1,3,4,5	5	3	7,8
		16	1,2,3,4,5	8	3	6,7,8
3	6,7,8	17	1,2,6	7	5	—
		18	1,5,8	2	6	—
		19	1,2,3,6	9	5	—
		20	1,2,4,6	8	5	—
		21	1,2,5,6	7	6	—
		22	1,2,5,8	5	6	—
		23	1,2,5,6,8	7	9	—
		24	1,3,4,7	8	6	—
		25	1,3,5,8	4	6	—
		26	1,4,5,8	3	6	—
		27	1,2,3,4,6	10	5	—
		28	1,2,3,4,7	11	6	—
		29	1,2,3,4,6,7	13	9	—
		30	1,2,3,5,6	9	6	—
		31	1,2,3,5,8	7	6	—
		32	1,2,3,5,6,8	9	9	—
		33	1,2,4,5,6	8	6	—
		34	1,2,4,5,8	6	6	—
		35	1,2,4,5,6,8	8	9	—
		36	1,3,4,5,7	8	7	—
		37	1,3,4,5,8	5	6	—
		38	1,3,4,5,7,8	8	10	—
		39	1,2,3,4,5,6	10	6	—
		40	1,2,3,4,5,7	11	7	—
		41	1,2,3,4,5,8	8	6	—
		42	1,2,3,4,5,6,7	13	10	—
		43	1,2,3,4,5,6,8	10	9	—
		44	1,2,3,4,5,7,8	11	10	—
		45	1,2,3,4,5,6,7,8	13	13	9
4	9	46	1,2,3,4,5,6,7,8,9	15	14	—

considered marked. The unmarked immediate followers of task 1 are tasks 2–5; they are placed in list  $F(S)$  corresponding to task 1 and are also

placed in stage 2 as marked tasks. Task 1 is then augmented to all subsets of the list  $F(S)$  that include tasks 2, 3, 4, and 5 to form the states of stage

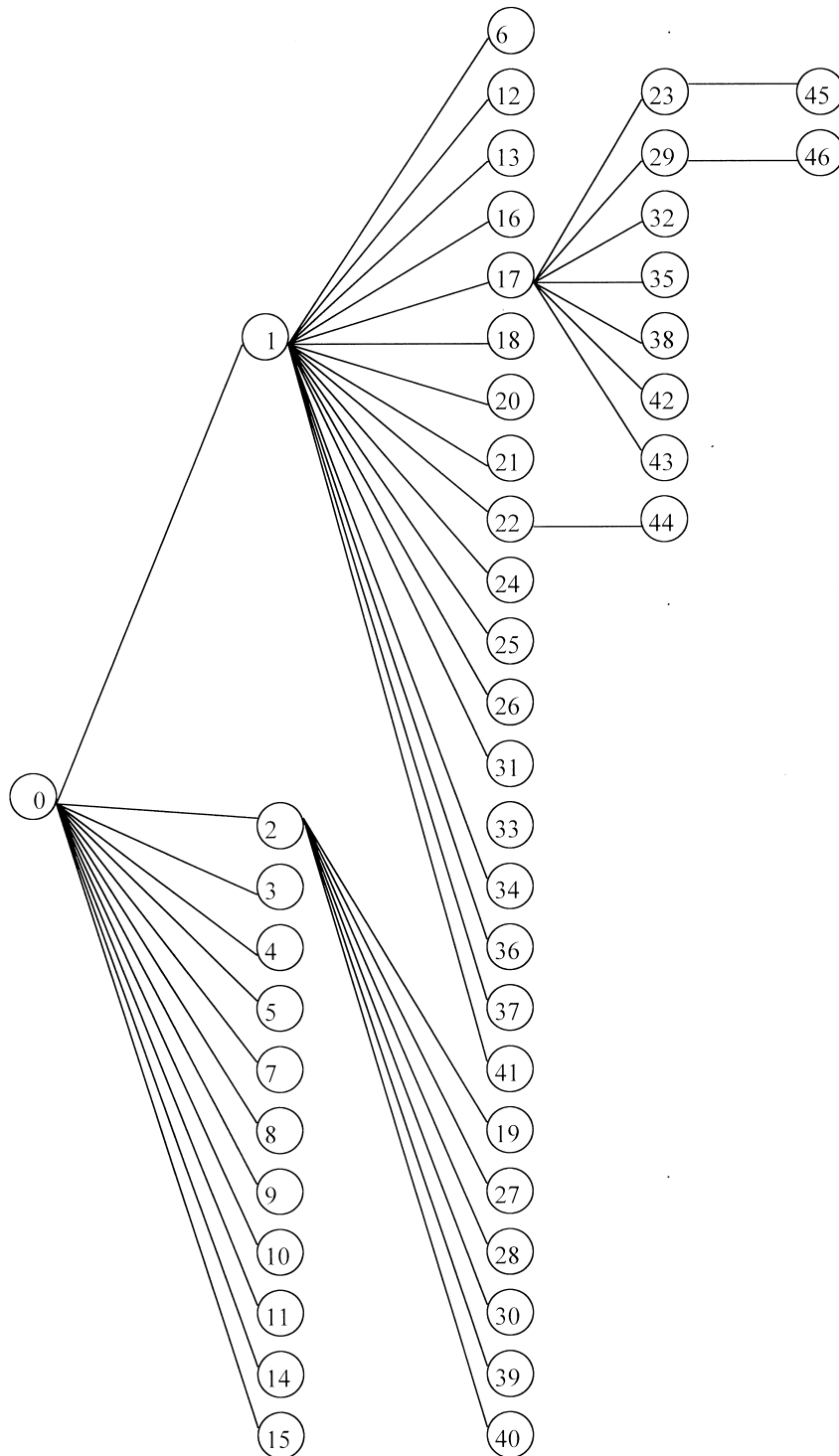


Fig. 3. Network of the example problem constructed only with the cycle time constraints.

2. For each state in stage 2, the corresponding  $F(S)$  list is found; this list is empty for some states. The procedure terminates in stage 4 since  $F(S)$  is empty for all the states of that stage. The total number of states generated is 46 which is considerably less than  $2^9 - 1 = 511$  that corresponds to the number of feasible states with no precedence relations.

Fig. 3 depicts the network that is constructed with the constraint of  $\tau_{A_j,m} - \tau_{A_i,m} \leq C_m$  for  $m = 1, 2$ . Nodes 1–5, 7–11, 14 and 15 comprise the first nodes, since  $\tau_{A_i,m} \leq C_m$  for  $m = 1, 2$  for these nodes. Note that node 6 is not among the first nodes since  $\tau_{A_6,1} = 7 > C_1$ , although  $\tau_{A_6,2} = 2 < C_2$ . From node 1, 18 new nodes can be constructed with  $A_1 \subset A_j$  and  $\tau_{A_j,m} - \tau_{A_1,m} \leq C_m$  for  $m = 1, 2$ . Similarly, from node 2, 6 new nodes can be constructed; these 24 new nodes comprise the second nodes. Node 46 that represents all the tasks in the problem is first obtained with 4 arcs; thus, the optimal solution has 4 stations. Table 3 shows the tasks in these 4 stations associated with each model.

We have solved the same problem using the shortest-route formulation of Roberts and Villa (1970). In the formulation, feasible states are generated for each model and the set of states for the problem is obtained from the Cartesian product of these feasible states. For model 1 and 2, we obtain 16 and 13 feasible states, respectively, as shown below.

*For model 1:* (1), (1,2), (1,3), (1,4), (1,2,3), (1,2,4), (1,2,6), (1,3,4), (1,2,3,4), (1,2,3,6), (1,2,4,6), (1,3,4,7), (1,2,3,4,6), (1,2,3,4,7), (1,2,3,4,6,7), (1,2,3,4,6,7,9)

*For model 2:* (1), (1,5), (1,6), (1,7), (1,5,6), (1,5,7), (1,5,8), (1,6,7), (1,5,6,7), (1,5,6,8), (1,5,7,8), (1,5,6,7,8), (1,5,6,7,8,9)

The Cartesian product of these states results in  $16 \times 13 = 208$  states as follows: [(1), (1)], [(1), (1,5)], ..., [(1,2,3,4,6,7,9), (1,5,6,7,8,9)]. In the resulting network, node 208 that represents all the

tasks in the problem is first reached with 3 arcs; thus, the optimal solution has 3 stations. Table 4 shows the tasks in these 3 stations associated with each model. Note that task 6 which is a common task among the models is assigned to different stations; the assumption of assigning the common tasks to the same stations of our model results in a solution with one extra station. However, the number of nodes in the network of Roberts and Villa (1970) is considerably higher (more than 4.5 times). As the number of models in the problem increases, the difference between the sizes of the networks will be significantly larger, since the Cartesian product will increase the network size quite rapidly. However, the increase in the size of the network of our model will be relatively slower as discussed in Section 2.3.1. Note also that assigning the common tasks to the same stations can also be a practical requirement for tasks that require special tools/fixtures; Sparling and Miltenburg (1998) also make this assumption in their study on U-line balancing problem.

Note that in the optimal solution of our model the work load difference in station 4 between the models is 3. If a maximum work load difference of 2 is imposed, then the arc from node  $i$  to node  $j$  requires the additional constraint  $|\tau_{A_j,1} - \tau_{A_i,1} - [\tau_{A_j,2} - \tau_{A_i,2}]| \leq 2$ . The corresponding network is depicted in Fig. 4 in which the optimal solution still requires 4 stations. Note that nodes 2, 6, 7, 12 and 18 are missing, since the arcs constructed on these nodes violate the maximum work load difference constraint. Thus, the network in Fig. 4 consists of 41 nodes. Table 5 shows the tasks in the 4 stations.

#### 4. Concluding remarks

shortest-route model for the mixed-model assembly line balancing problem is developed.

Table 3  
Optimal task assignment of the network in Fig. 3

Station	Tasks	Model A		Model B	
		Tasks	Work load	Tasks	Work load
1	1	1	2	1	2
2	2,6	2,6	5	6	3
3	3,4,7	3,4,7	6	7	4
4	5,8,9	9	2	5,8,9	5

Table 4  
Optimal task assignment for Roberts and Villa model

Station	Model A		Model B	
	Tasks	Work load	Tasks	Work load
1	1,2,4	6	1,6	5
2	3,7	5	5,7	5
3	6,9	4	8,9	4

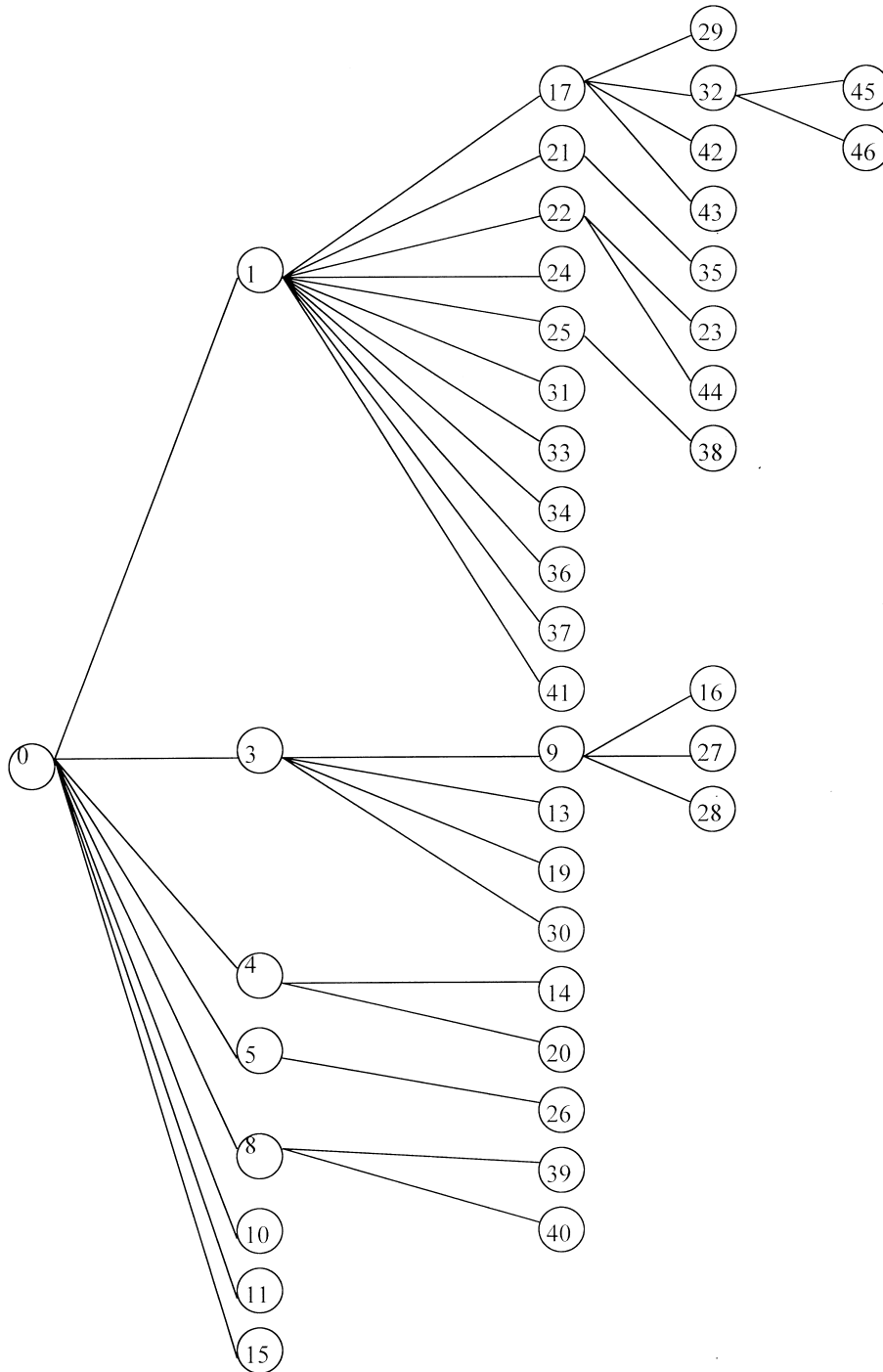


Fig. 4. Network of the example problem constructed with the cycle time constraints and a maximum work load difference of 2 time units.

Table 5  
Optimal task assignment of the network in Fig. 4

Station	Tasks	Model A		Model B	
		Tasks	Work load	Tasks	Work load
1	1	1	2	1	2
2	2,6	2,6	5	6	3
3	3,5,8	3	2	5,8	4
4	4,7,9	4,7,9	6	7,9	5

The model is based on the shortest-route model developed by Gutjahr and Nemhauser (1964) for the single-model assembly line balancing problem. The model is capable of considering any constraint that can be expressed as a function of the tasks assignments. The size of the network grows as  $N$  increases; however, assigning common tasks of different models to the same stations and the constraints that can be imposed by the designers limit the increase. Thus, it is more efficient than the shortest-route formulation of Roberts and Villa (1970) in which the nodes of the network consists of the Cartesian product of the nodes of each model. The model proposed in this note is a new shortest-route model of the mixed-model version in which a combined precedence diagram is utilized. It can be used as a framework to develop heuristic procedures to solve the mixed-model assembly line balancing problem.

## References

- Baybars, I., 1986. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science* 32, 909–932.
- Chakravarty, A.K., Shtub, A., 1985. Balancing mixed model lines with in-process inventories. *Management Science* 31, 1161–1174.
- Ghosh, S., Gagnon, J., 1989. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research* 27, 637–670.
- Gutjahr, A.L., Nemhauser, G.L., 1964. An algorithm for the line balancing problem. *Management Science* 11, 308–315.
- Klein, M., 1963. On assembly line balancing. *Operations Research* 11, 274–281.
- Mansoor, E.M., 1967. Improvement on the Gutjahr and Nemhauser algorithm for line balancing problems. *Management Science* 14, 250–254.
- Roberts, S.D., Villa, C.D., 1970. On a multiproduct assembly line balancing problem. *AIIE Transactions* 2, 361–364.
- Salveson, M.E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering* 6, 18–25.
- Schrage, L., Baker, K.R., 1978. Dynamic programming solution of sequencing problems with precedence constraints. *Operations Research* 26, 444–449.
- Sidney, J.B., Steiner, G., 1986. Optimal sequencing by modular decomposition: Polynomial algorithms. *Operations Research* 34, 606–612.
- Sparling, D., Miltenburg, J., 1998. The mixed-model U-line balancing problem. *International Journal of Production Research* 36, 485–502.
- Thomopoulos, N.T., 1970. Mixed model line balancing with smoothed station assignments. *Management Science* 16, 593–603.