

# The Work-Sharing in a Serial Production Line with Consideration of the Balance of Fixed Workload and the Accuracy of Information

Salah Kasmoo, Takao Enkawa, and Sadami Suzuki

**Abstract**—This paper addresses the interaction effect between the workload and the information accuracy in work-sharing environment on the performance of a serial production line. The work-sharing is organized by Dynamic Line Balancing (DLB) policy. Under this policy, the workers can help each other with some tasks, called shared tasks. We investigated two levels of information accuracy that support the decision of working on or passing the shared tasks to the next buffer. We employed different configurations of workload under these two levels of information accuracy. A measure of workload is proposed to ease the comparison. The results of simulation showed a remarkable influence of workload on the performance and found its patterns as work-in-process (WIP) level changes. These patterns are clearly affected by changing the information accuracy level.

**Index Terms**—Work-sharing, information accuracy, dynamic line balancing, fixed workload.

## I. INTRODUCTION

The trend to producing variety of products with small quantities, reducing the WIP that is necessary to achieve the targeted throughput, increasing the utilization has magnified the importance of the work-sharing. Actually, this concept originally appeared in cell manufacturing which is one of JIT revolution's outcomes.

The highest performance in these manufacturing environments can be achieved as much as workers approach full cross-trained level. Moreover, the inventory is considered an absolute evil that represents the Japanese perspective and their strategy is to eliminate all factors that necessitate the inventory. Whereas, in western production systems it is a necessary evil as it makes production running smoothly [1]. Practically, having fully cross-trained workers is expensive since it consumes time and money. Additionally, the fast change of products will keep the need to continue training in high expenses, and on the other hand, the worker's speed will be slower with many tasks (full cross-trained) than that with a limited number of tasks (partially cross-trained) with limited amount of WIP.

Among different ways of applying work-sharing, we focused in this research on the way that considers the previous point (partially cross-trained), which is referred to as Dynamic Line Balancing (DLB). In DLB, introduced by [2],

the workers stay at their stations and there is no movement between stations. Each worker is assigned to fixed tasks of a job that can be done only by him. He can also help the upstream and downstream workers in shared tasks after finishing the fixed ones. A worker chooses to pass on a job with the shared task undone or completes the shared task according to specific rules depending on system information. This way of work-sharing is in somehow similar to fixed task zone chaining introduced by [3] but here the flow of job is unidirectional and no movement between stations. Reference [3] concluded that this design can nearly reach the performance of a fully cross-trained system.

Reference [2] found that by using DLB with a half-full buffer (HFB) control rule, the Work-In-Process (WIP) inventory can be reduced and the efficiency can be improved. Reference [4] showed that DLB can increase the efficiency even with no buffer. They used a new model called subtask model.

Two types of models in term of information accuracy level used to make a decision of passing or working on the shared task have appeared in the literature; model A (as in [2], [4]) and model B (as in [4]). In model A, the decision is supported by the workcontent in downstream buffer only while in model B it is supported by the workcontent available to the downstream station, which includes the one in the buffer and in the downstream station under processing not completed. Reference [5] proposed a new control rule with the model A, called SRNS (Smallest R No Starvation). It showed that this rule performs well by comparing its resulting performance with the optimal performance. They also compared in a later paper [6] between models A and B, and also between SRNS and HFB with constant-WIP (CONWIP) policy.

Reference [7] compared between HFB and 50-50 rule which utilizes the ratio of workcontent at the upstream station to total workload in the system. It also found three factors that have a notable effect on the work-sharing's results. These factors are preemptability of shared task, granularity of shared task, and processing time variability.

In this paper, we studied an interaction influence of two new factors; workload and information accuracy. The workload refers here to the sizes of fixed tasks of adjacent stations. Reference [6] demonstrated by using some examples how DLB can remedy the loss in the efficiency due to imbalanced fixed tasks. Also [7] treated this factor so briefly by one example and found that the performance gets bad when the fixed workload is not balanced. It also used the workload to compare between HFB and 50-50 rules. We studied the workload in a completely different way than that in [7].

Manuscript received November 9, 2013; revised January 9, 2014.

The authors are with the Industrial Engineering and Management Department, Tokyo Institute of Technology, 2-12-1 Oh-okayama, Meguro-ku, Tokyo 152-8552 Japan (e-mail: {kasmoo.s.aa, enkawa.t.aa, Suzuki.s.ag}@m.titech.ac.jp).

Reference [8] studied a pinion cell with two and three workers. The results ended that the workload assigned to the individual worker and the level of shared workload are significant factors determining the performance.

We try in this study to explore the workload's influence in DLB environment on the efficiency under two levels of information accuracy and find out the efficiency's patterns. The workload has not been addressed in a full picture before and if it is mentioned, it is so briefly. In the previous researches, the balanced workload is mostly focused on which is a special case. However, the real production environments could encounter a variety of workload configurations. Considering the information accuracy helps getting more resemble model to the reality and its different choices, and discovers its effect on the different workload configurations' performance. Here, several configurations are examined with different shared task' sizes. We use two easy to apply threshold rules (SRNS, HFB) where each needs a different level of information accuracy. A measure of workload is proposed which facilitates tackling the objective.

The remaining of the paper is organized as follows Section II explains the modeling assumptions, describes the threshold rules, and the workload measure is clarified. Section III gives simulation results and analysis. The last section summarizes the outcomes.

## II. SIMULATION ENVIRONMENT

### A. Modeling Framework

A two-station production line is considered. One worker is at each station. Worker 1  $W_1$  attends at station 1 and worker 2  $W_2$  attends at station 2. The buffer capacity is infinite, but the total number of jobs in the line is restricted to  $N$  number of jobs. This inventory level is reserved by the CONWIP discipline. CONWIP, introduced by [9], keeps MaxWIP constant by preventing a new job to enter the line until a finished job leaves when the number of jobs reaches to Max WIP level.

We utilize small numbers of  $N$ ; 3, 4, and 5 since the goal is exploring the work-sharing effect and where the high level of WIP in the line will hide the work-sharing influence and result long cycle time and other side consequences accompanying that. Reference [10] demonstrated the bad effect of having large amount of WIP on the balanced use of cross-trained skills.

Each job has three types of tasks, A, B, C. Task A can only be performed by  $W_1$  at station 1 and task C can only be performed by  $W_2$  at station 2. Both of tasks A and C are called fixed tasks. Whereas task B can be done by either worker at his own station and it is called shared task. We assume that the tooling/ parts are available at the stations to operate task B and the distance between adjacent stations is negligible.

The subtask model is used. In this model [4], a job consists of  $T$  number of equal subtasks to be performed in sequence. For example, a 2-3-4 task division refers to two subtasks to task A, three subtasks to task B, and four subtasks to task C. The subtasks times are identically distributed, for instance, if they are exponentially distributed, the task will be Erlang- $k$  distributed where  $k$  represents the number of subtasks that

compose the task.

The job is processed in first come first serve (FCFS) queue. The shared task B is non-preemptive task. When a worker starts processing the shared task, it can be released until it becomes completed. The job can be sent from upstream to downstream but it cannot return upstream. The workers' speeds are equal and  $W_1$  is the one who decides to pass on or keep working on the shared task.

### B. Threshold Rules and Information Accuracy

Dynamic Line Balancing is used as a mechanism to manage the work-sharing here. In DLB,  $W_1$  after finishing task A has to make a decision whether to pass on or continue processing task B. This decision considers the current state of the system. Appropriate decisions [5], [6] will give more work to station 2 when it is more available or potentially so, or pass less work onto it when otherwise. In other words, it tries to reduce the starvation with fewer buffers.

On other side, it is not easy to make the best decision for all cases since this process is state-dependent and complicated to compute and implement in reality [7]. Threshold rules are more practical, where the decision is made based on the comparison between the available downstream workload and a specific threshold called  $R$ . Many threshold rules are found in the literature. These rules can be distinguished based on the information accuracy level required. Mostly, one of two levels of the information accuracy is needed to make the decision process to pass or keep the shared task. The first level requires the information about the available subtasks for station 2 in the buffer before it only. Whereas, the second is more accurate and includes the information about the available subtasks for station 2 in the buffer before it and the subtasks at station 2 not yet completed.

Among of threshold rules from these two categories, we used one rule from each class that is the most near optimum based on [6]. Of the first level, SRNS is used and of the second HFB is used. In SRNS (HFB) [6],  $W_1$  will start or keep working on the shared task if  $R$  or more units of subtasks are available in the buffer before (and at) station 2. And  $W_1$  will pass the shared task if the subtasks in buffer before (and at) station 2 are less than  $R$ . The threshold value  $R$  is given by (1) for SRNS and (2) for HFB, where  $N$  is the CONWIP level and  $t_c$  ( $t_b$ ) is the number of subtasks for task C (B):

$$R = \begin{cases} (N-2)t_c, & \text{if } t_b \geq (N-2)t_c, \\ [t_b, (N-2)t_c] & \text{if } t_b < (N-2)t_c \end{cases} \quad (1)$$

$$R = \frac{t_b}{2} + \frac{(t_b + t_c) + t_c}{2} \cdot \frac{N-2}{2} \quad (2)$$

### C. Workload Measure

Since the shared work is managed by DLB, the fixed work is mainly focused in this paper. Workload refers here to the size of fixed tasks and their distribution between the stations. For example, a 3-4-3 task division has two equal fixed tasks whereas 4-3-3 has different fixed tasks. 3-3-4 task division has the same shared task's size as 4-3-3 but the distribution of the fixed workload is opposite to the former division.

In this paper, a measure is proposed to ease investigating

and comparing between studied workload configurations. The measure starts from a theoretical assumption that all task divisions can be return to the case of balanced line with no shared task and then the shared task is composed from some last subtasks of station 1 and some first subtasks of station 2. For example, as in Fig. 1 (2-4-4 task division), the virtual breakeven point divides the job's subtasks into two equal groups of subtasks.

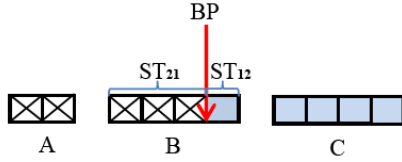


Fig. 1. Explanation of workload measure for a two-station line.

The shared task consists of three subtasks from station 1's side and one subtask from the other side.  $ST_{12}$  ( $ST_{21}$ ) represents the size of shared task that can be done by worker 1(2) from station 2 (1)'s side.

The measure of workload  $\alpha_{ij}$  is defined by the ratio of the shared subtasks' size done by worker ( $i$ ) from station ( $j$ ) side to the total shared task's size. The measure is given in the following equations (3), (4);

$$\alpha_{ij} = \frac{ST_{ij}}{B} \tag{3}$$

$$B = ST_{ij} + ST_{ji} \tag{4}$$

To explore several cases and get more general results,  $ST_{12}$ ,  $ST_{21}$  each is given four values; 1, 2, 3, and 4. Therefore, we get 24 cases. Table I presents some of these cases with calculated  $\alpha_{12}$ ,  $\alpha_{21}$  using (3).

TABLE I: SOME EXPERIMENTAL CASES

A	B	C	$ST_{12}$	$ST_{21}$	$\alpha_{12}$	$\alpha_{21}$
2	3	5	0	3	0	1
4	4	2	3	1	0.75	0.25
4	5	1	4	1	0.8	0.2
2	6	2	3	3	0.5	0.5
2	7	1	4	3	0.571	0.428

### III. SIMULATION RESULTS AND ANALYSIS

Visual Slam language (A simulation language) through AweSim software [11] is used to model and execute the simulation. The workers are working for 8 hours per day. The simulation is run for one year (1year  $\times$  250 days  $\times$  8 hours  $\times$  60 min = 120,000 min) with four replications and the warm-up period is 10000 min.

The processing time of a subtask is exponentially distributed with mean of one, and a task is Erlang- $k$  where  $k$  is equal to the number of subtasks composing this task. The total processing time of the job is 10 minutes plus the variability. To evaluate the performance, the efficiency is considered as a measure. It is defined as the ratio of simulated throughput rate over maximum achievable with balanced, deterministic processing times.

The studied cases are combined by considering the average efficiency into five combinations; ( $\alpha_{12}=0$ ,  $\alpha_{21}=1$ ), ( $\alpha_{12}=1$ ,  $\alpha_{21}=0$ ), ( $\alpha_{12}=\alpha_{21}=0.5$ ), ( $\alpha_{12}=0.25$ ,  $\alpha_{21}=0.75$ ), and ( $\alpha_{12}=0.75$ ,  $\alpha_{21}=0.25$ ). The average results of efficiency show three distinguished patterns according to WIP level. The first, when  $N$  equals 3. Fig. 2 and Fig. 3 present the results under HFB and SRNS. The plots show that as  $\alpha_{12}$  increases the efficiency rises until it reaches 0.75 then it deteriorates.  $\alpha_{21}$  has an inverse pattern since  $\alpha_{12} + \alpha_{21} = 1$ . It increases until 0.25 then starts decreasing. The highest efficiency is achieved for ( $\alpha_{12}=0.75$ ,  $\alpha_{21}=0.25$ ) and ( $\alpha_{12}=\alpha_{21}=0.5$ ) in HFB and ( $\alpha_{12}=0.75$ ,  $\alpha_{21}=0.25$ ) only in SRNS. The lowest efficiency is achieved for ( $\alpha_{12}=0$ ,  $\alpha_{21}=1$ ) in both rules.

By comparing between ( $\alpha_{12}=0$ ,  $\alpha_{21}=1$ ) and ( $\alpha_{12}=1$ ,  $\alpha_{21}=0$ ), we find that the second combination presents higher efficiency. The same is for ( $\alpha_{12}=0.25$ ,  $\alpha_{21}=0.75$ ) and ( $\alpha_{12}=0.75$ ,  $\alpha_{21}=0.25$ ). The reason might be that when  $\alpha_{12}$  is 0 or small, it means a large fixed task for station 2. As a result, this large fixed task will cause a smaller throughput since it becomes bottleneck. This can be alleviated as the fixed task of station 2 gets smaller or in other words as  $\alpha_{12}$  becomes bigger. However, this amount of fixed task or  $\alpha_{12}$  has a limit which is 0.75 after that the effect gets inversed as station 1 will be the bottleneck. Beside to the above reason, the small number of WIP in the line deepens this difference, since the station 1 will starve.

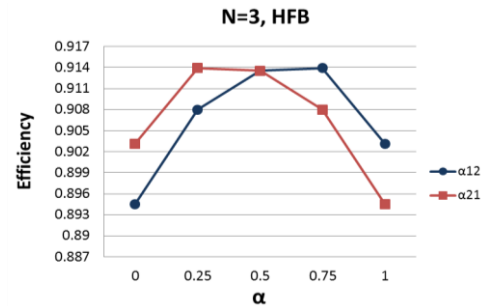


Fig. 2. The change of efficiency with  $\alpha$  for HFB under  $N=3$ .

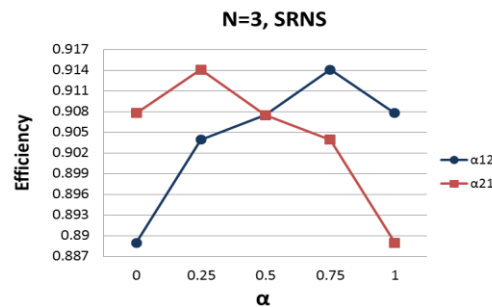


Fig. 3. The change of efficiency with  $\alpha$  for SRNS under  $N=3$ .

We noticed also that the case with balanced workload ( $\alpha_{12}=\alpha_{21}=0.5$ ) has better performance in HFB than that in SRNS. The cause returns to that the workload's effect is minimal in this case since each worker has the same probability to starve. As a result, the accurate information  $t$  used in HFB helps reducing the opportunity of both workers' starvation by sending the shared tasks as exact as  $W_2$  does need but without causing long starvation for  $W_1$ . For example, if  $W_2$  has just started processing a job, the buffer before it will become empty. If at the same time  $W_1$  has finished his fixed

task, (in HFB) he will send the job with considering how many subtasks still under processing at station 2. If they are enough, the job will not be sent. On the other hand, in SRNS  $W_1$  will surely send it if the buffer is empty which makes him starving longer.

Another interesting point, the performance of  $(\alpha_{12}=1, \alpha_{21}=0)$  in HFB is less than the one in SRNS. Also by comparing this combination with  $(\alpha_{12}=0.25, \alpha_{21}=0.75)$  in both cases, it is obvious that the results of that combination in HFB is less than  $(\alpha_{12}=0.25, \alpha_{21}=0.75)$ . The combination  $(\alpha_{12}=1, \alpha_{21}=0)$  has a big portion of subtasks for  $W_1$  (5 subtasks) while relatively small number of subtasks for  $W_2$  and  $R$  is small under  $N=3$ . This means that  $R$  could be close to fixed task of  $W_2$ . In the light of what mentioned above and since in HFB the uncompleted subtasks at station 2 are considered to make a decision to send or keep, the chance of having empty upstream buffer is more than in SRNS. As a result,  $W_2$  has a high possibility to starve longer in HFB.

The second case is when  $N$  is 4. Here as in Fig. 4 and Fig. 5, the differences between the opposite combinations get so small.  $(\alpha_{12}=0.25, \alpha_{21}=0.75)$ ,  $(\alpha_{12}=0.75, \alpha_{21}=0.25)$  and  $(\alpha_{12}=0, \alpha_{21}=1)$ ,  $(\alpha_{12}=1, \alpha_{21}=0)$  have a small difference in their efficiency.

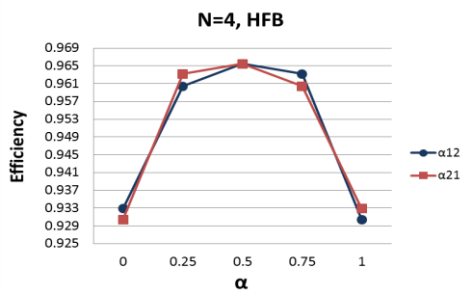


Fig. 4. The change of efficiency with  $\alpha$  for HFB under  $N=4$ .

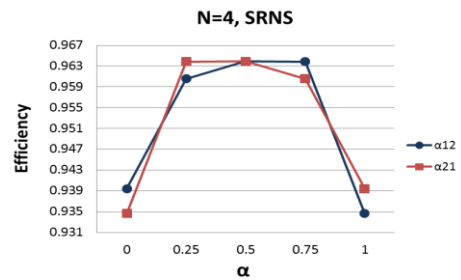


Fig. 5. The change of efficiency with  $\alpha$  for SRNS under  $N=4$ .

For  $\alpha_{12}=\alpha_{21}=0.5$ , the performance becomes as same as  $(\alpha_{12}=0.75, \alpha_{21}=0.25)$  in SRNS and gets a bit higher in HFB.

The surplus of WIP moderates the starvation of  $W_1$  and  $W_2$ , which results to minimize the differences between the opposite combinations.

In addition to the above mentioned, we have almost similar patterns for both rules. However, the workload configurations with extreme values of  $\alpha$  show a bit better efficiency in SRNS than that in HFB. With high level of information accuracy as in HFB, considering the number of subtasks at station 2 with ones in buffer helps sending no extra subtasks to downstream buffer whereas in SRNS the buffer will be full more, which can be noticed easily from a bit longer cycle time in SRNS. With the ample WIP, a line with SRNS experiences less starvation for station 2 and no negative effect on station 1. The effect of this behavior on the performance diminishes as the

workload configuration goes toward balanced case  $(\alpha_{12}=\alpha_{21}=0.5)$  where each worker has a relatively equal opportunity to starve.

The last case when  $N$  reaches to five. Each rule shows a different pattern of efficiency. In HFB (Fig. 6), the change of efficiency has almost the same pattern as in the previous case ( $N=4$ ) with higher efficiency. The higher level of information accuracy helps to have more robust pattern against increase of WIP. On the other side, Fig. 7 displays a pattern of SRNS that is opposite to the first case but the differences between the opposite combinations are a bit less. The efficiency gets better as  $\alpha_{12}$  gets bigger then it decreases after 0.5 while  $\alpha_{21}$  continues improving the efficiency till 0.75. The combination  $(\alpha_{12}=0, \alpha_{21}=1)$  presents higher efficiency than  $(\alpha_{12}=1, \alpha_{21}=0)$ . Also the combination  $(\alpha_{12}=0.25, \alpha_{21}=0.75)$  has better efficiency than  $(\alpha_{12}=0.75, \alpha_{21}=0.25)$  which is opposite to the case with  $N=3$ .  $(\alpha_{12}=0.25, \alpha_{21}=0.75)$  and  $\alpha_{12}=\alpha_{21}=0.5$  have the same highest efficiency.

We justify this trend in SRNS as follows. When  $\alpha_{12}$  is 1, that means the fixed task of station 1 gets so big (in our study 5), in other words it becomes a bottleneck. That will hinder inserting more jobs into the line. Besides that cause, having low information accuracy will result starvation of station 2, which leads to worsen the efficiency. On other hand, when  $\alpha_{12}=0$ , which means station 2 gets a bottleneck. Here, no station starves since there is an extra WIP. For  $\alpha_{12}=0.25$  and 0.75, the same analysis can be relied on and the performance improves as  $\alpha_{12}$  approaches 0 as the effect of bottleneck will diminish.

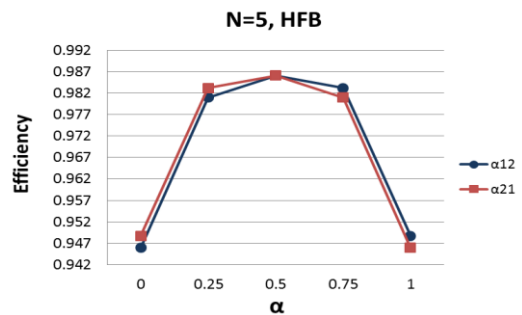


Fig. 6. The change of efficiency with  $\alpha$  for HFB under  $N=5$ .

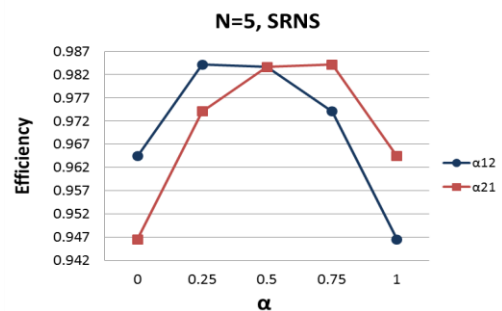


Fig. 7. The change of efficiency with  $\alpha$  for SRNS under  $N=5$ .

Another notable point, it is obvious from Fig. 7 that the differences between the opposite combinations are bigger than that in the previous case and in HFB for the same case. For example, for  $\alpha_{12}=1$  we have a big fixed task at station 1 that delays inserting more jobs and  $R$  decreases as the fixed task at station 2 shrinks. In this sense, station 2 starves longer thus the efficiency deteriorates. This matter is not existed in HFB and in the case with  $N=4$  where its effect is tiny since the

WIP is not so big.

#### IV. CONCLUSION

We simulated a two-station production line run under DLB policy and considered two levels of information accuracy represented by two threshold rules; HFB, SRNS. Several workload configurations were investigated. We proposed a measure for workload to find out the pattern of efficiency for different configurations of workload.

The simulation outcomes showed that the workload has an obvious effect on the efficiency and this effect has remarkable patterns with several values of  $\alpha$  under different levels of WIP. The information accuracy also affects the performance especially for high and low WIP and low information accuracy level generally presents better performance for extreme values of  $\alpha$ . On the other hand, higher level of information accuracy shows better performance as close as the fixed tasks approach to the balance state.

In addition, the high level of information accuracy displays low disparities between the opposite configurations of workload. For instance,  $(\alpha_{12}=0, \alpha_{21}=1)$ ,  $(\alpha_{12}=1, \alpha_{21}=0)$  have almost the same or so close performance in HFB comparing with that in SRNS. In other words, in case of exchanging the position of fixed tasks, high information accuracy shows more robust performance.

The results of this paper provide practical insights to the process designers and the people in work scheduling. These insights help to find the settings that achieve higher efficiency from work-sharing by manipulating the level of information accuracy, the distribution and size (if possible) of fixed workload.

#### REFERENCES

[1] S. Shingo, *Non-Stock Production: The Shingo System for Continuous Improvement*, Productive Press, 1985.

[2] J. Ostolaza, J. McClain, and J. Thomas, "The use of dynamic (state-dependent) assembly-line balancing to improve throughput," *Journal of Manufacturing and Operation Management*, vol. 3, pp. 105-133, 1990.

[3] H. Parvin, M. P. Van Oyen, D. G. Pandelis, D. P. Williams, and J. Lee, "Fixed task zone chaining: worker coordination and zone design for inexpensive cross-training in serial CONWIP lines," *IIE Transactions*, vol. 44, pp. 894-914, 2012.

[4] J. O. McClain, J. Thomas, and C. Sox, "On-The-Fly line balancing with very little WIP," *International Journal of Production Economics*, vol. 27, pp. 283-289, 1992.

[5] J. Chen and R. G. Askin, "Throughput maximization in serial production lines with worksharing," *International Journal of Production Economics*, vol. 99, pp. 88-101, 2006.

[6] R. G. Askin and J. Chen, "Dynamic task assignment for throughput maximization with worksharing," *European Journal of Operational Research*, vol. 168, pp. 853-869, 2006.

[7] E. S. Gel, W. J. Hopp, and M. P. V. Oyen, "Factors affecting opportunity of worksharing as a dynamic line balancing Mechanism," *IIE Transactions*, vol. 34, no.10, pp.847-863, 2002.

[8] V. I. Cesani and H. J. Steudel, "A study of labor assignment flexibility in cellular manufacturing systems," *Computers and Industrial Engineering*, vol. 48, pp. 571-591, 2005.

[9] J. A. C. Bokhorst, "The impact of the amount of work in process on the use of cross-training," *International Journal of Production Research*, vol. 49, no. 11, pp. 3171-3190, 2011.

[10] M. L. Spearman, D. L. Woodruff, and W. J. Hopp, "CONWIP: A pull alternative to Kanban," *Production Planning and Control*, vol. 28, No.5, pp. 879-894, 1990.

[11] A. Alan, B. Pritsker, and J. J. O'Reilly, "Simulation with visual SLAM and AweSim," 2nd ed. New York, U.S.: John Wiley & Sons, 1999.



**Salah Kasmó** was born in Aleppo, Syria. He received the B.Eng. degree in industrial engineering from University of Aleppo, Aleppo, Syria in 2005 and M.Eng. degree in industrial engineering and management from Tokyo Institute of Technology, Tokyo, Japan in 2011. He is currently a Ph.D. candidate in Department of Industrial Engineering and Management. His research interests include production system and operations management.



**Takao Enkawa** is currently a professor in the Department of Industrial Engineering and Management at Tokyo Institute of Technology, Tokyo, Japan. His research interests include quality control, product development, logistics, and production management.



**Sadami Suzuki** is currently an associate professor in the Department of Industrial Engineering and Management at Tokyo Institute of Technology, Tokyo, Japan. His research interests include supply chain management, theory of constraints, and production management.