



Pergamon

Computers ind. Engng Vol. 34, No. 2, pp. 451–461, 1998

© 1998 Elsevier Science Ltd. All rights reserved

Printed in Great Britain

0360-8352/98 \$19.00 + 0.00

PII: S0360-8352(97)00142-3

BINARY INTEGER FORMULATION FOR MIXED-MODEL ASSEMBLY LINE BALANCING PROBLEM

HADI GÖKÇEN¹ and ERDAL EREL²

¹Department of Industrial Engineering, Faculty of Engineering and Architecture, Gazi University, Maltepe 06570, Ankara, Turkey and ²Faculty of Business Administration, Bilkent University, Bilkent 06533, Ankara, Turkey

(Received 1 June 1997)

Abstract—The assembly line balancing problem has been a focus of interest to the academicians of production/operations management for the last 40 years. 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. In this paper, a binary integer programming model for the mixed-model assembly line balancing problem is developed and some computational properties of the model are given. © 1998 Elsevier Science Ltd. All rights reserved.

INTRODUCTION

The assembly line balancing problem has been a focus of interest to the academicians of production/operations management for the last 40 years. Although there are numerous studies published on the various aspects of the problem, the number of studies on mixed-model assembly lines is relatively small. The problem is NP-hard, since with a single model and tasks with no precedence relations, it is easy to reduce the problem to a bin packing problem which is NP-hard in the strong sense. Hence, the combinatorial nature of the mixed-model line balancing problem makes it difficult to obtain optimal solutions, though the mixed-model line is the most frequently encountered type in industry due to the pressure of producing several models to attain higher customer satisfaction.

A mixed-model assembly line balancing problem can be stated as follows: Given P models, the set of tasks and a cycle time 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 performance measure is optimized.

The first researcher who constructed a mathematical model of the single-model assembly line balancing problem and suggested a solution procedure was, to the best knowledge of the authors, Salvesson [1]. During the 1960's and 70's, numerous papers concerning the problem have been published: the majority suggested heuristic procedures to solve the single-model version of the problem. Interested readers should see the review papers by Baybars [2] and Ghosh and Gagnon [3]. Relatively fewer researchers attempted to solve the single-model version with optimum-seeking algorithms: Bowman [4], White [5], Thangavelu and Shetty [6], Patterson and Albracht [7], and Talbot and Patterson [8] constructed integer programming models, Gutjahr and Nemhauser [9] formulated the problem as a shortest-route network, and Jackson [10] developed a dynamic programming (DP) formulation for the problem. The computation and storage requirements of all these optimum-seeking algorithms were excessive even for problems of modest sizes. It is noteworthy, however, that the formulation of Patterson and Albracht [7] utilized properties that prevent the rapid increase of variables. The integer programming model and the solution procedure of Talbot and Patterson [8] is also reported to obtain optimal solutions in a reasonable amount of time for problems with up to 50 tasks. On the other hand, the number of studies conducted on the mixed-model version of the problem is considerably less. Roberts and Villa [11] were one of the few researchers attempting to solve the mixed-model assembly line balancing problem with an optimum-seeking procedure. They constructed a binary integer pro-

gramming model of the problem; however, the excessive number of variables and constraints prohibited the applicability of the model to problems of even small sizes. They also extended the shortest-route formulation of Gutjahr and Nemhauser [9] to handle a mixed-model version of the problem. Similar to the integer programming model, the number of nodes in the network also grows at a fast rate as the problem size increases. Recently, Berger *et al.* [12] presented a branch-and-bound algorithm with a truncated search for a special case of the mixed-model version of the problem; the models start diverging after all the common tasks are performed. In other words, production processes of the models start diverging after the common tasks are performed. Some researchers addressed the issue of developing sequencing algorithms for the mixed-model lines [13, 14].

In this paper, we develop an integer programming model for the mixed-model version of the problem in which we utilize some properties that prevent the fast increase in the number of variables. Due to the NP-hard nature of the problem, the size of our model would be too large to obtain optimal solutions for problems of realistic sizes. However, the model suggested in this paper presents a significant improvement relative to the models in the literature. It can also be used as a validation tool for heuristic procedures developed for the mixed-model version.

The paper is organized as follows: in Section 2, the binary integer programming model is constructed for the problem. The model is further clarified by an illustrative example in Section 3. In Section 4, some computational properties of the model are discussed and in Section 5, concluding remarks are given.

BINARY INTEGER PROGRAMMING FORMULATION

Notation

The notation used in the formulation is as follows:

N	= total number of tasks in the problem;
K	= number of stations;
P	= number of models (products);
PR_i	= subset of all tasks that precede task i , $i = 1, \dots, N$;
S_j	= subset of all tasks that follow task i , $i = 1, \dots, N$;
t_{im}	= performance time of task i of model m , $i = 1, \dots, N$; $m = 1, \dots, P$;
C_m	= cycle time of model m , $m = 1, \dots, P$;
E_{im}	= earliest station task i of model m can be assigned to, given the precedence relations, $i = 1, \dots, N$; $m = 1, \dots, P$;
L_{im}	= latest station task i of model m can be assigned to, given the precedence relations, $i = 1, \dots, N$; $m = 1, \dots, P$;
V_{ik}	= 1 if task i is assigned to station k ; 0 otherwise;
X_{km}	= 1 if station k is utilized for model m ; 0 otherwise;
A_k	= 1 if station k is utilized by all models; 0 otherwise;
W_{km}	= subset of all tasks that can be assigned to station k of model m ;
$ W_{km} $	= number of tasks in set W_{km} .

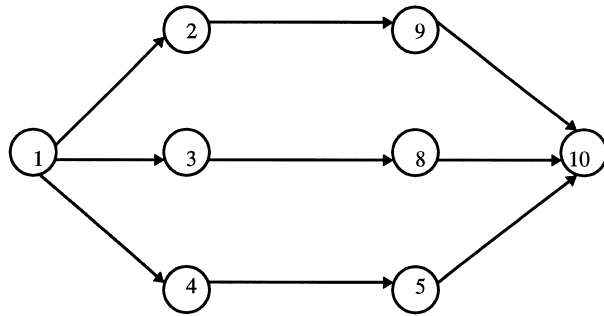
Note that W_{km} is obtained from E_{im} and L_{im} values. Note also that if X_{km} is equal to 1 for station k , for $m = 1, \dots, P$, then A_k equals 1; 0 otherwise.

Assumptions

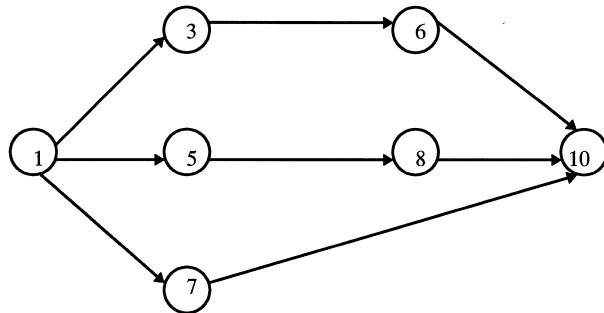
The assumptions of the model are listed below:

1. Task performance times associated with each model are known constants; common tasks among the models do not need to have the same performance times.
2. Precedence relations between the tasks of each model are known.
3. No WIP inventory buffer is allowed between stations.
4. Common tasks of different models must be assigned to the same stations.
5. The number of stations is the same for all models.
6. Parallel stations are not allowed.

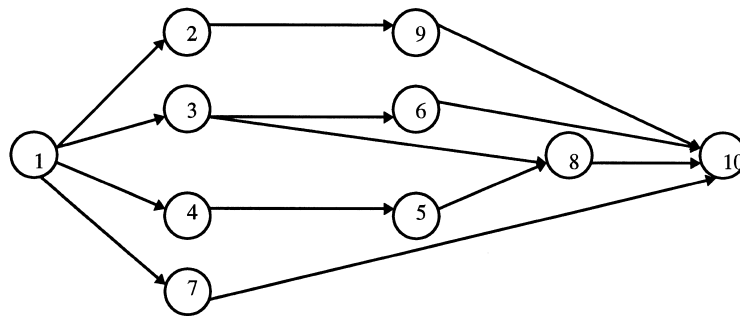
Typically there are several tasks common to the various models manufactured on a mixed-model assembly line with similar precedence relations among these common tasks. Thus, we will utilize the similarity between the precedence relations of different models in our model. Thomopoulos [15] used the concept of a combined precedence diagram to joint the precedence relations of different models on a single diagram. The construction of the combined precedence diagram is straightforward with precedence matrices. A precedence matrix is an upper-triangular



(a)



(b)



(c)

Fig. 1. Precedence diagrams of (a) model 1, (b) model 2 and (c) combined.

matrix with an ab th entry of 1 if the processing of task b requires the completion of task a . Otherwise, the entry is zero. The precedence matrix of the combined precedence diagram is constructed as follows: the ab th entry of the matrix is 1 if the ab th entry of any of precedence matrices of the models is 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 . The combined diagram reduces the number of variables and constraints of the model significantly. Thus, N is typically much smaller than the sum of the number of tasks of the models. A simple example is given in Fig. 1 to illustrate the process of constructing a com-

	1	2	3	4	5	8	9	10
1		1	1	1	1	1	1	1
2			0	0	0	0	1	1
3				0	0	1	0	1
4					1	0	0	1
5						0	0	1
8							0	1
9								1
10								

(a)

	1	3	5	6	7	8	10
1		1	1	1	1	1	1
3			0	1	0	0	1
5				0	0	1	1
6					0	0	1
7						0	1
8							1
10							

(b)

	1	2	3	4	5	6	7	8	9	10
1		1	1	1	1	1	1	1	1	1
2			0	0	0	0	0	0	1	1
3				0	0	1	0	1	0	1
4					1	0	0	1	0	1
5						0	0	1	0	1
6							0	0	0	1
7								0	0	1
8									0	1
9										1
10										

(c)

Fig. 2. Precedence matrices of (a) model 1, (b) model 2, and (c) combined.

bined precedence diagram. The numbers within the nodes represent tasks and the arrows connecting the nodes specify the precedence relations. Figure 2 depicts the precedence matrices of the models in Fig. 1 and the precedence matrix of the combined precedence diagram. Note that the 48th entry in the combined precedence matrix is 1 due to the implied precedence relation between tasks 4 and 8. The interested reader is referred to [15], [16] and [17] for a detailed discussion of combined precedence diagrams.

The earliest and latest stations task i can be assigned to, given the precedence relations, is a problem first developed by Patterson and Albracht [7] for the single-model assembly line balancing problem. The earliest station task i can be assigned to is based on the fact that a sufficient number of stations should be spared for the tasks preceding task i . A lower bound on the earliest station is the ratio of the sum of the performance times of the tasks in PR_i and C . Similarly, the latest station is associated with the tasks following task i on the precedence diagram. Utilizing these concepts greatly reduces the number of variables in the model; in fact, the formulation of Patterson and Albracht [7] required significantly less variables than the earlier

integer programming models published in the literature. We have modified these expressions for the mixed-model assembly line balancing problem as follows:

$$E_{im} = \left\lceil \frac{t_{im} + \sum_{j \in PR_i} t_{jm}}{C_m} \right\rceil \quad \text{for } i = 1, \dots, N, \quad m = 1, \dots, P$$

$$L_{im} = K + 1 - \left\lceil \frac{t_{im} + \sum_{j \in S_i} t_{jm}}{C_m} \right\rceil \quad \text{for } i = 1, \dots, N, \quad m = 1, \dots, P$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x . The earliest and latest stations task i on the combined precedence diagram can be assigned to are $\max_{m=1, \dots, P} \{E_{im}\}$ and $\min_{m=1, \dots, P} \{L_{im}\}$ respectively. The number of stations, K , can be estimated from the operational setting or by utilizing heuristic procedures shown to perform well; note that a loose upper bound on K is N .

Constraints

The constraints of the model can be grouped into four sets and are explained below.

Assignment constraints. This set of constraints assures that tasks of each model are assigned to at most one station and can be written as follows:

$$\sum_{k=E_i}^{L_i} V_{ik} = 1 \quad \text{for } i = 1, \dots, N$$

Precedence constraints. In the combined precedence diagram, the precedence relation between task a and task b , where b is an immediate follower of a , can be expressed as follows:

$$\sum_{k=E_a}^{L_a} k \cdot V_{ak} - \sum_{k=E_b}^{L_b} k \cdot V_{bk} \leq 0$$

where $L_a \geq E_b$ and $E_a \leq E_b$. Note that the above assignment and precedence constraints are also utilized by Patterson and Albracht [7].

Cycle time constraints. The sum of the task performance times for each model within a station must be less than or equal to the cycle time of the model, and this can be expressed as follows:

$$\sum_{i \in W_{km}} t_{im} \cdot V_{ik} \leq C_m, \quad k = 1, \dots, K, \quad m = 1, \dots, P$$

Stations constraints. The number of stations is the same for all models; i.e., if the work content of station k for a model is zero, then the work content of this station for all the other models must also be zero. This can be accomplished by introducing the following constraints:

$$\sum_{i \in W_{km}} V_{ik} - \|W_{km}\| X_{km} \leq 0 \quad \text{for } k = 1, \dots, K, \quad m = 1, \dots, P$$

$$\sum_{m=1}^P X_{km} - P \cdot A_k = 0 \quad \text{for } k = 1, \dots, K$$

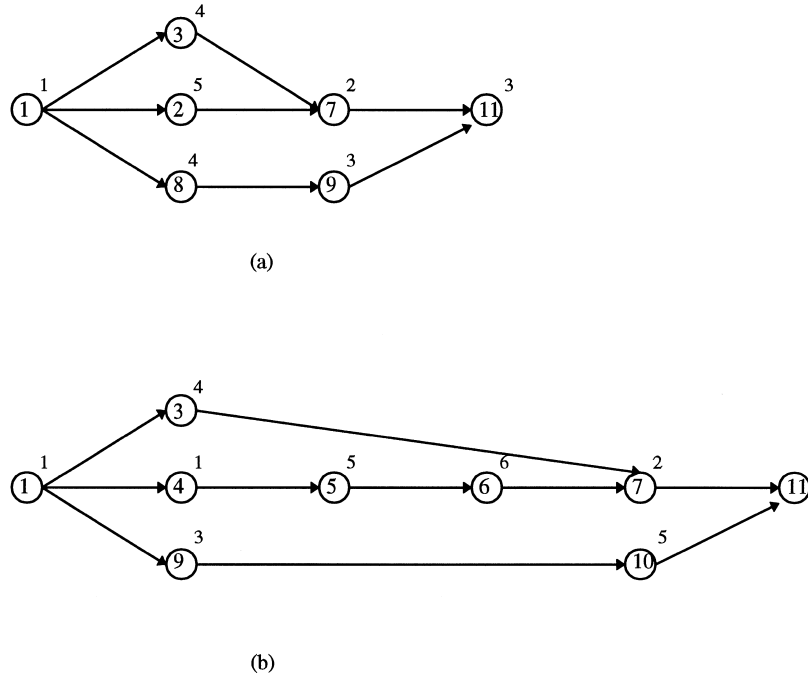


Fig. 3. Precedence diagrams of (a) model 1, (b) model 2, of the illustrative example.

Objective function

The objective is to minimize the number of stations utilized:

$$\text{Min } \sum_{k=1}^K A_k$$

ILLUSTRATIVE EXAMPLE

We apply the above model to a mixed-model assembly line balancing problem with two similar models taken from Bedworth and Bailey [18]. The precedence diagrams of the models and the combined diagram are depicted in Fig. 3 and Fig. 4, respectively. In Fig. 3, the numbers next to the nodes represent task performance times. Note that the combined diagram has 11 tasks, whereas the first and the second models have 7 and 9 tasks, respectively. Cycle time is taken as 10 minutes for each model and the number of stations is limited to 4.

The earliest and latest stations to which the tasks can be assigned to are given in Table 1.

The assignment constraints of the formulation are as follows:

$$V_{11} + V_{12} = 1$$

$$V_{21} + V_{22} + V_{23} + V_{24} = 1$$

$$V_{31} + V_{32} + V_{33} + V_{34} = 1$$

$$V_{41} + V_{42} + V_{43} = 1$$

$$V_{51} + V_{52} + V_{53} = 1$$

$$V_{62} + V_{63} = 1$$

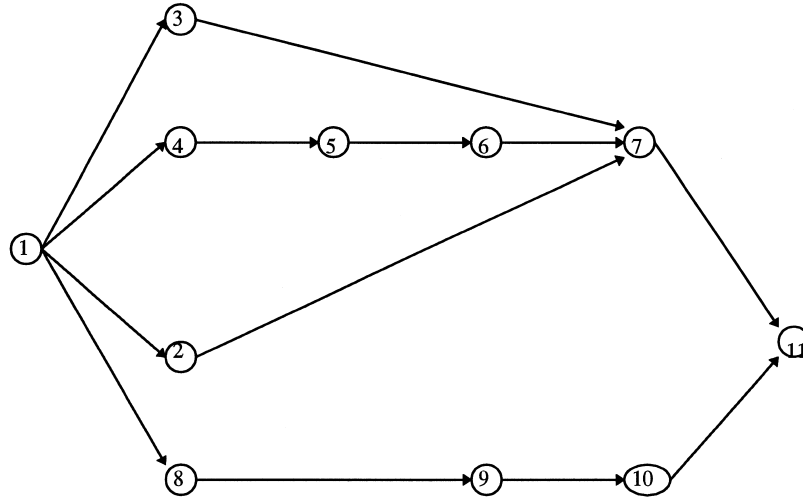


Fig. 4. Combined precedence diagram of the illustrative example.

$$V_{72} + V_{73} + V_{74} = 1$$

$$V_{81} + V_{82} + V_{83} + V_{84} = 1$$

$$V_{91} + V_{92} + V_{93} = 1$$

$$V_{101} + V_{102} + V_{103} + V_{104} = 1$$

$$V_{113} + V_{114} = 1$$

Note that the E_i and L_i values restrict the number of variables in the above constraints to 34. The precedence constraints of the formulation are as follows:

$$V_{11} + 2V_{12} - V_{31} - 2V_{32} - 3V_{33} - 4V_{34} \leq 0$$

$$V_{11} + 2V_{12} - V_{41} - 2V_{42} - 3V_{43} \leq 0$$

$$V_{11} + 2V_{12} - V_{21} - 2V_{22} - 3V_{23} - 4V_{24} \leq 0$$

$$V_{11} + 2V_{12} - V_{81} - 2V_{82} - 3V_{83} - 4V_{84} \leq 0$$

Table 1. The earliest and latest stations to which the tasks of the illustrative example can be assigned

Task (i)	E_i	L_i
1	1	2
2	1	4
3	1	4
4	1	3
5	1	3
6	2	3
7	2	4
8	1	4
9	1	3
10	1	4
11	3	4

$$V_{41} + 2V_{42} + 3V_{43} - V_{51} - 2V_{52} - 3V_{53} \leq 0$$

$$V_{51} + 2V_{52} + 3V_{53} - 2V_{62} - 3V_{63} \leq 0$$

$$V_{31} + 2V_{32} + 3V_{33} + 4V_{34} - 2V_{72} - 3V_{73} - 4V_{74} \leq 0$$

$$2V_{62} + 3V_{63} - 2V_{72} - 3V_{73} - 4V_{74} \leq 0$$

$$V_{21} + 2V_{22} + 3V_{23} + 4V_{24} - 2V_{72} - 3V_{73} - 4V_{74} \leq 0$$

$$V_{81} + 2V_{82} + 3V_{83} + 4V_{84} - V_{91} - 2V_{92} - 3V_{93} \leq 0$$

$$V_{91} + 2V_{92} + 3V_{93} - V_{101} - 2V_{102} - 3V_{103} - 4V_{104} \leq 0$$

$$2V_{72} + 3V_{73} + 4V_{74} - 3V_{113} - 4V_{114} \leq 0$$

$$V_{101} + 2V_{102} + 3V_{103} + 4V_{104} - 3V_{113} - 4V_{114} \leq 0$$

Each constraint above corresponds to a precedence relation in the combined precedence diagram. The cycle time constraints for the models are as follows:

Cycle time constraints (for model 1):

$$V_{11} + 5V_{21} + 4V_{31} + 4V_{81} + 3V_{91} \leq 10$$

$$V_{12} + 5V_{22} + 4V_{32} + 2V_{72} + 4V_{82} + 3V_{92} \leq 10$$

$$5V_{23} + 4V_{33} + 2V_{73} + 4V_{83} + 3V_{93} + 3V_{113} \leq 10$$

$$5V_{24} + 4V_{34} + 2V_{74} + 4V_{84} + 3V_{114} \leq 10$$

Cycle time constraints (for model 2):

$$V_{11} + 4V_{31} + V_{41} + 5V_{51} + 3V_{91} + 5V_{101} \leq 10$$

$$V_{12} + 4V_{32} + V_{42} + 5V_{52} + 6V_{62} + 2V_{72} + 3V_{92} + 5V_{102} \leq 10$$

$$4V_{33} + V_{43} + 5V_{53} + 6V_{63} + 2V_{73} + 3V_{93} + 5V_{103} + 3V_{113} \leq 10$$

$$4V_{34} + 2V_{74} + 5V_{104} + 3V_{114} \leq 10$$

The first and the last four constraints above are associated with the first and the second models, respectively. The station constraints of the formulation are as follows:

$$V_{11} + V_{21} + V_{31} + V_{81} + V_{91} - 5X_{11} \leq 0$$

$$V_{12} + V_{22} + V_{32} + V_{72} + V_{82} + V_{92} - 6X_{21} \leq 0$$

$$V_{23} + V_{33} + V_{73} + V_{83} + V_{93} + V_{113} - 6X_{31} \leq 0$$

$$V_{24} + V_{34} + V_{74} + V_{84} + V_{114} - 5X_{41} \leq 0$$

$$V_{11} + V_{31} + V_{41} + V_{51} + V_{91} + V_{101} - 6X_{12} \leq 0$$

$$V_{12} + V_{32} + V_{42} + V_{52} + V_{62} + V_{72} + V_{92} + V_{102} - 8X_{22} \leq 0$$

Table 2. Optimal station assignment of the illustrative example

Station	Model 1			Model 2	
	Tasks	Tasks	Station time	Tasks	Station time
1	1,4,5,8,9	1,8,9	8	1,4,5,9	10
2	3,6	3	4	3,6	10
3	2,7,10,11	2,7,11	10	7,10,11	10

$$V_{33} + V_{43} + V_{53} + V_{63} + V_{73} + V_{93} + V_{103} + V_{113} - 8X_{32} \leq 0$$

$$V_{34} + V_{74} + V_{104} + V_{114} - 4X_{42} \leq 0$$

$$X_{11} + X_{12} - 2A_1 = 0$$

$$X_{21} + X_{22} - 2A_2 = 0$$

$$X_{31} + X_{32} - 2A_3 = 0$$

$$X_{41} + X_{42} - 2A_4 = 0$$

Finally, the objective function of the formulation is as follows:

$$\text{Min } A_1 + A_2 + A_3 + A_4$$

The above formulation has 46 binary integer variables and 44 constraints. The optimal solution is shown in Table 2. Only three stations are utilized in the optimal solution; the total idle time associated with Model 1 and Model 2 are 8 and 0, respectively. Note that tasks 1 and 9 for station 1, task 3 for station 2, and tasks 7 and 11 for station 3 are common to both models.

PERFORMANCE OF THE MODEL

We have attempted to solve problems of various sizes using the General Algebraic Modeling System (GAMS) Release 2.25 on a 486 66 MHz personal computer. Table 3 depicts the sizes of the problems, the average CPU times and the number of iterations. The difficulty level of a problem is a function of the number of tasks on the combined precedence diagram and the number of precedence relations among the tasks. The precedence relations play a dominant role in specifying the computational and storage requirements of problems; for example, the requirements of

Table 3. Experimentation results

Number of tasks	F-ratio	Number of problems solved	Average CPU time (min)	Average number of iterations
10	0.644	4	5.83	22 980
10	0.444	4	3.51	19 639
10	0.113	4	1.70	7 610
20	0.710	6	31.50	81 993
20	0.536	6	12.90	51 846
20	0.147	6	7.80	28 317
30	0.751	6	43.37	144 365
30	0.441	6	37.63	134 793
30	0.112	6	15.07	62 669
40	0.801	3	77.35	250 000
40	0.510	3	53.01	202 841
40	0.173	3	42.17	137 000
60	0.828	3	180.75	450 000*
60	0.509	3	155.19	450 000*
60	0.070	3	153.33	450 000*

* Optimal solution was not obtained.

Table 4. Results of comparison between our model and the model of Roberts and Villa

Number of models	Total number of tasks	Number of tasks in combined diagram	F-ratio of combined diagram	Integer model		Roberts and Villa's model [11]	
				Number of constraints	Number of variables	Number of constraints	Number of variables
2	9	5	0.0	26	18	32	45
2	16	11	0.45	39	32	61	144
3	51	30	0.24	122	132	208	1020
3	61	28	0.28	125	154	248	1403
4	39	16	0.49	91	89	169	468
4	56	23	0.29	126	136	242	952

a problem that has several tasks with no precedence relations will be larger than the requirements of a problem that has a serial precedence diagram. The Flexibility ratio (F-ratio), developed by Dar-El [19], is a measure of the number of feasible sequences that could be generated from the precedence diagram. Thus, it can be used as a measure of computational and storage requirements of problems. If H is the number of zeros in this matrix, then the F-ratio is defined as

$$\text{F-ratio} = \frac{2H}{N(N-1)}$$

where N is the number of tasks in the problem. It ranges from one for precedence diagrams with tasks having no precedence relations to zero for precedence diagrams with tasks ordered serially. As depicted in Table 3, a wide range of F-ratio values is included in the experimentation. The number of stations utilized to specify the earliest and latest stations to which tasks can be assigned is determined by the well-known heuristic procedure "Ranked Positional Weight Technique" of Helgeson and Birnie [20]. The program terminates if the upper bound of 450 000 iterations is reached.

Examining Table 3 reveals the expected results that as the number of tasks and F-ratios increase, the computational requirements increase. Optimal solutions of the problems with up to 40 tasks have been obtained in less than 450 000 iterations. However, all the 60-task problems required more than 450 000 iterations to obtain the optimal solutions.

We have also compared the size of our model with that of the model of Roberts and Villa [11] on several problems. To the best knowledge of the authors, the model of Roberts and Villa [11] is the only integer programming model to solve the mixed-model version of the problem in the literature. In the model of Roberts and Villa, the concept of the earliest and latest stations to which the tasks can be assigned has not been utilized. The combined precedence diagram has also not been considered. Furthermore, the upper bound on the number of stations has been taken to be equal to the number of tasks. In our model, the above concepts are taken into account; thus, the size of our model is significantly smaller than that of the model of Roberts and Villa [11]. Table 4 depicts the number of constraints and variables of the models in various problems with up to four models. The difference in the total number of tasks and the number of tasks in the combined diagram is due to the common tasks among the models.

CONCLUDING REMARKS

We have developed a binary integer programming model for the mixed-model assembly line balancing problem in which some tasks are common to different models. We have attempted to decrease the size of the model by utilizing a combined precedence diagram and some variables that limit the increase in the number of decision variables and constraints. The resulting model is significantly superior to the one reported in the literature with respect to the number of decision variables and constraints. The experimentation revealed that the model is capable of solving problems with up to 40 tasks in the combined precedence diagram. On the other hand, due to the NP-hardness of the problem, the model size would be too large to obtain the optimal sol-

utions of larger problems. The model serves as a starting point for researchers in the field, and may be used as a validation tool for heuristic procedures.

REFERENCES

1. Salveson, M. E., The assembly line balancing problem. *Journal of Industrial Engineering*, **6**, 1955, 18–25.
2. Baybars, I., A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, **32**, 1986, 909–932.
3. Ghosh, S. and Gagnon, J., A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, **27**, 1989, 637–670.
4. Bowman, E. H., Assembly line balancing by linear programming. *Operations Research*, **8**, 1960, 385–389.
5. White, W. W., Comments on a paper by Bowman. *Operations Research*, **9**, 1961, 274–276.
6. Thangavelu, S. R. and Shetty, C. M., Assembly line balancing by zero–one integer programming. *AIIE Transactions*, **3**, 1971, 61–68.
7. Patterson, J. H. and Albracht, J. J., Assembly line balancing: zero–one programming with Fibonacci search. *Operations Research*, **23**, 1975, 166–172.
8. Talbot, F. B. and Patterson, J. H., An integer programming algorithm with network cuts for solving the assembly line balancing problem. *Management Science*, **30**, 1984, 85–99.
9. Gutjahr, A. L. and Nemhauser, G. L., An algorithm for the line balancing problem. *Management Science*, **11**, 1964, 308–315.
10. Jackson, J. R., A computing procedure for a line balancing problem. *Management Science*, **2**, 1956, 261–272.
11. Roberts, S. D. and Villa, C. D., On a multiproduct assembly line balancing problem. *AIIE Transactions*, **2**, 1970, 361–364.
12. Berger, I., Bourjolly, J. M. and Laporte, G., Branch-and-bound algorithms for the multi-product assembly line balancing problem. *European Journal of Operational Research*, **58**, 1992, 215–222.
13. Ding, F. Y. and Cheng, L., A simple sequencing algorithm for mixed-model assembly lines in just-in-time production systems. *Operations Research Letters*, **13**, 1993, 27–36.
14. Sumichrast, R. T. and Russell, R., Evaluating mixed-model assembly line sequencing heuristics for just-in-time production systems. *Journal of Operations Management*, **9**, 1990, 371–390.
15. Thomopoulos, N. T., Mixed-model line balancing with smoothed station assignments. *Management Science*, **16**, 1970, 593–603.
16. Macaskill, J. L. C., Production line balances for mixed-model lines. *Management Science*, **19**, 1972, 423–433.
17. Chakravarty, A. K. and Shtub, A., Balancing mixed model lines with in-process inventories. *Management Science*, **31**, 1985, 1161–1174.
18. Bedworth, D. D. and Bailey, J. E., *Integrated Production Control Systems*. Wiley, New York, 1982, pp. 281–293.
19. Dar-El, E. M., MALB—a heuristic technique for balancing large single-model assembly lines. *AIIE Transactions*, **5**, 1973, 343–356.
20. Helgeson, W. B. and Birnie, D. P., Assembly line balancing using ranked positional weight technique. *Journal of Industrial Engineering*, **12**, 1961, 394–398.