A Sequencing Problem for Mixed-Model Assembly Line with the Aid of Relief-Man*

Jae Kyu YOO**, Yoshiaki SHIMIZU** and Rei HINO**

There usually exist large variations in assembly times at mixed-model assembly lines depending on difference of product-models. To increase efficiency of line handling under such circumstance, this paper concerns with a sequencing problem for the mixed-model assembly lines where each product is supposed to be assembled within the same cycle time. Then, we formulate a new type of the sequencing problem minimizing the weighted sum of the line stoppage times and the idle times, and propose a new sequencing method with the aid of Relief Man (*RM*). Since the resulting problem refers to a combinational optimization problem, we develop a hybrid method that applies meta-heuristics like SA (Simulated Annealing) together with TS (Tabu Search) in a hierarchical manner. Finally, we examine the effectiveness of the proposed method through computer simulations and show the advantage of using *RM* against various changes in production environment.

Key Words: Line Handling, Mixed-Model Assembly Line, Sequencing Problem, Relief-Man, Meta-Heuristics

1. Introduction

Sequencing is viewed as an important stage for raising the efficiency of line handling of the assembly line where mixed-models are assembled every constant cycle time. In the mixed-model sequencing problem, one of the two major goals is to level the workload at each workstation on the assembly line against different assembly time per product-model (Miltenburg, 1989)(1). Another one is to keep the constant usage rate of every part at the assembly line (Duplag and Bragg, 1998)(2). Concerns about these two goals have been widely discussed in the literatures. For examples, the workload-leveling problem was addressed by Okamura and Yamashina (1978)(3). Yano and Rachamadugu (1991)(4) concerned with the problem that aims to minimize the risk of assembly line stop. Sumichrast and Russell (1990)(5) discussed the parts-usage smoothing problem. Moreover, the problem to attain these two goals simultaneously was discussed by Korkmazel and Meral (2001)(6). However, all these studies are not applicable in the case where large variations exist in the assembly times among different product-models. By installing a bypass line, Tamura (1999)(7) dealt with the inefficiency such as line stoppage or idle time of workers due to the difference in the assembly times. However, this approach causes another disadvantage that we need to remove the installation after it becomes unnecessary.

In this paper, we concern with a sequencing problem in the mixed-model assembly line that achieves the above two goals while overcoming the disadvantages mentioned above. That is, instead of installation of the bypass line as a hard facility, we will introduce *RM* (Relief Man) as a soft facility. Here, we define *RM* as multi-facility worker who supports general workers (called just *worker* hereinafter) on the assembly line. After formulating the problem, we will propose its practical solution method that can solve the combinational optimization regarding injection sequence of the mixed-models and traveling routes of *RM* to support *worker*s. Then, we develop a hybrid method that employs meta-heuristics like SA (Simulated Annealing) and TS (Tabu Search) in a hierarchical manner. Effectiveness of the proposed method is verified through some numerical experiments.

This paper is organized as follows. In section 2, we develop a model for the mixed-model assembly line with the aid of RM and formulate its sequencing problem as a mathematic programming model. In section 3, we introduce a solution method of realizing the sequencing with high practicality. Section 4 is devoted to the numerical

* Received 30th July, 2004 (No. 04-4175)
** Faculty of Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tenpaku, Toyohashi 441-8580, Japan.
E-mail: yoo@pse.tut.ac.jp

experiments.

2. Formulation by an Optimization Model

Figure 1 shows an example of the mixed-model assembly line where K workstations are linked with a conveyor moving at constant speed. Each product is fed to the assembly line from the first workstation every interval of cycle time (CT). When the *worker* of each workstation is unable to finish his/her work within CT, *worker* stops the assembly line and needs to wait for the remaining work.

We further assume the following conditions.

1. Total number of order is I.
2. The maximum number of parts used on the workstation is M.
3. There are two kinds of worker, i.e., *worker* and *RM*.
4. *Worker*s are confined to their workstation k ($=1,\ldots,K$) during assembly work, and their working time do not exceed CT.
5. When *worker* must do his/her work exceeding over CT, *RM* supports them.
6. *RM* can cover all workstations (multi-faculty), but cannot operate more than two workstations simultaneously (see Fig. 2).
7. The line stoppage occurs due to the part shortage and work delays of *RM* as well.
8. Idle time occurs when the assembly time of *RM* and/or *worker* is less than CT.

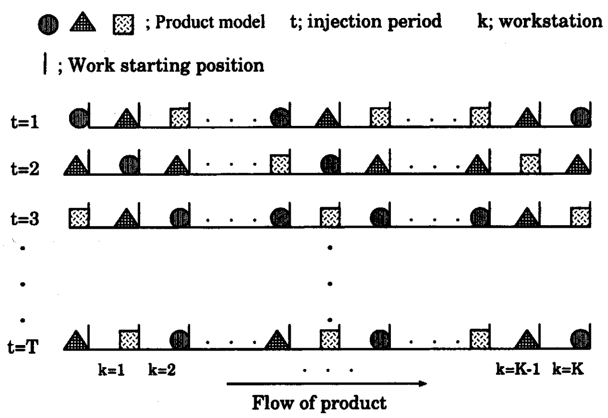

Fig. 1 Mixed-model assembly line

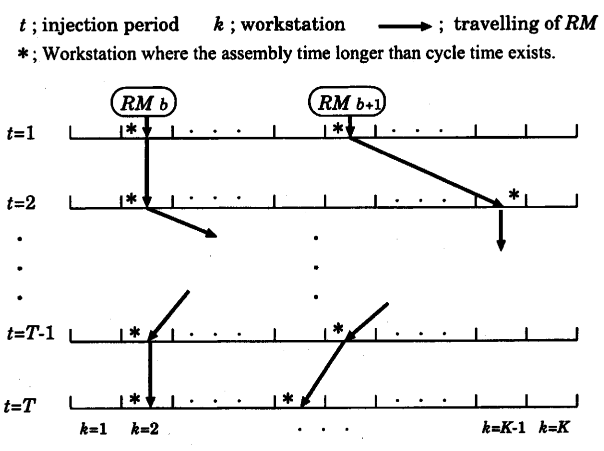

Fig. 2 Traveling of *RM* for workstations

After all, our sequencing problem can be formulated as follows:

$$\min_{\pi\in\Pi} z=\rho\times\sum_{t=1}^{T}\left\{\max_{1\le k\le K}\left(P_k^t,\ \sum_{b=1}^{N^t}X_{bk}^t A_k^t,\ 0\right)\right\} +(1-\rho)\times\sum_{t=1}^{T}\left\{\sum_{k=1}^{K}D_k^t+R^t+\sum_{b=1}^{N^t}\sum_{k=1}^{K}X_{bk}^t G_k^t\right\} \quad (1)$$

s.t

$$\sum_{k\in S^t} X_{bk}^t=1, \qquad b=1,\ldots,N^t,\ t=1,\ldots,T \quad (2)$$

$$\sum_{b\in N^t} X_{bk}^t=1, \qquad k=1,\ldots,K,\ t=1,\ldots,T \quad (3)$$

$$w_k^t\ge(t-1)\times CT, \quad k=1,\ldots,K,\ t=1,\ldots,T \quad (4)$$

The notation is summarized in the following.

Π : set of sequence ($\pi\in\Pi$)
P_k^t : line stoppage time by the part shortage in workstation k at injection period t ($=1,\ldots,T$)
A_k^t : line stoppage time by the work delay of *RM* in workstation k at injection period t
D_k^t : idle time of *worker* in workstation k at injection period t
R^t : idle time of *RM* whose work is not assigned at injection period t
G_k^t : idle time of *RM* whose finishing time is earlier than CT in workstation k at injection period t
S^t : set of workstation that needs *RM* at injection period t
w_k^t : work starting time of *RM* in workstation k at injection period t
X_{bk}^t : 0–1 variable that takes 1 if *RM* b is assigned to workstation k at injection period t, otherwise, 0
ρ : weighting factor ($0<\rho<1$)

The objective function Eq. (1) is given as the weighted sum of the line stoppage time and the idle time. Sequence π and 0–1 variable X_{bk}^t are the decision variables of this problem. Among the constrains, Eqs. (2) and (3) are the conditions of generalized assignment, and Eq. (4) is regarding the starting time of *RM*'s work. Moreover, each time will be described below in detail.

Figure 3 illustrates a feature that the part shortage will happen when the quantity of part m used actually ($\sum_i a_{im}^k x_i^t$) exceeds its ideal quantity (tr_m^k) at the injection period. In this case, P_k^t is given as Eq. (5):

$$P_k^t=\max\left(\max_{1\le m\le M}\left(\frac{\sum_{i=1}^{I}a_{im}^k x_i^t-tr_m^k}{r_m^k}CT\right),0\right), \quad (5)$$

where a_{im} is the quantity of part m required per model i; x_i^t is the cumulative amount of production for model i during

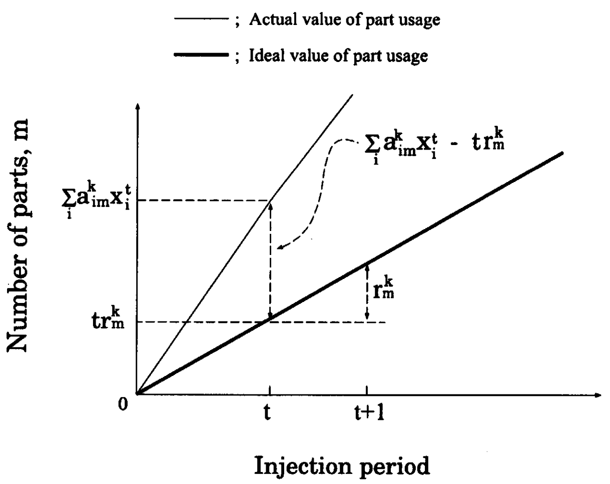

Fig. 3 Calculation of line stoppage time based on goal chasing method

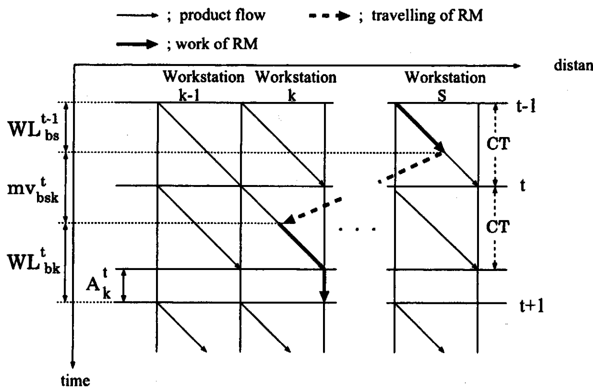

Fig. 4 Line stoppage due to work delay of *RM*

injection period from 1 to t, and r_m is the ideal usage rate of part m. In Fig. 4, let mv_{bsj}^t (see Eq. (6)) be the traveling time of *RM* b who moves from workstation s to k during injection period from $t-1$ to t, and WL_{bs}^{t-1} and WL_{bk}^t the working time of *RM* b on s at $t-1$ and k at t, respectively. A_k^t is given by Eq. (7):

$$mv_{bsk}^t = |WE_{bs}^{t-1} - WS_{bk}^t| \times O, \qquad (6)$$

$$A_k^t = \max(mv_{bsk}^t + WL_{bs}^{t-1} + WL_{bk}^t - 2CT, 0),$$
$$\forall s \in S^{t-1},\ b = 1, 2, \ldots, N^t, \qquad (7)$$

where WE_{bs}^{t-1} denotes the completion time of work on the workstation s at the injection period $t-1$, and WS_{bk}^t the start time of work on k at t, and O the travel time of *RM* per workstation. The idle time of *worker* occurs when the work time denoted by L_k^t on workstation k at injection period t exceeds CT. Consequently, D_k^t is given as Eq. (8).

$$D_k^t = \max(CT - L_k^t, 0), \qquad (8)$$

Moreover, the idle time of *RM* is classified into two categories, i.e., that of "off *RM*" (R^t) and of "standby *RM*" (G_k^t). The former is given by Eq. (9) where N^t denotes the *RM* number working at the injection period t and B the maximum number of *RM* necessary throughout the production planning horizon. The latter is equivalent to the idle time of *RM* b who cannot begin the work yet at the injection period t even though he or she has reached the workstation k from s (see Fig. 5), and given by Eq. (10).

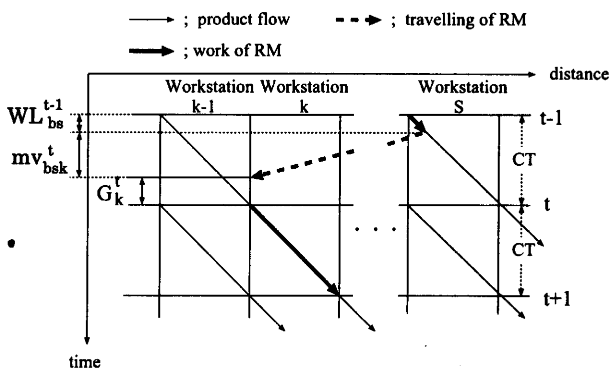

Fig. 5 Idle time of RM

$$R^t = (B - N^t)CT, \qquad (9)$$

$$G_k^t = \max(CT - mv_{bsk}^t - WL_{bs}^{t-1}, 0),$$
$$\forall s \in S^{t-1},\ b = 1, 2, \ldots, B, \qquad (10)$$

3. Solution Method

Since the above combinatorial problem becomes NP hard, an approximated solution method is more desirable than the rigid one for practical applications. From this aspect, we have developed a hybrid method that divides the original problem into two sub-problems in a hierarchical manner so that we can apply the efficient solution methods for each of them. Its upper level solves a sequencing problem based on SA (Simulated Annealing), and the lower a route selection problem of *RM* using TS (Tabu Search) and NNM (Nearest Neighbor Method; Yanaura and Ibaraki, 2001)[8]. Moreover, we apply the goal chasing method to derive an initial sequence for effective search in SA, and the levering method to decide the *RM* number. Below, each algorithm will be explained in detail (see Fig. 6).

3.1 Proposed solution algorithm

3.1.1 Upper level

Step 1: Generate the initial sequence (solution) based on the goal chasing method.

Step 2: Go to lower level.

Step 3: Check the convergence condition about SA, and stop if satisfied. Otherwise, generate a new sequence using the swap neighborhood, and go back to Step 2.

3.1.2 Lower level

Step 1: Calculate the maximum number of *RM* required throughout the production planning horizon after smoothing the workloads under the given sequence by the levering method (see Fig. 7).

Step 2: Initialize tabu-list where the route of all *RM*s is memorized as a solution.

Step 3: Generate each initial traveling route for the *RM*s using the nearest neighbor method.

Step 4: After generating neighborhood solutions, choose a solution having the minimum objective function value from these.

Step 5: If the convergence condition about TS is satisfied, go to Step 6. Otherwise, update the tabu-list, and go back to Step 4.

Step 6: Investigate the convergence condition about the *RM* number, and go back to Step 3 of upper level if satisfied. Otherwise, change the *RM* number, go back to Step 2.

To decide the initial traveling route of *RM* in the above Step3, we use the following algorithm based on NNM.

3.2 Initial route selection algorithm

Step3-1: Decide workstations k^t_j ($\in S^t, j=1,2,\ldots$) where the assist of *RM* is necessary per each injection period under the given sequence. Allocate each *RM* at random to k^t_j ($t=1$).

Step3-2: Let $t=t+1$. After making each pair of *RM* and k^t_j, P^t_{bj} ($b=1,2,\ldots,B$), generate every combination C^t_v ($v=1,2,\ldots$) (P^t_{bj}, P^t_{xy} ($x,y=1,2,\ldots$); $b\neq x, j\neq y$).

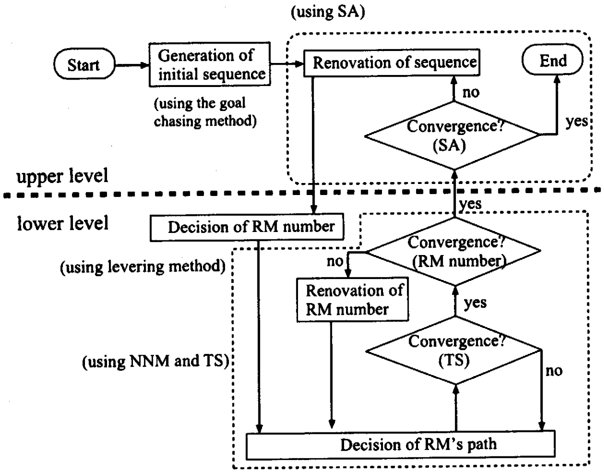

Fig. 6 Proposed solution method

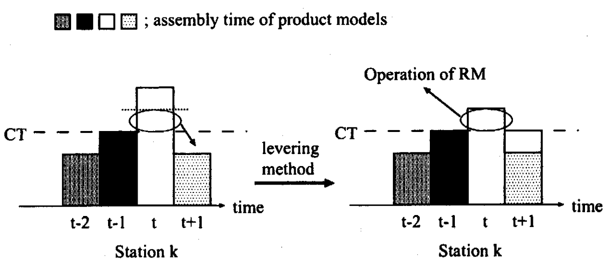

Fig. 7 Arising operation for *RM* after levering method

Step3-3: Evaluate each C^t_v on the basis of objective function value, and decide the assignment based on the combination having the minimum value. Here, when there exist multiple minima, decide one according to the following priority.

3.3 Priority from the top to the bottom

(1) One with the smallest sum of *RM*'s travailing time from $t-1$ to t.

(2) One whose the work start time of *RM* at t is nearest to $t\cdot CT$.

(3) Arbitrary.

Step3-4: If $t=T$, stop. Otherwise, go back to Step3-2.

4. Numerical Experiments

Numerical experiments are taken place under the conditions shown in Tables 1 and 2. We also give the parameters about SA and TS in Table 3. And, the weighting factor ρ in Eq. (1) is set at 0.9. Moreover, we evaluated the results on the basis of average over 100 data sets generated randomly.

4.1 Effect of using *RM*

To examine the effectiveness of using *RM* regarding the line stoppage time and the value of objective function, we compared the results with those without *RM*. From Table 4, the existence of *RM* is known to realize the efficient assembly line management, i.e. drastic decrease in

Table 1 Input parameter

Cycle time[min]	3
Station number	20
Product model(A~E)	5
Total production number	100
Part number	10
Part number used per workstation	0~2
Travelings time of RM per workstation[min]	0.1

Table 2 Production number and assembly times

Product model	A	B	C	D	E
Production rate in total number	0.1	0.2	0.4	0.2	0.1
Assembly time per workstation [min]	1	1~6	1~6	1~6	6

Table 3 Parameters of SA and TS

SA	Initial temperature	0.95
	Annealing ratio	0.95
	Number of search per every temperature	100
	Number of annealing	100
TS	Number of search	100
	Number of tabu-list	20
	Number of neighborhood	10

Table 4 Effect of the existence of *RM*

	With 4*RM*	Without *RM*
Objective function value, Eq.(1)	136.6	179.87
*LST** by delayed work[min]	8.9	100.6

* ; line stoppage time.

Table 5 Necessary *RM* number VS. total production number and rate of product model E

Total product number	10	30	100				
*Rate**	-	-	0.1	0.2	0.3	0.4	0.5
Necessary *RM* number	2	3	4	6	9	9	11

* ; Rate of model E in total production number.

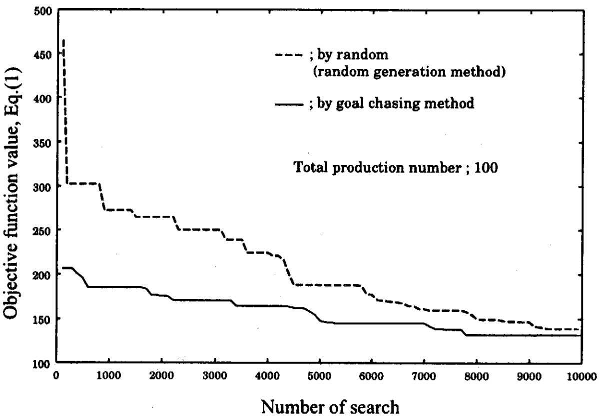

Fig. 8 Comparison of convergence between two generating methods for initial sequence

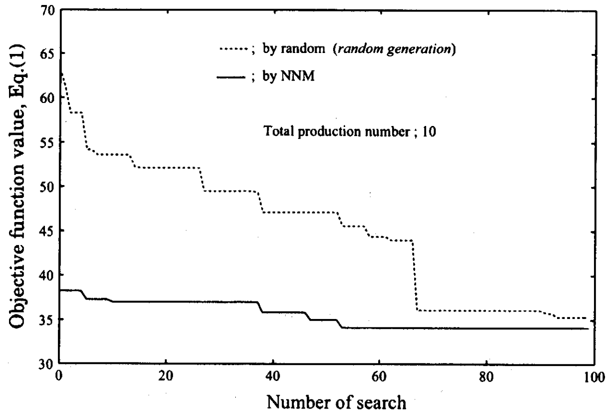

Fig. 9 Comparison of convergence between NNM and *random generation*

Table 6 Comparison between proposed solution method and total enumeration method

	Product number					
	8		30		100	
	*PM**	*TE**	*PM**	*TE**	*PM**	*TE**
Objective function value, Eq.(1)	24.9	20.8	64.4	-	136.6	-
CPU time[sec]	126	45190	1124	-	5392	-
Necessary *RM* number	2	2	3	-	4	-

*PM** ; Proposed solution method (Figure 6).
*TE** ; Total enumeration method.

the line stoppage time, and pretty improvement in the objective function value. This means that introduction of *RM* can realize leveling of the unbalanced workloads every injection period.

Table 5 shows the changes in the necessary *RM* number according to the increases in total production numbers and the product model E with the longest assembly time among all models. As supposed a priori, the necessary *RM* number also increases accordingly with the growth of the total production number and the rate of model E as well. However, each increasing rate is small compared with that of its number or that of the rate of model E. From these results, we know that assigning appropriate numbers of *RM* is effective and essential in dealing with the production fluctuations.

4.2 Discussion of the proposed solution procedure

Figure 8 shows the convergence process when the goal chasing method is used to generate the initial sequence. The horizontal axis expresses the number of searches and the vertical axis the objective function value. The goal chasing method has a rapid convergence compared with random generation method that the initial sequence is generated at random. This is because a practical solution can be obtained within the reasonable time since the search process begins from a sequence (solution) that has the minimum line stoppage against parts shortage.

To verify the effectiveness of the NNM algorithm in deciding the initial traveling route of *RM*, we compared in Fig. 9 the result with the case where the route is generated at random (hereinafter, *random generation*). The NNM algorithm outperforms the *random generation* about the convergence speed. This is due to the fact that starting the search process from a route having the shorter traveling time can reach a near optimum solution more rapidly.

Moreover, in Table 6, we illustrate the computation time (CPU time) and the objective function value about total enumeration method and the proposed solution method (Fig. 6) that applies SA together with TS in a hierarchical manner when the total production number increases. Regarding the CPU time, the case of the proposed solution method is shorter than that of total enumeration method, and the difference between each objective function value is very small when the total production number is 8. From the above results, we can confirm that the proposed solution method gives us a near optimum solution when the problem scale is small, and can expect that it obtains an approximate solution within the realistic computation time compared with the case of the total enumeration method when its scale is large.

5. Conclusion

In this paper, we have considered a sequencing problem of the mixed-model assembly line where the assembly time differs greatly depending on the product-models. After introducing *RM* as a key factor for the problem-solving, we formulated this problem as a combinatorial optimization problem minimizing the line stoppage time and the idle time simultaneously through the leveling of part usage and workload. Then we have proposed a practical solution method that uses the meta-heuristic methods in a hierarchical manner and can solve the real-life problems in a reasonable computation time. Through numerical experiments, our approach is known to have a great effect on reduction of the line stoppage time. Finally, we emphasize another advantage of the proposed approach such that the production changes due to the demand fluctuations for example can be managed just by changing dynamically *RM* number.

References

(1) Miltenburg, J., Level Schedules for Mixed-Model Assemble Lines in Just-in-Time Production Systems, Management Science, Vol.35, No.2 (1989), pp.192–207.

(2) Duplaga, E.A. and Bragg, D.J., Mixed-Model Assembly Line Sequencing Heuristics for Smoothing Component Parts Usage, Int. J. of Production Research, Vol.36, No.8 (1998), pp.2209–2224.

(3) Okamura, K. and Yamashida, H., A Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor, Int. J. of Production Research, Vol.17, No.3 (1979), pp.233–247.

(4) Yano, C.A. and Rachamadugu, R., Sequencing to Minimize Work Overload in Assembly Lines with Product Options, Management Science, Vol.37 (1991), pp.572–586.

(5) Sumichrast, R.T. and Russsell, R.S., Evaluating Mixed-Model Assembly Line Sequencing Heuristics for Just-in-Time Production Systems, J. of Operational Research, Vol.9 (1990), pp.371–399.

(6) Korkmazel, T. and Meral, S., Bicriteria Sequencing Methods for the Mixed-Model Assembly Line in Just-in-Time Production Systems, European J. of Operational Research, Vol.131, No.1 (2001), pp.188–207.

(7) Tamura, T., Long, H. and Ohno, K., A Sequencing Problem to Level Part Usage Rates and Work Loads for a Mixed-Model Assembly Line with a Bypass Subline, Int. J. of Production Economics, (1999), pp.557–564.

(8) Yanaura, M. and Ibaraki, T., Combinatorial Optimization, (in Japanese), (2001).