Int J Adv Manuf Technol (2013) 69:937–959
DOI 10.1007/s00170-013-5027-5

ORIGINAL ARTICLE

Training and assignment of multi-skilled workers for implementing *seru* production systems

ChenGuang Liu · Na Yang · WenJuan Li · Jie Lian · Steve Evans · Yong Yin

Received: 7 December 2012 / Accepted: 22 April 2013 / Published online: 8 June 2013

Abstract Confronted with high variety and low volume market demands, many companies, especially the Japanese electronics manufacturing companies, have reconfigured their conveyor assembly lines and adopted *seru* production systems. *Seru* production system is a new type of work-cell-based manufacturing system. A lot of successful practices and experience show that *seru* production system can gain considerable flexibility of job shop and high efficiency of conveyor assembly line. In implementing *seru* production, the multi-skilled worker is the most important precondition, and some issues about multi-skilled workers are central and foremost. In this paper, we investigate the training and assignment problem of workers when a conveyor assembly line is entirely reconfigured into several *serus*. We formulate a mathematical model with multiple objectives which aim to minimize the total training cost and to balance the total processing times among multi-skilled workers in each *seru*. To obtain the satisfied task-to-worker training plan and worker-to-*seru* assignment plan, a three-stage heuristic algorithm with nine steps is developed to solve this mathematical model. Then, several computational cases are taken and computed by MATLAB programming. The computation and analysis results validate the performances of the proposed mathematical model and heuristic algorithm.

Keywords *Seru* production system · Training of multi-skilled workers · Multi-skilled workers assignment · Reconfiguration of production systems · Heuristic algorithm

C.-G. Liu (✉) · N. Yang · W.-J. Li · J. Lian · Y. Yin
School of Economics and Management, Xi'an University of Technology, Xi'an 710048, China
e-mail: liu@xaut.edu.cn

C.-G. Liu · S. Evans
Institute for Manufacturing, University of Cambridge, Cambridge CB3 0FS, UK

Y. Yin
Department of Economics and Business Management, Yamagata University,
Yamagata 990-8560, Japan

1 Introduction

Along with the transformation of consumption structure, nowadays, the market demand presents a growing trend towards high variety and low volume. Under such a market demand, the inherent weaknesses of conveyor assembly lines are gradually appearing. Various drawbacks, such as mass of work-in-process (WIP) and finished goods inventory, frequent over- or underproduction, poor effectiveness of single-skilled workers, and long response time to unexpected customer orders are reported in the literature [10, 15, 44]. These drawbacks mainly result from the inflexibility of conveyor assembly lines. In fact, flexibility is an important performance indicator to evaluate a production system [1]. In order to effectively and efficiently meet the diversified and fluctuant needs of customers, many kinds of flexible production systems are developed based on conveyor assembly lines.

Among various innovative production systems, *seru* production system has been attracting an increasing amount of attention from the academic and industrial communities [23, 38]. From 1992, many companies, especially the Japanese electronics manufacturing companies, have reconfigured their traditional conveyor assembly lines and adopted a type of work-cell-based manufacturing system. Such a work-cell-based manufacturing system, different from the traditional cellular manufacturing system, is known as the *seru* production system [24]. The differences between the *seru* production system and the traditional cellular manufacturing system are presented elsewhere [13, 23, 38].

Seru production system is an innovative mode of production system based on the conveyor assembly lines. It merges the flexibility of job shop and the efficiency of conveyor assembly line to some extent [34, 46]. In *seru* production, great benefits have been achieved by some Japanese manufacturing giants including Canon, Sony, Panasonic, Fujitsu, NEC, and Hitachi [14, 20, 28, 40, 45]. Among lots of successful cases, the implementation of *seru* production in Canon is the most prominent. During the period from 1999 to 2003, Canon reconfigured about 20,000 m of conveyor assembly lines in its 54 plants throughout the world and then adopted *seru* production systems. Relying on this reconfiguration of production systems, Canon has evolved into a high-performance organization [6]. Its S-class workers can assemble a product with 2,700 parts within only 2 h [19].

Serus, mainly being assembly cells, are the foundational physical units to form a *seru* production system. A *seru* consists of one or more multi-skilled workers and several simple equipment. There exist three basic types of *seru*: divisional *seru*, rotating *seru*, and *yatai* [23, 38]. When reconfiguring conveyor assembly lines, divisional *serus* are first formed. In a divisional *seru*, the workers also cooperate with one another as in the conveyor assembly line, but each worker operates more tasks than before. In a rotating *seru*, the multi-skilled workers, one by one in a fixed order, complete all tasks from one workstation to another. A *yatai* is a special rotating *seru* with a single worker. In a *yatai*, the worker has no need to run from one workstation to another because workstations and necessary tools are accessible. In *seru* production, a *seru* devotes to one or several specific product types, and some or all of the tasks are completed within a single *seru* [35]. To cope with the fluctuant market demand, we can adjust the number of workers in each *seru* or the number of *serus* in production system.

The multi-skilled worker is one of indispensable components to successfully implement *seru* production. Many researches indicated that the multi-skilled worker is the most important resource for implementing *seru* production. From this viewpoint, *seru* production has been regarded as a representative of human-centered production [23, 42]. In *seru* production, each multi-skilled worker is responsible for several or all of tasks of a product. Sometimes, these multi-skilled workers are also responsible for production and quality control functions. With enriched job and increased individual value, the multi-skilled workers in *seru* production usually show higher job satisfaction [23].

At conveyor assembly lines, the workers are usually single-skilled. To successfully implement *seru* production after reconfiguring conveyor assembly lines, training for multi-skilled workers is required to carry out timely. Considering the training cost and the capability to acquire skills of each worker (not every worker can perform all complicated tasks of one or several product types), it is not a reasonable method to train all skills for each worker. In fact, in some cases, there is no need for workers to master all the skills to produce a product. For example, in divisional *serus*, each worker performs only several tasks of a product. For a specific worker, if divisional *serus* are adopted, training should focus on extending the skill range based on his/her current specialized skill. If rotating *serus* or *yatais* are adopted, training needs to cover all skills required in the entire production process.

In *seru* production systems, workers are assigned to different *serus* to produce different products. Even if several workers are assigned to the same *seru* when divisional *serus* are adopted, these workers also perform different tasks. Since different workers are responsible for different tasks, the training skills for these workers should be varied from one another. Moreover, in making task-to-worker training plan, the balance of processing times among workers in each *seru* should be in a satisfied situation, so as to ensure the effective operation of production.

In this paper, we focus on training and assignment problem of multi-skilled workers, an important problem for implementing *seru* production systems. We will investigate how to obtain the task-to-worker training and worker-to-*seru* assignment plans when the differences of training cost and processing time of each task for different workers are considered simultaneously. The rest of the paper is organized as follows. Section 2 provides a review of the related literature. Based on the analysis of the features of *seru* production system, Section 3 builds a comprehensive mathematical model with multiple objectives. Section 4 is dedicated to the development of a heuristic algorithm to solve the proposed mathematical model. Section 5 tests stability and validity of the developed algorithm via several computational cases. Section 6 draws a conclusion and presents some further research topics along with this work.

2 Literature review

2.1 Cross-training and assignment of workers in manufacturing systems

Cross-training is an important and critical technique to obtain multi-skilled workers. Cross-training and assignment problems of workers in manufacturing systems have been extensively investigated in the literature. Many researches noted that cross-training increases labor flexibility to deal with fluctuating demands or supply of new labors [27, 41]. Cross-training encourages workers to share workload. Generally speaking, high workload imbalance requires more extensive cross-training to improve job performance [7]. Cross-training also increases the possibilities that workers boost their feeling of justice and equity. However, a high level of cross-training may lead to some disadvantages such as huge training cost, productivity loss owing to the shift of workers between machines, and too much time needed to learn and relearn new work [18, 37]. Cross-training strategy is a hot research topic in the area of cross-training of workers. The work due to Hopp et al. [11] is one of the representatives. They compared two cross-training strategies, straightforward capacity-balancing strategy and innovative overlapping zone strategy. As a conclusion, innovative overlapping zone strategy was regarded as a robust and efficient approach to realize workforce agility. Iravani et al. [12] created a WS-APL methodology to solve complex stochastic problem of designing effective workforce cross-training structures in call centers.

The worker assignment problem is a main part of the assignment problem, a classic problem in academic research. Worker assignment problems have been studied in many works of cellular manufacturing systems. Ertay and Ruan [9] used data envelopment analysis to make optimal worker allocation plans in manufacturing cells. Davis and Mabert [8] investigated order dispatching and labor assignment/reassignment decisions in two different cellular manufacturing settings of independent cells and linked cells. Kuo and Yang [21] formulated a mixed integer programming formulation for the mixed-skill multi-line worker allocation problem in labor-intensive manufacturing system. Süer and Tummaluri [39] formulated three models to assign workers to labor-intensive cells considering skill level, skill-based operation times, and worker learning and forgetting with no concern for cross-training. The first two models determined optimal cell configuration and cell loads. The last one completed the assignment of workers to operations, which gave a way of thinking for multi-skilled worker assignment problem in assembly cell environment.

Most studies on cross-training and assignment of workers in cellular manufacturing focus on the related cost. Based on the technical and human skills presented in Warner et al. [43], Norman et al. [29] proposed a profit function including productivity, quality costs, and training costs to solve worker assignment problem in manufacturing cells. Considering customer demand, skill depth, and job rotation, McDonald et al. [25]) developed a dynamic model for minimizing net present cost generated from the assignment of multi-skilled workers to tasks, which includes initial and incremental training costs, inventory costs, and quality costs. Satoglu and Suresh [33] proposed a goal-programming model for designing hybrid cellular manufacturing systems. In their approach for solving the proposed mathematical model, one phase was carried out to allocate cross-trained workforce from job shops to cells considering cross-training cost, hiring and firing cost, and over assignment of workers to more than one cell.

Some works paid attention to workload balance, or both aspects of cross-training cost and workload balance. Cesaní and Steudel [5] analyzed labor assignment flexibility to emphasize that workload balance and workload sharing among workers are important in evaluating the performance of cellular manufacturing system. Slomp et al. [37] developed an integer programming model considering worker training costs and workload balance among them to determine which worker should be trained for which machine in a manufacturing cell. To solve this mathematical model, worker training cost and workload balance were connected together by weights in their work.

Cross-training cost and workload balance are two key performance indicators of production systems. In most researches with regard to the design of production systems, usually only one of the two indicators is taken into account. Although several studies incorporated cross-training cost and workload balance indicators into their mathematical model, these two indicators are simply transformed into one in solving the mathematical models. Such an approach weakens the applicability in real-life production. In this paper, we will take both cross-training cost and workload balance into account and will compute the satisfied solution by developing an effective heuristic algorithm.

2.2 Multi-skilled workers in *seru* production systems

Seru production system is an innovation of production systems based on conveyor assembly lines. It generated in Japan 20 years ago and is extensively implemented by manufacturing companies in electronics industry. The detailed introduction of *seru* production system can be found in several works, such as those of Liu et al. [23, 24] and Stecke et al. [38]. In the literature on *seru* production system, many researchers highlighted the importance of multi-skilled workers in *seru* production. Isa and Tsuru [13]) compared the workforce agility between *serus* and manufacturing cells and indicated that the importance degrees of multi-skilled workers within them are different. By analyzing the human resource management practices in 20 Japanese plants that have adopted *seru* production, Sakikawa [31, 32] revealed that multi-skilled workers are the core element to successfully implement such a system. Shirai [36] also described that the implementation of *seru* production system relies heavily on multi-skilled workers.

Multi-skilled workers can bring great benefits in implementing *seru* production. Sakazume [30] summarized the influence of multi-skilled workers in *serus* by a case study on eight companies in Japan from three aspects. Firstly, multi-skilled workers could help the workers with lower abilities to share workload so as to gain workload balance within a *seru*. Secondly, multi-skilled workers in a *seru* can help reduce WIP inventory and improve product quality by highly efficient cooperation. Finally, when the production suffers variety and volume fluctuation, workers can be shifted among *serus* to balance production. The various benefits achieved in *seru* production, such as decreasing in lead time, capital investment, setup time, WIP, and finished-product inventory, required workforce and shop floor space, have been shown in many literatures [13, 14, 19, 20, 28, 40, 45].

While the importance and benefits of multi-skilled workers in *seru* production system is analyzed, the training problem of workers in *seru* production systems is also discussed in many works. Liu et al. [23] indicated that the cross-training for multi-skilled workers is not limited to technical tasks but also extends to managerial tasks in *seru* production, since multi-skilled workers in *seru* production systems undertake more responsibilities than technical jobs, for example, participating in production discussions and decision-making activities to improve efficiency of production system. Miyake [26] stressed that physical reconfiguring process from conveyor assembly lines to *serus* is simple, but cross-training of workers is a long-term, difficult, and costly process. Some companies perhaps lose the opportunities of implementing *seru* production system just because of vast worker training costs. Asgari and Yen [3] also indicated that *seru* production system is a continual learning organization. However, persistent learning to acquire abundant and fresh knowledge also increases the pressure on workers [28]. Liu et al. [22] emphasized that worker assignment matched with workers' abilities should be better designed to utilize their potential and improve productivity.

Recently, several researchers investigated the assignment problem of multi-skilled workers in implementing *seru* production system by quantitative study method. Kaku et al. [17] proposed a human-task-related performance evaluating model for converting conveyor assembly line to *seru* production system. In the process of assigning the workers, the positive and negative human-related factors were analyzed, such as the possible added tasks for each worker as a negative factor and improved skill range from the positive viewpoint. They also suggested that a human-factor-based training approach has an important influence on the performance of *seru* production system. Kaku et al. [16] established a bi-objective mathematical model to decide how many *serus* should be built and how many workers should be assigned to each *seru* in implementing *seru* production. Later, by taking the individual ability differences of multi-skilled workers into account, Liu et al. [22] investigated the reconfiguration problem from conveyor assembly line to *serus* aiming to minimize the makespan of the reconfigured *seru* production system.

Although these works involve assignment problem of multi-skilled workers in implementing *seru* production system, quantitative study on it is really sparse. These limited number of quantitative studies mostly focus on the number of workers in each *seru* rather than the matching between workers and tasks. In this work, we will consider the training cost and the balancing of processing times among multi-skilled workers in each *seru* simultaneously and use quantitative study method to obtain the training plan and assignment plan of multi-skilled workers in a *seru* production system.

3 Problem formulation

3.1 Problem description

In this study, we discussed a special case of reconfiguration from a conveyor assembly line to a *seru* production system. For some company, there exists a traditional conveyor assembly line which can produce several types of products. This conveyor assembly line includes several workstations. Each workstation corresponds to a specific task. At each workstation, there is a single-skilled worker for the corresponding task. There is a one-to-one correlation between a single-skilled worker and a specific task. When producing a specific type of products, some workers for some tasks may not be needed since this product type does not include the corresponding tasks.

When this conveyor assembly line is used to produce products, production pace is determined by the worker who has the slowest operation speed. Waiting time between operations due to the difference of workers' abilities appears. For this reason, those workers with higher abilities can not be utilized sufficiently both in physical and mental levels [2]. Moreover, once a worker is absent, the whole running production line will be interrupted. If different product types are produced at this conveyor assembly line, long setup time is required.

Confronted with the above internal situation and the fluctuant outside market demand, the company reconfigures the conveyor assembly line to a *seru* production system as shown in Fig. 1. To successfully implement the *seru* production system, single-skilled workers at the original conveyor assembly line are required to be cross-trained and become multi-skilled even full-skilled workers.

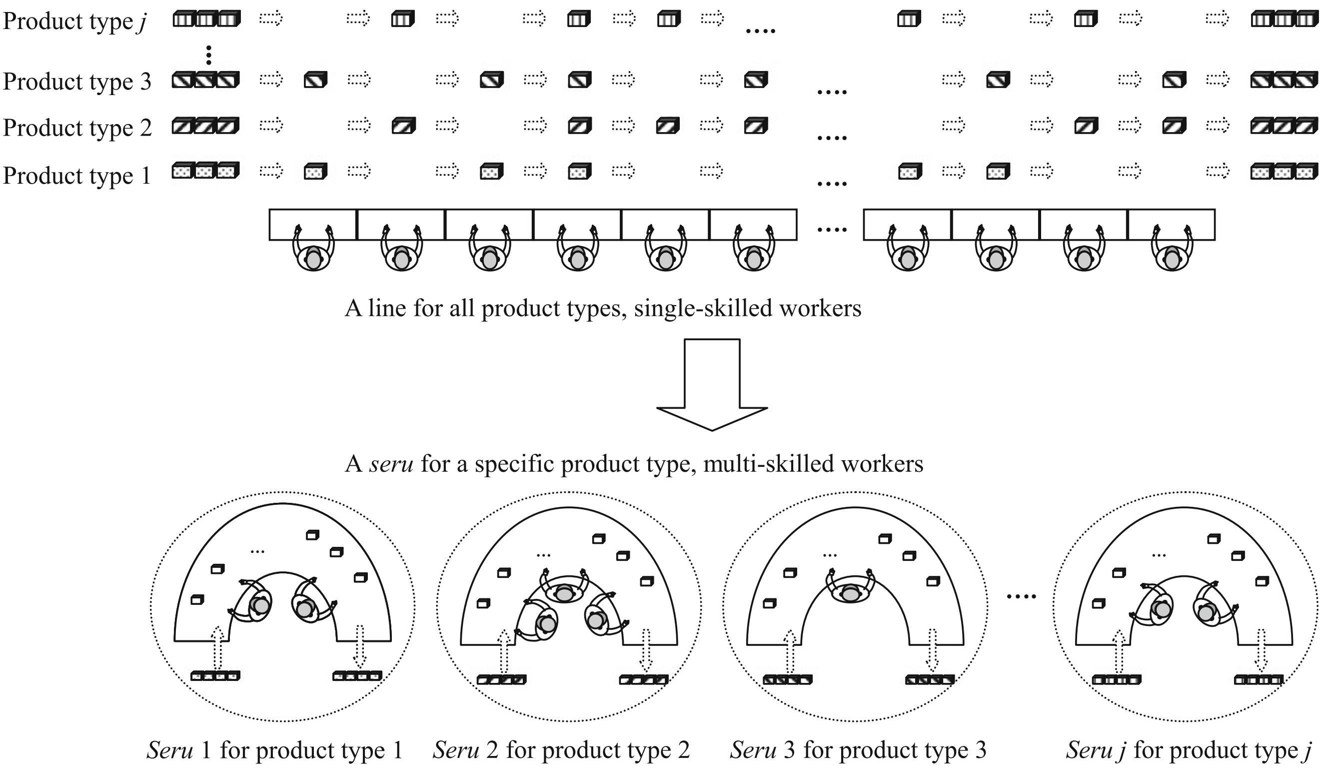

Fig. 1 Reconfiguration of a conveyor assembly line to *serus*

For the production system reconfigured from the conveyor assembly line, a pure *seru* system is considered in this paper. Moreover, each *seru* is specified to be a divisional *seru* or a *yatai*, which is responsible for a specific product type. All products belonging to the same type are completed in a specific *seru*. The number of *serus* is equal to the number of product types. All *serus* run independently without disturbing one another. Such a *seru* system is different from these investigated in Liu et al. [22] and Kaku et al. [17]. In their works, the hybrid system with assembly lines and *serus* are considered.

Regarding the above reconfiguration, our issue is to determine the satisfied or optimal worker-to-*seru* assignment plan and task-to-worker training plan with the objectives to minimize the total training cost and balance the processing times of workers in each *seru*. That is, we investigate how to train workers and assign workers to *serus* with some measurement performances.

3.2 Problem assumption

In real-life production cases, the training and assignment problem of workers described above is very complex if various manufacturing attributes are considered. In order to construct a mathematical model for such a problem, we make the following assumptions.

(1) A *seru* is only for producing one product type.

(2) There is no limit on the number of accepted skills for a specific worker. A worker is capable of processing multiple, even all, tasks of a product.

(3) For a specific task of one product type, the processing time is the same for all workers. That is, the processing time of each task is standard time.

(4) The number of product types is less than or equal to that of workers at the original conveyor assembly line. That is, each *seru* includes at least one worker.

(5) For the workers and tasks at the original conveyor assembly line, there is a one-to-one correlation between a single-skilled worker and a specific task.

(6) The number of workers is fixed before and after the reconfiguration of a production system.

3.3 Notation

- Model parameters

i the index of worker (i=*1, 2, ..., I*).

j the index of *seru* or product type (j=*1, 2, ..., J*).

k the index of task or workstation at the conveyor assembly line (k=*1, 2, ..., K*).

t_j^k the standard processing time of task k of product type j.

c_{ij}^k the cost for training worker i to master the skill for task k of product type j.

W_j the size of *seru* j regarding workers.

C the total training cost spent on all workers.
E_j the average processing time over all workers in *seru* j.
D_j the sum of squares of deviations from mean of processing times over all workers in *seru* j.

- Decision variables

$$\alpha_{ij}^k = \begin{cases} 1, & \text{if worker } i \text{ is assigned to } seru\ j \\ & \text{and processes task } k; \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_{ij} = \begin{cases} 1, & \text{if worker } i \text{ is assigned to } seru\ j; \\ 0, & \text{otherwise.} \end{cases}$$

3.4 Mathematical model

For product type j, if it does not include task k, we set $t_j^k = 0$. In this case, there is no need to train the worker to master the skill for task k, and we can think the training cost to master the skill for this task is infinity for any worker, that is, $c_{ij}^k = \infty$. Following the above notations, we know that E_j, the average processing time over all workers in *seru* j, can be obtained by

$$E_j = \frac{\sum_{k=1}^{K} t_j^k}{\sum_{i=1}^{I} \beta_{ij}}. \tag{1}$$

Considering the production situation and assumptions described above, we formulate the following mathematical model for our research problem.

Minimize

$$C = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} c_{ij}^k \cdot \alpha_{ij}^k \tag{2}$$

$$D_j = \sum_{i=1}^{I}\left[\left(E_j - \sum_{k=1}^{K} t_j^k \cdot \alpha_{ij}^k\right) \cdot \beta_{ij}\right]^2, \quad \forall j \tag{3}$$

subject to

$$\sum_{j=1}^{J} \beta_{ij} = 1, \qquad \forall_i \tag{4}$$

$$\sum_{k=1}^{K} \alpha_{ij}^k > 1, \qquad \forall_{i,j} \mid \beta_{ij} = 1 \tag{5}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} \alpha_{ij}^k = 1, \qquad \forall_k \tag{6}$$

$$1 \le \sum_{i=1}^{I} \beta_{ij} \le W_j, \qquad \forall_j \tag{7}$$

This is a multi-objective mathematical model to simultaneously achieve task-to-worker training plan and worker-to-*seru* assignment plan in reconfiguring a production system. Objective function (2) is to minimize the total training cost spent on all workers. Objective function (3) is to minimize the sum of squares of deviations from mean of processing times for all workers in each *seru*, which shows the workload balance among all workers in each *seru*. Regarding the parameter c_{ij}^k in objective functions, if worker i has already mastered the skill for processing task k of product type j, we set c_{ij}^k=0.

Constraint (4) manifests that a worker can only be assigned to one *seru*. That is, the shift of workers among *serus* is not allowed. Constraint (5) indicates that each worker in a *seru* can operate more than one task of a product. This reflects that workers in *serus* have multiple skills, which is different from that in the original conveyor assembly line. Constraint (6) ensures that a task in a specific *seru* is only processed by one worker. Work sharing among workers is not considered. Constraint (7) assures that there is at least one worker and at most W_j workers in *seru* j.

4 A heuristic algorithm

The problem presented by the above mathematical model is a multi-objective optimization problem. For such an issue, it requires to turn the concept of "optimum" into "best compromise" and reach trade-off among targets [4]. Based on the analysis of characteristics of the investigated problem, we develop a three-stage heuristic algorithm with nine steps to solve it.

In general, the proposed heuristic algorithm can be divided into three stages. The first stage is to get a worker-to-*seru* assignment plan. The second one is to obtain all feasible task-to-worker training plans. The third stage is to determine the final satisfied task-to-worker training plan. The specific approach of the proposed heuristic algorithm can be outlined as follows.

Stage 1: To get a worker-to-*seru* assignment plan.

- Step 1: For each worker, calculate the total training cost for acquiring all skills to process all tasks of each product type. That is, compute $\sum_{k=1}^{K} C_{ij}^k$ for any i, j.
- Step 2: For each worker, assign him/her to a *seru* associated with a specific product type on which he/she has the minimum total training cost. If there are more than one such *serus*, assign him/her to a *seru* which is associated with a product type with the minimum total processing time. If there still exist two or more such

serus, assign him/her to the *seru* with the smallest index number.

- Step 3: Form set S_s using the *serus* to which no workers are assigned as elements and form set S_w using the *serus* in which the number of workers exceeds the predetermined size as elements. Check whether $S_s = \phi$.

  - ⋆ Case 1: If $S_s = \phi$. Check whether $S_w = \phi$.
    - ∗ Sub-case 1: If $S_w = \phi$, end stage 1.
    - ∗ Sub-case 2: If $S_w \neq \phi$, considering the minimization of the increased total training cost, remove one or more workers from *serus* belonging to S_w to *serus* not belonging to S_w, to make all the *serus* satisfy the constraints in *seru* size. End stage 1.
  - ⋆ Case 2: If $S_s \neq \phi$. Check whether $S_w = \phi$.
    - ∗ Sub-case 1: If $S_w = \phi$, considering the minimization of the increased total training cost, remove one or more workers from *serus* not belonging to S_s to *serus* belonging to S_s, to make each *seru* has at least one worker. End stage 1.
    - ∗ Sub-case 2: If $S_w \neq \phi$, to make the number of workers in all *serus* satisfy the constraints in *serus* size and each *seru* has at least one worker, considering the minimization of the increased total training cost, remove one or more workers from *serus* belonging to S_w to *serus* belonging to S_s. If the number of the remaining workers in all *serus* belonging to S_w drop to the predetermined *seru* size but there still exist *serus* in which no workers are assigned, continuously transfer one or more workers from *serus* belonging to $S_w \bigcup S_s$ to them in considering the minimization of the increased total training cost, until the number of workers in each *seru* satisfies the constraints in *seru* size and each *seru* has at least one worker. End stage 1.

When stage 1 is finished, we can obtain the worker-to-*seru* assignment plan. In the following stage 2 and stage 3, the approaches for all *serus* are the same. Therefore, for each *seru*, we have the following steps.

Stage 2: To obtain all feasible task-to-worker training plans.

- Step 4: According to the training costs on each task for different workers in the *seru*, assign each task to the worker who has the minimum training cost on it. If the minimum training cost corresponds to two or more workers, assign the task to one worker with the smallest index number. Considering the task-to-worker assignment plan, we get an initial feasible training plan.
- Step 5: According to the feasible training plan under investigation, compute the total processing time of each worker. Locate the worker ($W^{\max}$), who has the maximum total processing time, and the worker ($W^{\min}$), who has the minimum total processing time. If there are several corresponding workers, select the worker with the smallest index number. Then, for these two workers, compute the difference between the maximum total processing time and the minimum total processing time, d^t.
- Step 6: Considering all tasks assigned to worker $W^{\max}$, compute the difference of training costs between worker $W^{\max}$ and worker $W^{\min}$. Locate the task with the minimum difference of training cost. If there are more than one tasks that correspond to the minimum difference of training cost, locate the task with the minimum difference of training cost and the minimum processing time. If there are still more than one task, locate one randomly. This task is denoted by O^s.
- Step 7: Check whether d^t is larger than the processing time of task O^s.
  - ⋆ Case 1: If d^t is larger than the processing time of task O^s, remove task O^s from worker $W^{\max}$ to worker $W^{\min}$. Form a new feasible training plan. Then go to step 5.
  - ⋆ Case 2: If d^t is not larger than the processing time of task O^s, go to step 8.
- Step 8: Collect all feasible training plans. All these training plans are all non-inferior task-to-worker training plans.

Stage 3: To determine the satisfied task-to-worker training plan(s).

- Step 9: According to some measurement performance, determine one or more satisfied task-to-worker training plans from the feasible training plans following a specific rule. End.

For the above approaches, we make further explanation from two points. The first one is about the method to remove workers used in stage 1. A specific approach can be presented as follows:

- Step I: Using the workers in *serus* out of which some workers should be removed as elements, form a set W_O. Moreover, S_O denotes the *serus* out of which some workers should be removed, and S_E denotes the *serus* to which some workers would be removed.

- Step II: For any worker i belonging to W_O and any *seru* j belonging to S_E, compute $d_{ij} = \sum_{k=1}^{K} C_{ij}^{k} - \min_i \left\{ \sum_{k=1}^{K} C_{ij}^{k} \right\}$. Then, locate the minimum d_{ij} and remove the corresponding worker i from the original *seru* to the corresponding *seru* j.
- Step III: After removing, check whether each *seru* satisfies the size constrain and has at least one worker.

  ⋆ Case one. If each *seru* satisfies the size constraint and has at least one worker, end.
  ⋆ Case two. If there is a *seru* that can not satisfy the size constrain or has no worker assigned to it, go to step I.

The second one is the method to select out a satisfied training plan from the feasible training plans. In fact, once a new feasible training plan is obtained following the approaches of stage 2, compared to the previous training plan, the value of objective function (3) decreases, but the value of objective function (2) may increase. Generally, there are two frequently used rules to evaluate these feasible plans and select one or more satisfied plans. One is to set weight coefficients to the value of objective function (2) and the value of objective function (3) according to the managers' preference, then compute and select the satisfied plan with the minimum weighted value. Another one is to compare the increasing rate of the value of objective function (3) and the decreasing rate of the value of objective function (2), and then select the satisfied plan(s) with less increasing rate and larger decreasing rate.

The flow-process of the proposed heuristic algorithm is presented in Fig. 2.

5 Computational performance

5.1 A simple computational case

In this computational case, there is a conveyor assembly line with eight workstations. Correspondingly, there are eight single-skilled workers. There are four types of products; each type of products has six tasks. The conveyor assembly line is entirely reconfigured to four *serus*, and each *seru* is dedicated to a specific product type. The size of each *seru* regarding workers is two. The standard processing time of each task of each product type and the training cost for each worker are presented in Tables 1 and 2, respectively.

In Table 1, the sign of "/" means the task does not exist in the corresponding product type. In Table 2, the number of "0" means that the worker has already mastered

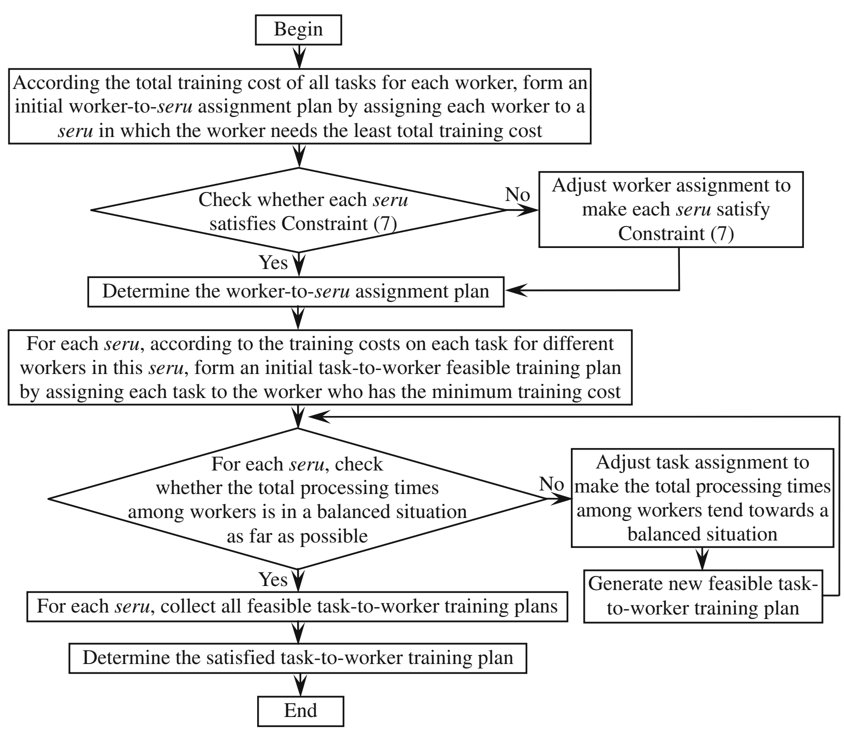

Fig. 2 The flow-process diagram of the proposed heuristic algorithm

Table 1 The standard processing time of each task of each product type

Product type	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
1	4.03	/	4.35	3.95	4.07	4.19	4.02	/
2	/	3.97	3.26	/	4.19	4.16	4.54	4.33
3	3.81	3.92	/	/	4.23	3.86	3.77	4.37
4	3.71	3.36	3.96	3.91	/	4.43	/	4.27

Table 2 The training cost of each task of each product type for each worker

	Worker	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Product type 1									
	1	0	/	88	128	86	168	175	/
	2	143	/	156	137	177	119	147	/
	3	111	/	0	175	173	107	91	/
	4	133	/	159	0	59	130	171	/
	5	99	/	141	150	0	126	139	/
	6	99	/	66	65	123	0	121	/
	7	59	/	125	158	114	59	0	/
	8	71	/	71	130	129	167	141	/
Product type 2									
	1	/	187	182	/	169	153	119	162
	2	/	0	197	/	113	142	135	166
	3	/	91	0	/	74	101	127	161
	4	/	88	167	/	100	133	86	163
	5	/	91	157	/	0	197	157	128
	6	/	174	172	/	127	0	77	119
	7	/	82	199	/	168	70	0	52
	8	/	197	138	/	111	92	152	0
Product type 3									
	1	0	119	/	/	121	137	56	69
	2	132	0	/	/	168	51	196	157
	3	121	60	/	/	87	186	106	107
	4	99	131	/	/	178	190	93	194
	5	165	111	/	/	0	80	129	130
	6	176	140	/	/	111	0	52	136
	7	136	168	/	/	143	145	0	81
	8	71	166	/	/	188	50	173	0
Product type 4									
	1	0	62	128	90	/	65	/	127
	2	171	0	120	53	/	100	/	182
	3	130	103	0	99	/	176	/	148
	4	65	132	190	0	/	126	/	149
	5	84	122	58	73	/	59	/	140
	6	60	170	63	137	/	0	/	86
	7	165	167	56	149	/	135	/	148
	8	150	162	183	142	/	50	/	0

the skill for the corresponding task, and the sign of "/" means that there is no need to train such a skill for the corresponding task since this task for this product type does not exist. The process routings of four types of products are 1-3-4-5-6-7, 2-3-5-6-7-8, 1-2-5-6-7-8, and 1-2-3-4-6-8 respectively, where the numbers denote the indexes of tasks. This computational case has been solved by MATLAB programming. In order to understand the proposed algorithm, the specific computation approach is presented as follows.

(1) According to Table 2, for each worker, compute the total training cost for acquiring all skills to process all tasks of each product type. The total training cost of each product type for each worker is shown in Table 3.
(2) According to Table 3, assign each worker to a *seru* associated with a specific product type on which he/she has the minimum total training cost. The worker-to-*seru* assignment plan is shown in Table 4. In Table 4, the number of "1" means that the corresponding assignment exists, "0" means that the corresponding assignment does not exist.
(3) In Table 4, we can see that the number of workers in *serus* 1 and 4 exceeds the predetermined *seru* size 2. Hence, we have $S_s = \phi$ and $S_w = \{1, 4\}$, where 1 and 4 mean *seru* 1 and *seru* 4, respectively. Now, we remove some workers from *serus* 1 and 4 to *serus* 2 and 3.

According to the worker-to-*seru* assignment plan in Table 4, we have $W_O = \{1, 2, 4, 5, 6, 7\}$, $S_O = \{1, 4\}$, and $S_E = \{2, 3\}$. For any worker i belonging to W_O and any *seru* j belonging to S_E, the differences of total training cost for worker i between two different *serus* in S_O and S_E, d_{ij}, can be obtained by $d_{ij} = \sum_{k=1}^{K} C_{ij}^{k} - \min_{i} \left\{ \sum_{k=1}^{K} C_{ij}^{k} \right\}$ and shown in Table 5.

In Table 5, we can see that the number "30" is minimum. Therefore, worker 1 is removed from *seru* 4 to *seru* 3. After updating W_O, S_O, and S_E, in the same way, remove worker 7 from *seru* 1 to *seru* 2. Up to now, each *seru* satisfies the constrains, and the adjusted worker-to-*seru* assignment plan is formed and shown in Table 6.
(4) Based on the above worker-to-*seru* assignment plan, for each *seru*, according to the processing time, assign each task to the worker who has the least processing time on it. The initial task-to-worker training plan is shown in Table 7.
(5) According to Table 7, for each *seru*, compute the total processing times that happened in each worker, locate the worker who has the maximum total processing time and the worker who has the minimum total processing time, and then compute the difference of the total processing time between the maximum and the minimum total processing times. For example,

Table 3 The total training cost on each product type (*seru*) for each worker

Worker	Product type 1 (*seru* 1)	Product type 2 (*seru* 2)	Product type 3 (*seru* 3)	Product type 4 (*seru* 4)
1	645	972	502	472
2	879	753	704	626
3	657	554	667	656
4	652	737	885	662
5	655	730	615	536
6	474	669	615	516
7	515	571	673	820
8	709	690	648	687

Table 4 The worker-to-*seru* assignment plan

Worker	*seru* 1	*seru* 2	*seru* 3	*seru* 4
1	0	0	0	1
2	0	0	0	1
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1
6	1	0	0	0
7	1	0	0	0
8	0	0	1	0

Table 5 The differences of total training cost between S_O and S_E for worker i

Worker	*seru* 2	*seru* 3
1	500	30
2	127	78
4	85	233
5	194	79
6	195	141
7	56	158

Table 6 The adjusted worker-to-*seru* assignment plan

Worker	*seru* 1	*seru* 2	*seru* 3	*seru* 4
1	0	0	1	0
2	0	0	0	1
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1
6	1	0	0	0
7	0	1	0	0
8	0	0	1	0

for *seru* 1, worker 6 ($W^{\max}$) has the maximum total processing time 16.59 and worker 4 ($W^{\min}$) has the minimum total processing time 8.02. The difference (d^t) is 8.57.

(6) For each *seru*, considering each task assigned to worker $W^{\max}$, compute the difference of training cost between $W^{\max}$ and $W^{\min}$. For example, for *seru* 1, the differences of training cost between worker 6 and worker 4 for all tasks are shown in Table 8.

In Table 8, we can see that task 1 (O^s) has the minimum difference of training cost, and the corresponding processing time of task 1 is 4.03.

(7) Since d^t is 8.57 which is larger than 4.03, then we remove task 1 from worker 6 to worker 4. A new feasible task-to-worker training plan is formed. In the same way, form all possible feasible task-to-worker training plans.

(8) For each *seru*, collect all feasible task-to-worker training plans which are presented in Table 9.

Table 8 The differences of training cost between worker 6 and worker 4

Task	Processing time	Difference of training cost
1	4.03	34
7	4.02	50
3	4.35	93
6	4.19	130

In Table 9, T_c represents the total training cost for all workers in the corresponding *seru*, and D_j represents the sum of squares of deviations from mean of processing times among the workers in the corresponding *seru*.

(9) Following two selecting rules, we determine the satisfied task-to-worker training plans.

The satisfied task-to-worker training plans by the weight method. For each *seru*, supposing that the weights of the total training cost and the sum of squares of deviations from mean of processing times among all workers are given as 0.7 and 0.3, respectively. Take *seru* 1 as an example, the weight sum value is 252.52 in the first task-to-worker training plan, and the weight sum value is 265.33 in the second task-to-worker training plan. Hence, comparatively speaking, the first task-to-worker training plan is better and can be chosen as the satisfied one. In such an assumption of weights, the satisfied task-to-worker training plan is presented in Table 10.

Table 7 The initial task-to-worker training plan

	Worker	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Seru 1									
	4	0	0	0	1	1	0	0	0
	6	1	0	1	0	0	1	1	0
Seru 2									
	3	0	0	1	0	1	0	0	0
	7	0	1	0	0	0	1	1	1
Seru 3									
	1	1	1	0	0	1	0	1	0
	8	0	0	0	0	0	1	0	1
Seru 4									
	2	0	1	0	1	0	0	0	0
	5	1	0	1	0	0	1	0	1

Table 9 All feasible task-to-worker training plans

		Worker	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	T_c	D_j
Seru 1												
	Plan 1										345	36.723
		4	0	0	0	1	1	0	0	0		
		6	1	0	1	0	0	1	1	0		
	Plan 2										379	0.130
		4	1	0	0	1	1	0	0	0		
		6	0	0	1	0	0	1	1	0		
Seru 2												
	Plan 1										278	45.601
		3	0	0	1	0	1	0	0	0		
		7	0	1	0	0	0	1	1	1		
	Plan 2										287	1.296
		3	0	1	1	0	1	0	0	0		
		7	0	0	0	0	0	1	1	1		
Seru 3												
	Plan 1										346	28.125
		1	1	1	0	0	1	0	1	0		
		8	0	0	0	0	0	1	0	1		
	Plan 2										393	0.058
		1	1	0	0	0	1	0	1	0		
		8	0	1	0	0	0	1	0	1		
Seru 4												
	Plan 1										394	41.405
		2	0	1	0	1	0	0	0	0		
		5	1	0	1	0	0	1	0	1		
	Plan 2										435	0.029
		2	0	1	0	1	0	1	0	0		
		5	1	0	1	0	0	0	0	1		

The satisfied task-to-worker training plans by the change rate method. For each *seru*, compare the increasing rates of total training cost and the decreasing rate of sum of squares of deviations from mean of processing times between two task-to-worker training plans. For example, for *seru* 1, the two indicators are 0.0986 and 0.9965, respectively. The increasing rate of total training cost is much smaller than the decreasing rate of sum of squares of deviations from mean of processing times. Hence, comparatively speaking, it is better to select the second task-to-worker training plan as the satisfied one. Following such a rule, the satisfied task-to-worker training plan is presented in Table 11.

For this computational case, at the Pentium IV-based IBM PC, programming coded in MATLAB R2011b, the CPU time spent on the above computation is 2.843 s.

5.2 Several computation cases with large-scale data

To further verify the efficiency of the proposed model and heuristic algorithm, several more complex computational cases are selected. For the next computational case, there are 20 workers at the original conveyor assembly line, with eight product types. Each product type has 12 tasks. The *seru* size regarding workers is set as 3. The known standard processing time of each task of each product type is shown in Table 12, and the training costs of each task for each worker is presented in Table 13. In order to simplify

Table 10 The satisfied task-to-worker training plan by the weight method

	Worker	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Seru 1									
	4	0	0	0	1	1	0	0	0
	6	1	0	1	0	0	1	1	0
Seru 2									
	3	0	1	1	0	1	0	0	0
	7	0	0	0	0	0	1	1	1
Seru 3									
	1	1	1	0	0	1	0	1	0
	8	0	0	0	0	0	1	0	1
Seru 4									
	2	0	1	0	1	0	0	0	0
	5	1	0	1	0	0	1	0	1

the page layout, "Task*k*" is used to present "Task *k*" in the following tables.

This problem is solved by our proposed algorithm, the worker-to-*seru* assignment plan, and all feasible task-to-worker training plans are shown in Table 14. At the Pentium IV-based IBM PC, with a programming coded in MATLAB R2011b, it spends CPU time of 7.328 s to get the above results.

To further test the computational performance of the proposed algorithm, two more large-sized problems were solved (at Pentium IV-based IBM PC, and programming coded in MATLAB R2011b) to check the CPU time. One problem includes 30 workers and 13 product types, and each product type has 23 tasks. We obtained the worker-to-*seru* assignment plan, and all feasible task-to-worker training plans within 19.109 s. Another problem includes 40 workers and 14 product types, and each product type has 31 tasks. It spent 43.712 s to get the worker-to-*seru* assignment plan and all feasible task-to-worker training plans. These computational performances show that our proposed mathematical model and heuristic algorithm are effective.

Table 11 The satisfied task-to-worker training plan by the method of change rate

	Worker	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Seru 1									
	4	1	0	0	1	1	0	0	0
	6	0	0	1	0	0	1	1	0
Seru 2									
	3	0	1	1	0	1	0	0	0
	7	0	0	0	0	0	1	1	1
Seru 3									
	1	1	0	0	0	1	0	1	0
	8	0	1	0	0	0	1	0	1
Seru 4									
	2	0	1	0	1	0	0	0	0
	5	1	0	1	0	0	1	0	1

Table 12 The standard processing time of each task of each product

Product	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
1	0	4.31	0	0	0	3.98	0	4.14	0	4.35	3.53	4.34	3.9	0	0	4.26	3.89	4.43	3.73	4.12
2	4.23	0	3.66	0	4.13	3.85	0	4.19	0	3.68	3.89	0	3.93	0	4.33	3.76	0	4.27	4.22	0
3	0	3.61	0	4.42	4.13	3.4	4.07	0	3.89	4.03	0	3.88	4.43	0	4.08	0	0	0	3.96	3.85
4	3.64	3.57	0	0	0	3.85	0	3.87	3.84	4.03	4.36	4.5	3.64	0	3.82	3.8	0	4.02	0	0
5	0	4.17	0	4.02	4.05	3.86	0	4.28	3.24	3.91	3.73	0	4.03	4.04	0	0	0	0	3.69	3.93
6	3.75	0	3.69	4.22	3.89	0	0	4.17	0	4.43	3.79	4.06	0	3.97	0	3.87	3.75	0	3.88	0
7	3.65	4.36	4.01	3.81	0	0	4	3.72	0	3.74	4.38	0	0	4.32	3.82	0	4.29	0	3.67	0
8	0	3.65	0	3.91	3.68	4.08	3.99	0	4.25	4.05	4.01	3.87	0	0	0	3.83	0	4.45	4.38	0

6 Conclusions and future work

Seru production is a new production mode developed in Japan in recent several years. *Seru* production system is an innovation result of conveyor assembly line to obtain flexibility. Such an advanced production system can merge the flexibility of job shop and the efficiency of mass production, as flexible manufacturing system has done for machining. It has achieved great successes in Japanese electronics manufacturing industry. In this area, academic research is just beginning and far behind its applications in practice. For the design of *seru* production systems, lots of issues have not been solved at the moment.

For one of the important problems in designing *seru* production systems, cross-training and assignment problem of multi-skilled workers, an quantitative investigation was done in this paper. Considering the total training cost and the balance of processing times among workers in each *seru*, we formulated a comprehensive mathematical model. Since this model is NP-hard, we developed a heuristic algorithm to solve it. The computation of several numerical examples through MATLAB programming verifies the effectiveness of the proposed model and algorithm.

Although many real-life manufacturing attributes were incorporated into the mathematical model in this work, the proposed mathematical model is still limited in practice. For example, batch size, the number of identical *serus*, the learning and forgetting effect of workers, and some other manufacturing attributes were not taken into consideration. To make the research more accessible and relevant to actual implementation, it is a research topic for future to incorporate more manufacturing attributes into mathematical model. Moreover, in this paper, only a static and single-period mathematical model was proposed. However in practice, not only the type and level of skills for workers but also the market demand are dynamic. It is needed to extend the static and single-period mathematical model to a dynamic and multi-period model. We suggest this as another future research topic along with this work.

From the viewpoint of algorithm design, a heuristic algorithm was developed in this paper. For the computational performance, we say it is effective only based on the used CPU time. Since the same problem has not been investigated in the literature, we can not make a comparison of computational performance between our proposed algorithm and others. We think the proposed mathematical model could be solved by other kinds of algorithms, such as ant colony algorithm and genetic algorithm. We hope the effectiveness and efficiency of our developed algorithm can be further investigated by comparing with other algorithms in future works.

Table 13 The training cost of each task of each product type for each worker

Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
Product 1																				
1	/	65	/	/	/	194	/	111	/	71	87	57	55	/	/	53	93	101	133	104
2	/	0	/	/	/	96	/	171	/	88	193	58	156	/	/	120	95	163	103	166
3	/	80	/	/	/	127	/	198	/	85	51	189	65	/	/	152	171	159	63	70
4	/	144	/	/	/	157	/	77	/	153	129	170	171	/	/	180	67	59	164	161
5	/	189	/	/	/	151	/	72	/	169	151	160	138	/	/	107	80	81	100	99
6	/	170	/	/	/	0	/	118	/	88	94	170	168	/	/	192	143	158	195	105
7	/	134	/	/	/	179	/	155	/	106	99	63	197	/	/	111	200	184	172	186
8	/	156	/	/	/	108	/	0	/	150	149	89	165	/	/	50	152	189	56	128
9	/	193	/	/	/	122	/	173	/	178	150	189	118	/	/	51	131	143	124	137
10	/	190	/	/	/	136	/	60	/	0	112	105	115	/	/	161	129	185	141	128
11	/	138	/	/	/	134	/	85	/	138	0	171	195	/	/	157	182	78	52	78
12	/	150	/	/	/	132	/	131	/	78	178	0	88	/	/	195	159	193	136	191
13	/	75	/	/	/	185	/	69	/	81	152	173	0	/	/	104	173	97	199	166
14	/	74	/	/	/	107	/	187	/	136	52	154	155	/	/	183	93	85	56	187
15	/	191	/	/	/	192	/	144	/	196	121	73	93	/	/	154	60	62	116	90
16	/	172	/	/	/	190	/	76	/	72	61	52	158	/	/	0	182	54	95	75
17	/	133	/	/	/	84	/	185	/	175	192	88	130	/	/	139	0	73	107	165
18	/	176	/	/	/	68	/	56	/	176	177	131	83	/	/	165	76	0	192	139
19	/	140	/	/	/	118	/	184	/	55	168	146	166	/	/	193	151	123	0	122
20	/	151	/	/	/	169	/	196	/	160	112	84	166	/	/	136	98	111	70	0
Product 2																				
1	0	/	65	/	175	79	/	155	/	/	139	101	/	191	/	73	77	/	192	90
2	80	/	138	/	199	92	/	183	/	/	180	65	/	61	/	159	132	/	197	139
3	188	/	0	/	55	87	/	135	/	/	69	184	/	173	/	79	120	/	102	171
4	108	/	168	/	58	127	/	115	/	/	81	140	/	197	/	171	184	/	55	78
5	131	/	172	/	0	134	/	121	/	/	170	64	/	110	/	113	134	/	170	115
6	164	/	143	/	198	0	/	190	/	/	100	141	/	158	/	155	54	/	190	137
7	172	/	65	/	114	177	/	108	/	/	83	84	/	154	/	183	196	/	65	126
8	190	/	184	/	147	84	/	0	/	/	76	53	/	176	/	97	88	/	71	156
9	58	/	192	/	185	164	/	70	/	/	184	58	/	200	/	101	162	/	56	128
10	172	/	110	/	195	151	/	94	/	/	192	200	/	133	/	175	137	/	177	119
11	83	/	69	/	67	162	/	183	/	/	0	97	/	175	/	177	143	/	181	160
12	142	/	149	/	169	119	/	80	/	/	129	0	/	128	/	154	195	/	190	159
13	192	/	173	/	122	126	/	88	/	/	124	188	/	130	/	54	93	/	178	124

Table 13 (continued)

Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
Product 2																				
14	182	/	172	/	167	142	/	105	/	/	195	125	/	0	/	66	148	/	71	199
15	158	/	61	/	99	58	/	56	/	/	178	165	/	116	/	186	177	/	193	75
16	168	/	145	/	113	164	/	80	/	/	67	140	/	57	/	0	148	/	96	135
17	75	/	102	/	55	117	/	141	/	/	142	73	/	164	/	112	0	/	148	150
18	117	/	145	/	79	147	/	117	/	/	104	92	/	170	/	195	99	/	161	119
19	115	/	81	/	82	64	/	180	/	/	121	172	/	199	/	125	85	/	0	146
20	107	/	101	/	183	94	/	127	/	/	130	125	/	69	/	114	68	/	160	0
Product 3																				
1	/	153	/	180	65	162	60	/	190	140	/	143	60	/	174	/	/	/	79	93
2	/	0	/	75	195	162	112	/	95	151	/	58	56	/	114	/	/	/	78	86
3	/	186	/	67	196	57	192	/	73	89	/	181	106	/	78	/	/	/	156	164
4	/	199	/	0	72	123	52	/	121	88	/	74	100	/	59	/	/	/	106	70
5	/	62	/	161	0	164	175	/	196	167	/	82	140	/	196	/	/	/	154	84
6	/	160	/	140	52	0	76	/	69	81	/	159	79	/	198	/	/	/	168	119
7	/	116	/	82	197	137	0	/	171	122	/	81	193	/	131	/	/	/	179	131
8	/	83	/	192	178	149	92	/	185	88	/	79	121	/	65	/	/	/	79	129
9	/	187	/	167	61	143	79	/	0	146	/	122	53	/	193	/	/	/	197	132
10	/	108	/	177	128	127	85	/	124	0	/	52	137	/	116	/	/	/	158	145
11	/	156	/	75	186	119	164	/	92	78	/	132	106	/	109	/	/	/	130	50
12	/	148	/	88	124	147	171	/	122	180	/	0	68	/	69	/	/	/	67	98
13	/	110	/	179	142	181	72	/	92	190	/	85	0	/	146	/	/	/	138	72
14	/	184	/	106	57	123	58	/	101	84	/	58	153	/	117	/	/	/	73	86
15	/	159	/	60	177	75	141	/	195	121	/	157	140	/	0	/	/	/	152	176
16	/	50	/	192	127	109	181	/	125	57	/	121	167	/	54	/	/	/	67	142
17	/	199	/	83	74	67	137	/	68	158	/	79	149	/	141	/	/	/	188	110
18	/	68	/	65	154	110	161	/	136	161	/	55	175	/	72	/	/	/	111	109
19	/	60	/	76	117	134	190	/	160	57	/	84	117	/	79	/	/	/	0	51
20	/	119	/	174	113	144	121	/	158	87	/	130	105	/	81	/	/	/	102	0
Product 4																				
1	0	56	/	/	/	140	/	136	114	/	120	141	169	80	/	130	151	/	98	/
2	190	0	/	/	/	98	/	156	67	/	104	177	92	83	/	91	79	/	54	/
3	145	125	/	/	/	119	/	66	51	/	169	193	158	153	/	128	76	/	177	/
4	73	98	/	/	/	119	/	96	129	/	83	171	93	60	/	82	60	/	96	/
5	138	111	/	/	/	183	/	144	63	/	109	180	188	63	/	172	106	/	87	/

Table 13 (continued)

Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
Product 4																				
6	199	176	/	/	/	0	/	57	60	/	88	151	108	165	/	181	130	/	194	/
7	119	144	/	/	/	94	/	109	188	/	102	159	102	81	/	120	93	/	170	/
8	118	142	/	/	/	199	/	0	55	/	139	88	140	56	/	139	59	/	187	/
9	158	62	/	/	/	114	/	59	0	/	156	71	191	50	/	88	111	/	107	/
10	55	130	/	/	/	149	/	77	178	/	198	145	154	110	/	94	53	/	145	/
11	195	124	/	/	/	158	/	141	119	/	0	194	121	57	/	165	105	/	104	/
12	81	156	/	/	/	198	/	111	83	/	148	0	71	116	/	180	116	/	146	/
13	110	171	/	/	/	85	/	200	129	/	161	199	0	144	/	75	160	/	160	/
14	56	143	/	/	/	85	/	146	93	/	100	109	64	0	/	90	176	/	68	/
15	131	81	/	/	/	88	/	110	130	/	175	93	144	116	/	133	185	/	88	/
16	183	196	/	/	/	116	/	177	57	/	107	167	143	106	/	0	80	/	151	/
17	199	160	/	/	/	122	/	58	191	/	118	56	102	185	/	189	0	/	170	/
18	88	75	/	/	/	108	/	65	164	/	55	173	188	58	/	88	59	/	84	/
19	101	137	/	/	/	182	/	162	73	/	163	76	170	72	/	176	167	/	0	/
20	164	104	/	/	/	89	/	107	139	/	166	197	168	187	/	75	162	/	200	/
Product 5																				
1	/	199	/	164	98	69	/	192	82	80	144	/	54	138	/	/	/	/	150	60
2	/	0	/	80	159	190	/	51	98	52	179	/	65	181	/	/	/	/	148	157
3	/	164	/	176	149	154	/	130	130	102	166	/	143	67	/	/	/	/	173	177
4	/	59	/	0	105	176	/	154	76	107	191	/	127	154	/	/	/	/	131	96
5	/	107	/	195	0	133	/	74	176	180	187	/	114	192	/	/	/	/	72	99
6	/	79	/	115	113	0	/	82	173	68	171	/	87	140	/	/	/	/	63	70
7	/	173	/	108	92	128	/	85	81	148	60	/	113	61	/	/	/	/	70	134
8	/	67	/	60	62	159	/	0	118	169	89	/	193	134	/	/	/	/	94	90
9	/	189	/	111	199	77	/	130	0	83	54	/	151	101	/	/	/	/	183	177
10	/	186	/	76	137	86	/	140	55	0	73	/	172	74	/	/	/	/	104	95
11	/	113	/	137	122	128	/	120	76	178	0	/	114	74	/	/	/	/	99	78
12	/	57	/	64	159	82	/	75	140	126	122	/	147	74	/	/	/	/	116	59
13	/	134	/	123	65	64	/	186	152	104	175	/	0	177	/	/	/	/	127	94
14	/	174	/	191	136	87	/	154	168	121	147	/	83	0	/	/	/	/	131	52
15	/	111	/	75	119	180	/	153	50	124	165	/	176	135	/	/	/	/	59	94
16	/	121	/	107	77	198	/	93	107	200	177	/	148	142	/	/	/	/	122	82
17	/	158	/	76	89	151	/	160	172	193	119	/	133	187	/	/	/	/	161	54
18	/	177	/	196	132	112	/	94	61	172	136	/	192	163	/	/	/	/	61	193

Table 13 (continued)

Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
Product 5																				
19	/	119	/	89	86	95	/	165	196	54	124	/	194	71	/	/	/	/	0	195
20	/	163	/	76	146	161	/	104	95	129	85	/	150	63	/	/	/	/	102	0
Product 6																				
1	0	/	97	150	96	/	/	157	/	181	122	71	/	194	/	138	112	/	128	/
2	79	/	133	187	131	/	/	66	/	161	189	135	/	169	/	182	174	/	189	/
3	175	/	0	55	161	/	/	146	/	110	107	122	/	54	/	110	167	/	153	/
4	155	/	139	0	162	/	/	111	/	164	92	144	/	119	/	165	92	/	131	/
5	159	/	173	198	0	/	/	112	/	171	194	126	/	50	/	144	171	/	69	/
6	123	/	151	105	138	/	/	103	/	183	125	64	/	192	/	149	197	/	149	/
7	96	/	104	93	192	/	/	188	/	155	100	51	/	92	/	144	59	/	77	/
8	199	/	55	161	161	/	/	0	/	113	122	200	/	190	/	150	142	/	87	/
9	90	/	172	146	189	/	/	186	/	183	149	161	/	95	/	125	133	/	99	/
10	104	/	132	136	184	/	/	113	/	0	57	150	/	57	/	157	198	/	143	/
11	183	/	162	67	53	/	/	128	/	165	0	200	/	126	/	154	184	/	151	/
12	115	/	120	109	132	/	/	166	/	184	102	0	/	147	/	125	88	/	122	/
13	52	/	93	80	65	/	/	106	/	100	64	167	/	193	/	84	133	/	121	/
14	171	/	101	53	187	/	/	119	/	152	104	185	/	0	/	55	91	/	52	/
15	89	/	135	123	110	/	/	95	/	175	163	138	/	121	/	134	98	/	178	/
16	66	/	198	135	192	/	/	97	/	178	127	128	/	98	/	0	58	/	76	/
17	91	/	97	194	137	/	/	160	/	57	157	178	/	131	/	65	0	/	144	/
18	93	/	144	59	126	/	/	54	/	175	116	177	/	109	/	135	108	/	57	/
19	71	/	124	188	196	/	/	150	/	72	69	199	/	159	/	95	54	/	0	/
20	72	/	72	148	147	/	/	125	/	182	179	75	/	181	/	176	130	/	102	/
Product 7																				
1	0	161	78	132	/	/	161	94	/	88	196	/	/	164	119	/	112	/	158	/
2	184	0	120	129	/	/	95	156	/	118	139	/	/	70	63	/	66	/	59	/
3	105	200	0	61	/	/	60	82	/	135	142	/	/	87	134	/	121	/	70	/
4	182	110	58	0	/	/	69	93	/	68	150	/	/	55	77	/	52	/	78	/
5	138	171	134	135	/	/	104	100	/	70	125	/	/	53	81	/	193	/	145	/
6	117	145	132	109	/	/	130	157	/	77	136	/	/	56	112	/	137	/	146	/
7	60	156	185	108	/	/	0	100	/	86	196	/	/	58	84	/	199	/	55	/
8	185	195	126	93	/	/	56	0	/	179	152	/	/	199	174	/	162	/	82	/
9	149	85	133	80	/	/	165	166	/	116	83	/	/	63	170	/	106	/	103	/
10	142	120	95	105	/	/	176	135	/	0	153	/	/	126	168	/	115	/	163	/

Table 13 (continued)

Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20
Product 7																				
11	162	121	146	153	/	/	169	178	/	77	0	/	/	172	180	/	194	/	126	/
12	87	116	68	149	/	/	71	112	/	78	95	/	/	50	183	/	52	/	106	/
13	129	75	180	175	/	/	87	164	/	66	115	/	/	156	146	/	82	/	64	/
14	62	117	59	128	/	/	139	151	/	128	192	/	/	0	56	/	185	/	180	/
15	114	126	89	177	/	/	189	149	/	188	77	/	/	125	0	/	181	/	110	/
16	125	177	116	176	/	/	179	121	/	142	151	/	/	53	149	/	92	/	78	/
17	146	127	191	157	/	/	121	98	/	187	200	/	/	86	101	/	0	/	110	/
18	142	53	71	190	/	/	99	193	/	94	73	/	/	89	195	/	194	/	63	/
19	168	60	66	105	/	/	145	98	/	68	163	/	/	52	91	/	126	/	0	/
20	189	80	65	51	/	/	181	173	/	151	172	/	/	98	120	/	99	/	179	/
Product 8																				
1	/	134	/	128	132	172	142	/	126	61	114	166	/	/	/	81	/	62	95	/
2	/	0	/	94	165	75	179	/	147	162	62	196	/	/	/	144	/	173	86	/
3	/	70	/	56	117	166	144	/	132	81	188	146	/	/	/	152	/	162	64	/
4	/	68	/	0	136	107	75	/	80	138	163	82	/	/	/	142	/	81	111	/
5	/	142	/	57	0	173	121	/	194	181	52	163	/	/	/	184	/	169	110	/
6	/	172	/	129	119	0	112	/	73	76	130	159	/	/	/	130	/	180	61	/
7	/	199	/	134	165	78	0	/	108	198	112	143	/	/	/	176	/	64	74	/
8	/	117	/	168	67	140	85	/	53	178	146	59	/	/	/	134	/	114	168	/
9	/	142	/	169	59	87	139	/	0	94	193	122	/	/	/	89	/	123	58	/
10	/	181	/	118	153	104	162	/	181	0	131	65	/	/	/	194	/	147	94	/
11	/	183	/	68	134	174	191	/	57	91	0	104	/	/	/	163	/	171	116	/
12	/	182	/	112	66	60	70	/	117	183	180	0	/	/	/	130	/	88	55	/
13	/	163	/	139	200	75	111	/	198	147	123	166	/	/	/	127	/	123	115	/
14	/	120	/	104	118	192	92	/	103	149	111	74	/	/	/	147	/	114	165	/
15	/	144	/	66	70	129	147	/	171	129	88	199	/	/	/	153	/	183	63	/
16	/	140	/	95	127	87	149	/	115	148	83	186	/	/	/	0	/	175	103	/
17	/	67	/	85	106	101	101	/	185	78	200	199	/	/	/	108	/	70	155	/
18	/	162	/	150	112	89	82	/	82	171	186	147	/	/	/	132	/	0	139	/
19	/	83	/	146	122	64	150	/	129	118	115	200	/	/	/	92	/	150	0	/
20	/	183	/	56	119	195	118	/	129	92	197	168	/	/	/	75	/	131	91	/

Table 14 The worker-to-*seru* assignment plans and feasible task-to-worker training plans

	Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20	T_c	D_j
Seru 1																							
plan 1																						887	129.283
	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0		
	16	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0		1	0	1	1		
Plan 2																						891	32.967
	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0		
	16	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1		
Plan 3																						896	0.157
	1	0	1	0	0	0	1	0	0	0	1	0	1	1	0	0	0	1	0	0	0		
	16	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	1	1		
Seru 2																							
Plan 1																						914	25.861
	5	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0		
	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0		
	20	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1		
Plan 2																						954	0.469
	5	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0		
	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0		
	20	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1		
Seru 3																							
Plan 1																						573	0.361
	2	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0		
	6	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0		
	19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1		
Seru 4																							
Plan 1																						667	91.669
	9	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0		
	14	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0		
	18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0		
Plan 2																						683	0.001
	9	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0		
	14	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
	18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0		

Table 14 (continued)

	Worker	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11	Task12	Task13	Task14	Task15	Task16	Task17	Task18	Task19	Task20	T_c	D_j
Seru 5																							
Plan 1																						690	36.835
	8	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0		
	10	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0		
	11	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1		
Plan 2																						695	1.190
	8	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		
	10	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0		
	11	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1		
Seru 6																							
Plan 1																						923	130.771
	7	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0		
	13	1	0	1	1	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0		
Plan 2																						934	38.816
	7	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0		
	13	1	0	0	1	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0		
Plan 3																						947	0.414
	7	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0		
	13	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0		
Seru 7																							
Plan 1																						751	0.825
	3	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0		
	4	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0		
	12	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0		
Seru 8																							
Plan 1																						1,542	0
	15	0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	1	0	1	1	0		

Acknowledgments This research is supported by the National Natural Science Foundation of China (71171161). We would like to express our heartfelt and sincere thanks to Weiyue Zhang for her valuable discussions and comments. We gratefully acknowledge the anonymous referees for their constructive criticisms and valuable comments. We greatly appreciate Ms. Bunavilla V. Gabuna for her kind work and time spent on editing our paper.

References

1. Asano S (1997) Internationalization of Japanese company and change of production system. Rikkyo Econ Rev 51(1):29–55. (in Japanese)
2. Asao U, Fujita E, Tamura Y (2004) The division of labor on the shop floor, and its social and institutional background in Japanese motor vehicle and electric industries from the viewpoint of Swedish experience. Douzieme Rencontre Internationale du Gerpisa, Paris
3. Asgari B, Yen LW (2009) Accumulated knowledge and technological progress in terms of learning rates: a comparative analysis on the manufacturing industry and the service industry in malaysia. Asian J Technol Innov 17(2):71–99
4. Benayoun R, Montgolfier JD, Tergny J, Laritchev O (1971) Linear programming with multiple objective functions: step method (stem). Math Program 1(1):366–375
5. Cesaní VI, Steudel HJ (2005) A study of labor assignment flexibility in cellular manufacturing systems. Comput Ind Eng 48(3):571–591
6. D&M Nikkei Mechanical (2003) The challenge of cano-part 3. D&M Nikkei Mechanical 588:70–73
7. Davis D, Kher HV, Wagner BJ (2009) Influence of workload imbalances on the need for worker flexibility. Comput Ind Eng 57(1):319–329
8. Davis D, Mabert VA (2000) Order dispatching and labor assignment in cellular manufacturing systems. Decis Sci 31(4):745–771
9. Ertay T, Ruan D (2005) Data envelopment analysis based decision model for optimal operator allocation in CMS. Eur J Oper Res 164(3):800–810
10. Estrada DC, Shukla A, Cochran DS (2000) Converting from moving assembly lines to cells. In: Proceedings of the third world congress on intelligent manufacturing processes and systems, MA, pp 28–30
11. Hopp W, Tekin E, Van OM (2004) Benefits of skill chaining in serial production lines with cross-trained workers. Manag Sci 50(1):83–98
12. Iravani SM, Kolfal B, Van OMP (2007) Call-center labor cross-training: it's a small world after all. Manag Sci 53(7):1102–1112
13. Isa K, Tsuru T (2002) Cell production innovation in Japan: toward a new model for Japanese manufacturing? Ind Relat 41(4):548–578
14. Iwamuro H (2004) Cellular manufacturing system. Nikkan Kogyo Shimbun, Tokyo. (in Japanese)
15. Johnson DJ (2005) Converting from moving assembly lines to cells at sheet metal products: insights on performance improvements. Int J Prod Res 43(7):1483–1509
16. Kaku I, Gong J, Tang J, Yin Y (2009) Modeling and numerical analysis of line-cell conversion problems. Int J Prod Res 47(8):2055–2078
17. Kaku I, Murase Y, Yin Y (2008) A study on human-task-related performances in converting conveyor assembly line to cellular manufacturing. Eur J Ind Eng 2(1):17–34
18. Kher HV, Malhotra MK (1994) Acquiring and operationalizing worker flexibility in dual resource constrained job shops with worker transfer delays and learning losses. Int J Manag Sci 22(5):521–533
19. Kimura T, Yoshita M (2004) Cellular manufacturing runs into trouble when nothing is done. Nikkei Monozukuri 7:38–61. (in Japanese)
20. Kono H (2004) The aim of the special issue on *seru* manufacturing. IE Rev 45:4–5
21. Kuo Y, Yang T (2007) Optimization of mixed-skill multi-line operator allocation problem. Comput Ind Eng 53(3):386–393
22. Liu C, Li W, Lian J, Yin Y (2012) Reconfiguration of assembly systems from conveyor assembly line to serus. J Manuf Syst 31(3):312–325
23. Liu C, Lian J, Yin Y, Li W (2010) Seru seisan—an innovation of the production management mode in Japan. Asian J Technol Innov 18(2):89–113
24. Liu C, Stecke K, Lian J, Yin Y (2013) An implementation framework for *Seru* production. Int Trans Oper Res doi:10.1111/itor.12014
25. McDonald T, Ellis KP, Van AEM, Koelling CP (2009) Development and application of a worker assignment model to evaluate a lean manufacturing cell. Int J Prod Res 47(9):2427–2447
26. Miyake DI (2006) The shift from belt conveyor line to work-cell based assembly systems to cope with increasing demand variation in Japanese industries. Int J Automot Technol Manag 6(4):419–439
27. Molleman E, Slomp J (1999) Functional flexibility and team performance. Int J Prod Res 37(8):1837–1858
28. Noguchi H (2003) Production involution in Japan. Nikkan Kogyo Shimbun, Tokyo. (in Japanese)
29. Norman BA, Tharmmaphornphilas W, Needy KL, Bidanda B, Warner RC (2002) Worker assignment in cellular manufacturing considering technical and human skills. Int J Prod Res 40(6):1479–1492
30. Sakazume Y (2006) Conditions for successful implementation of assembly cells. Ind Eng Manag Syst 5(2):142–148
31. Sakikawa T (2005) HRM practices and operational/financial performance: evidence from cell production at Japanese electronics manufacturers. Ann Rep Econ, Nigata University 29:17–45
32. Sakikawa T (2006) HRM practices for Japanese cell production. Ann Rep Econ, Nigata University 30:1–29
33. Satoglu SI, Suresh NC (2009) A goal-programming approach for design of hybrid cellular manufacturing systems in dual resource constrained environments. Comput Ind Eng 56(2):560–575
34. Shinohara T (1995) Shocking news of the removal of conveyor systems: single-worker cellular manufacturing system. Nikkei Mech 459:20–38
35. Shinohara T (1997) Seru seisan mode: a great turning of manufacturing floor from the division of labor to the integration of labors. Nikkei Mech 497(1):39–41. (in Japanese)
36. Shirai K (2001) Cell production and managing human resources. In: Tsuru, T (ed) Innovation and evolution of production systems: diffusion of cell production among Japanese firms. *Nihon Hyoron Sya*, pp 87–121 (in Japanese)

37. Slomp J, Bokhorst JAC, Molleman E (2005) Cross-training in a cellular manufacturing environment. Comput Ind Eng 48:609–624
38. Stecke KE, Yin Y, Kaku I, Murase Y (2012) Seru: the organizational extension of JIT for a super-talent factory. Int J Strateg Decis Sci 3(1):106–119
39. Süer GA, Tummaluri RR (2008) Multi-period operator assignment considering skills, learning and forgetting in labour-intensive cells. Int J Prod Res 46(2):469–493
40. Takeuchi N (2006) Manufacturing methods illustrated: cell production system. JMA Management Center Inc., Tokyo (in Japanese).
41. Treleven M (1989) A review of dual resource constrained systems research. IIE Trans 21(3):279–287
42. Tsuru T (1997) Current trends in production systems and human resources management in the Japanese machinery industries. The Economic Research Institute of Japan Society for the Promotion of Machine Industry, Tokyo
43. Warner RC, Needy KL, Bidanda B (1997) Worker assignment in implementing manufacturing cells. In: Proceedings of the 1997 industrial engineering research conference. Miami Beach
44. Wemmerlov U, Johnson DJ (1997) Cellular manufacturing at 46 user plants: implementation experiences and performance improvements. Int J Prod Res 35(1):29–49
45. Yamada H, Kataoka T (2001) Unusual production revolution. (in Japanese)
46. Yagyuu S (2003) Synchronized cellular manufacturing system. Nikkan Kogyo Shimbunsha, Tokyo. (in Japanese)