Investigating the performance improvement by conversion of assembly line configuration to a pure cell system in manufacturing industry

T. Venkata Deepthi

Department of Mechanical Engineering,
Koneru Lakshmaiah Education Foundation,
Green Fields, Vaddeswaram, A.P, India
Email: venkatadeepthi.t@gmail.com

K. Ramakotaiah

Department of Mechanical Engineering,
KKR & KSR Institute of Technology and Sciences,
Vinjanampadu, Vatticherukuru Mandal, Guntur, AP, India
Email: krk_ipe@yahoo.co.in

Vijaya Kumar Manupati*

Department of Mechanical Engineering,
National Institute of Technology Warangal,
Warangal, Telangana, India
Email: manupati.vijay@nitw.ac.in
*Corresponding author

Chaitanya Gangal

Purdue University,
IN 47907, USA
Email: gangal.chaitanya@gmail.com

Abstract: Seru production systems implement reconfiguration of traditional assembly lines to a flexible cell system that aim at reducing the required workforce while at the same time augmenting the productivity manifold. For evaluating the overall performance improvement, cell formatting and assigning workers to serus takes form of a complicated decision problem. In this paper, with the objective of reducing the total cost for training the worker, minimising the processing time and the total throughput time, mathematical insights on the solution space of a multi-objective line-cell conversion model are identified, in turn proving it to be an NP-hard model. By applying the proposed heuristic algorithm on several numerical simulations, a Pareto-optimal solution of this multi-objective model is obtained. With experimental results and comparative studies, the proposed approach proves its effectiveness that may lead to further improvement in seru production systems competitive advantage to cope with fluctuating market demands by enhancing the flexibility as well as the efficiency of the system. [Submitted: 18 June 2016; Revised 29: August 2017; Revised: 25 August 2018; Revised: 5 December 2018; Accepted: 8 January 2019]

Keywords: seru; manufacturing systems; production systems; flexibility; reconfiguration; cellular manufacturing.

Reference to this paper should be made as follows: Deepthi, T.V., Ramakotaiah, K., Manupati, V.K. and Gangal, C. (2019) 'Investigating the performance improvement by conversion of assembly line configuration to a pure cell system in manufacturing industry', *European J. Industrial Engineering*, Vol. 13, No. 6, pp.723–745.

Biographical notes: T. Venkata Deepthi received her BTech degree in Department of Mechanical Engineering at Sir. C.R. Reddy College of Engineering and her MTech in Computer Integrated Manufacturing at the Department of Mechanical Engineering as a specialisation from K.L. University. She is currently pursuing her PhD degree in the area of recent assembly systems in manufacturing organisation.

K. Ramakotaiah received his PhD from SCSUMU University. He received under graduation and post-graduation degrees in the mechanical and manufacturing engineering respectively. His current research interests include machine tool vibrations, micro machines and condition monitoring of machines. He has published several papers in reputed International journals.

Vijaya Kumar Manupati received his PhD in Department of Industrial and Systems Engineering from Indian Institute of Technology Kharagpur. Currently, he is working as an Assistant Professor in Department of Mechanical Engineering, National Institute of Technology Warangal. His current research interest includes intelligent manufacturing systems, agent/multi-agent/mobile-agent systems for distributed control, simulation, integration of process planning and scheduling in distributed manufacturing environment. He published more than 60 papers which includes journals like *International Journal of Production Research*, *Computers and Industrial Engineering*, *Computer Integrated Manufacturing*, *International Journal of Advanced Manufacturing Technology*, etc. He is a reviewer for more than 40 peer reviewed journals and member of Institute of Industrial and Systems Engineering.

Chaitanya Gangal received his BTech degree in the Department of Mechanical Engineering from VIT University. He is pursing his Masters from Purdue University.

1 Introduction

Due to complexity and competitiveness of today's commerce, the key to consistent business success is application of newer and more effective production systems. Under the traditional batch of manufacturing systems, like conveyor assembly lines, workers do not have major responsibility for quality control during their work. Although workers are encouraged to make good-quality products, rarely are they penalised for producing poor work quality Furthermore, in an assembly line production, work can become repetitive offering little in the way of 'mental stimulation and creative critical thinking'. The volatility of market demands has impelled leading manufacturers to pursue alternative production systems.

Seru production system is one such system which aims at eliminating the inherent drawbacks of traditional methods. The prime objectives of seru production systems are achieving a low inventory and higher flexibility. With the ability to be responsive to the rapidly changing consumption structure, under seru production systems, the quality goes up, lead times goes down, factors of production are more efficient, and waste is reduced, causing costs to drop, all at the same time. Japanese manufacturing companies such as Sony, Panasonic and Hitachi have significantly increased their profit margins and production pace by switching to this seru manufacturing systems.

Serus are assembly cells which form the foundation of cell-based production models as defined by Endou (2004). Unlike in conveyor assembly systems, in each seru a single worker is devoted to the assembly of a single product such a seru is called a yatai (Stecke et al., 2012). Here, the multi-skilled workers complete all the tasks in a fixed order from one work station to another. There also exist intermediate serus like divisional and rotating where semi-skilled workers co-operate with one another to complete the job. But still each worker operates more tasks than he used to in conveyer assembly line systems. Among these different kinds of serus, the efficiency of a yatai is the highest and the ultimate goal of reconfiguration of conveyor assembly line is to make multi-skilled workers to perform in these yatais. Ying and Tsai (2017) aimed at improving the efficiency and minimising the total cost of workers by introducing the seru production system which emphasises on flexibility and multi-skilled workers. The seru assignment plan with respect to the worker was also derived through a two-phase heuristic algorithm.

A large number of Japanese manufacturing factories have gained great benefits using serus (Yin et al., 2008; Stecke et al., 2012). Yang (2014) developed a multi-objective optimisation model to investigate the line to cell configuration conversion through the two factors of total throughput time and total labour hours. Seru production has been called beyond lean in Japan (Shinobu, 2003) and can be considered to be an ideal manufacturing mode to realise mass customisation (Liu et al., 2010). The theory of swift, even flow, with implications for future research of trade-offs related to production efficiency, responsiveness, and competitiveness in high-cost markets was elaborated through many case studies (Yong et al., 2017; Stecke et al., 2012; Kaku, 2017). Early data gathered by the Japan Society for the Promotion of Machinery Industry (1998) through a large survey of Japanese manufacturing industries showed that about 48% of the 227 respondents had implemented or were planning to adopt *seru* production. The key enabling technologies for seru production are identified through systematic review and evaluation as they are directly related to the sustainable performance of the seru production and aid in developing practical methods to implement the sustainable operations (Xiao et al., 2017). Kaku et al. (2009) evaluates the sustainability of implementation of seru systems by undertaking a thorough research on Japanese seru production through a number of real case studies. (Liu et al., 2015) developed a non-dominated sorting genetic algorithm II (NSGA-II) considering three numeral examples to investigate the performance measures like economy and environment of a seru production problem and confirmed the effectiveness of the proposed algorithm. Line-seru problem has been characterised with respect to many orders of one product type to minimise the total labour cost and total flow time (Shao et al., 2016a). Seru-line conversion and different seru combinations has been analysed to improve efficiency by taking a real example of an electronics industries (Shao et al., 2016b) addressed seru production system based on queuing theory for maximising the utilisation of workers by forming suitable serus. A bi-objective optimisation nonlinear stochastic model has been developed to simplify the problem of serus formation and arranging product orders among them.

With various market environments, as well as conditions varying from factory to factory, the implementation details of seru production are not always identical. However, a general implementation framework should benefit most factories. Based on extensive investigations of many manufacturing factories' practices, and considering the fact that most manufacturing managers are unfamiliar with basic theoretical knowledge of seru production, we have proposed a general implementation framework for seru production systems.

In this paper, we have analysed three major factors that influence the productivity of a cell-based manufacturing system. With the goals of reducing the total training cost of workers, processing time of each worker and minimising the total throughput time, the mathematical insights on the solution space of the multi-objective problem are identified. Then with the help of a numerical example, we proved that formatting workers in the line-cell conversion model is an NP hard problem. For this formatting of workers in different serus we used a heuristic algorithm which based on workers efficiency, processing time and their training cost formulates a worker to seru assignment plan. After this formatting, a Pareto-optimal solution was achieved and the paper was concluded after analysing the results from this plan. Moreover, a comparative study has been conducted and the results establish the fact that the proposed approach performs consistently well.

The paper has been arranged as follows: Section 1 gives the introduction to the problem discussing its scope in the present scenario. The description of the multi-objective problem and its goals are presented in Section 2. In Section 3 the problem is solved with the help of the proposed heuristic algorithm. Experimentation is carried out in Section 4. Results of computed experiments were discussed in Section 5. In Section 6 the necessary conclusions are derived depending on the efficiency of the system and finally concludes the paper providing directions for future work.

2 Problem description

Most manufacturing companies employ traditional conveyor lines for assembly of products. However, in the present market scenario, which exhibits high fluctuation in demand with higher variety and lower volume needs, traditional conveyer assembly systems prove to be a bit less suitable. To overcome this drawback of the traditional system, seru production systems were introduced in Japan, specifically in the electronic industry where product assembly is the major part of production process. Such production system is thus seen as the core element in the canon company, extensive information regarding the same is available in Liu et al. (2013). Ricoh, a middle scale electronic enterprise in Japan can be taken as a real example of seru production system. After undergoing crucial changes in the production system, Ricoh has adopted 'seru-formation production system', in which serus comprise of whole manufacturing process without using conveyor belts. It resulted in around eighty times less power consumption than previous levels. In traditional systems, the each worker perform a single task, which leads to higher processing time and reduced overall flexibility Liu et al. (2013). Here every worker takes different time to perform the task and thus the production pace is determined by the performance of the least skilled worker. Furthermore, since there is only a one-to-one relation between worker and a particular task, work tends to get monotonous and there is no augmentation of skills relating to the assembly of the product.

In contrast, seru production systems requires workers to be cross-skilled or fully-skilled in the assembly of a product, which enables them to perform multiple tasks on the product or complete the entire assembly of a product by themselves. Such a system leads to increase in productivity and product quality while reducing manufacturing work space, lead times and investment capital. In this system, each seru represents a single product and it contains one or more than one workers who finish the product from start to finish in the seru itself. Thus the highly efficient workers can perform to the best of their abilities. Additionally, it eliminates the risk of causing further delay in production due to the absenteeism or inconsistency of a worker as there are multiple serus which are producing the product simultaneously.

In this project, we consider a seru production system with several work stations. Each workstation corresponds to a specific task and all the same type of products is made in the same seru. All serus run independently without disturbing one another. Our main concern is to determine the satisfied or optimal worker-to-seru assignment plan and task-to-worker with the objective to minimise the total training cost and reducing the processing time of each worker and the total through put time. We can evaluate the total through put time (TTPT) and total labour power (hours), where the former represents the system productivity that is the time of all the product batches assembled and later represents the work efficiency.

2.1 Assumptions

The following assumptions are entailed in our paper

1 A seru can only process one specific product at a time.

2 The workers engaged are preferred to be multi-skilled so that they can finish the whole product by themselves. Training a single-skilled worker to a multi-skilled worker by improving the process skills in serus is necessary to adjust the station operations.

3 The number of workers in the cell-based system is almost equal or slightly more than the total number of products manufactured by the system. Since each worker is multi-skilled and capable to process all operations required for a product type thus, it is ideal to have only a single worker in each seru.

4 The number of workers remains even after the reconfiguration of the manufacturing system to avoid the fluctuations in processing time and total training cost.

5 The setup time is only considered when a single seru produces two different products successively otherwise it is assumed to be zero.

6 As all the tasks of a product are performed in a single seru and there is no delay between consecutive tasks because all the tasks required for a particular product type are performed by a multi-skilled worker.

7 It is assumed that the types and batches for producing the products are known in advance, so that the workers can be trained accordingly.

2.2 Problem formulation

An assembly product mix with M product batches and N product types has been considered for this paper. When assembly lines are reconfigured to cells, W workers are assigned to these assembly cells. A first-come-first-serve principle is used for assignment of batches to these cells. The same principle is used to define the total throughput time of the cell system. As we have already discussed, reallocation of the semi-skilled workers from the assembly lines to the seru workstation may cost us more times as the worker may not be familiar with the assembly of the entire product. It is thus a reasonable assumption that a worker's skill level varies with the task that has been assigned to him or her; so if a worker p's tasks within a cell exceeds his or her upper bound n_p, i.e., $W > n_p$, then the worker costs more task time than his or her task time in the original assembly line.

We set $t_q^l = 0,$ if the seru q does not include task l in it. Thus, for this case, the training cost required to train the worker for task is assumed to be infinity as the worker need not be trained for this task. That is, $C_{pq}^l = \infty$ where E_q in equation (1) is the average processing time over all workers in seru q, can be obtained by

$$E_q = \frac{\sum_{l=1}^{L} t_q^l}{\sum_{p=1}^{P} \phi_{pq}} \tag{1}$$

Initially the total training cost is calculated and then the basic assignment plan is prepared. In this project, we assume that if the number of worker p*'s* tasks within a cell is over her or his upper bound η_p i.e., $W > \eta_p$, then according to equation (2) the worker will cost more average task time than her or his task time within the original assembly line. The details are given as follows:

$$C_p = \begin{cases} 1 + \in_p \left(W - \eta_p\right). & W > \eta_p \\ 1, & W \leq \eta_p \end{cases} \quad \forall p \tag{2}$$

The task time of the product varies with workers skill levels. Thus task time is calculated i.e., time of product batch m per station in a cell can be represented by the following equation (3)

$$TC_m = \frac{\sum_{v=1}^{V} \sum_{p=1}^{W} \sum_{q=1}^{Q} \sum_{l=1}^{U} S_{uv} T_v \phi_{vp} C_p G_{pq} H_{uql}}{\sum_{p=1}^{W} \sum_{q=1}^{Q} \sum_{l=1}^{U} G_{pq} H_{uql}} \tag{3}$$

Finally, the setup time MN_u, the flow time EF_u and the begin time EFG_u of product batch m are represented in equations (4), (5) and (6) for the minimisation of total through put time (TTPT).

$$MN_u = \begin{cases} MNP_u M_{uv}, & M_{uv} = M_{u'v} = 1 \\ 0, & M_{uv} = 1,\ M_{u'v} = 0 \end{cases} \quad \left(u'/H_{uql} = 1,\ H_{u'q(l-1)} = 1,\ \forall q, l\right) \tag{4}$$

$$EF_u = \frac{B_u TC_u W}{\sum_{p=1}^{W} \sum_{q=1}^{Q} \sum_{l=1}^{M} G_{pq} H_{uql}} \tag{5}$$

$$EFG_u = \sum_{s=1}^{u-1} \sum_{q=1}^{q} \sum_{l=1}^{m} \left(EF_s + MN_s\right) H_{uql} H_{sq(l-1)} \tag{6}$$

The details are as follows: As the workers' skill levels vary, the task time of each product will also vary. For a particular cell, the average task time of workers in a cell is taken to be the task time of the product assembled in the cell.

3 Heuristic algorithm

The above reconfiguration problem is solved by using a Heuristic algorithm having two phases (Liu et al., 2013). In the first phase, the worker to seru assignment plan is chalked by including the total training cost of each worker while the second phase focuses on the minimisation of the total throughput time.

3.1 Phase 1: to find worker to seru and task to worker assignment plan

In this paper, the concept of 'optimum' is modified into 'best compromise' solution for the proposed multi-objective problem. The analysis of the characteristics of the above problem allows us to develop a three stage's algorithm with nine steps to attain the goals of drawing up the assignment plans. The proposed algorithm has the following three stages viz. to get a worker-to-seru assignment plan, to obtain all feasible task-to-worker assignment plan and lastly to determine the final satisfied task-to-worker training plan. The detailed methodology of the algorithm is given as follows.

3.1.1 Stage 1: to obtain the worker to seru assignment plan

Step 1 Calculate the training cost for each worker for attaining all the skills required in all tasks of each product. That is, calculate $\sum_{l=1}^{L} C_{pq}^{l}$ for any p, q.

Step 2 Assign each worker to a seru of a specific product type for which he/she has the minimum training cost. If there are more than one serus where he can be assigned, chose the seru which has the least processing time. If there still is a clash, assign him to the one with the smallest index number.

Step 3 Form the set S_s whose elements are all the serus which have not been allocated any workers in them and form set S_w with serus having workers more than the predetermined size as its elements. Check if S_s is 0

Case 1: If $S_s = 0$. Check whether $S_w = Ø$.

Sub-case 1: If $S_w = 0$,

End stage 1.

As this condition satisfies the constraint related to the number of workers of seru, i.e., there should not be any seru which does not have any workers and also there should not be any which exceeds the predetermined worker size, the stage 1 ends of the algorithm ends here.

Sub-case 2: If S_w _ = Ø.

Thus by considering the minimum of the increased training cost, we remove one or more workers from serusS_w and shift them to serus belonging to S_s and make all serus of the predetermined sizes. Then we end stage 1.

Case 2: If S_s_ = 0, Check if $S_w = 0$.

Sub-case 1: If $S_w = 0$.

Sub-case 2: If S_w _ = 0.

If the number of the remaining workers in all serus belonging to S_w drop to the predetermined seru size (i.e., when $S_w = 0$), but there still exist serus in which no workers are assigned, continuously transfer one or more workers from serus not belonging to S_s to serus in S_s. This should be done by considering the minimisation of the increased total training cost, until the number of workers in each seru is more than 0. End stage 1. For the case when there are serus which have more number of workers than the predetermined sizes (i.e., S_w_ = 0), again consider the minimisation of the increased training cost and accordingly remove one or more workers from serus in S_w to serus in Ss and continue doing so till it satisfies the constraints in seru size and also such that each seru has at least one worker. End stage 1. After finishing stage 1 we obtain the worker-to seru assignment plan. In the following stage 2 and stage3, the approaches for all serus are same. Therefore, for each seru, we have the following steps.

3.1.2 Stage 2: to obtain a feasible task-to-worker training plan

Step 4 Assign a task to that worker who requires the least training cost for it. If there is a clash between two or more workers, assign the task to the worker who has the minimum index number. From this assignment process we get the initial task to worker plan.

Step 5 Now for this task-to-worker assignment plan, compute again the total training cost required for each worker. Find the worker who needs maximum training i.e., who has the maximum training cost ($W_{\max}$) and also the worker who has the minimum training cost (W_{min}). If there is a clash between them, resolve it by selecting the workers with the minimum index number. Now calculate the difference between the maximum processing time and minimum processing time between these two workers (dt).

Step 6 Now calculate the difference between the training cost of each task between the workers W_{max} and W_{min}. Find the task which corresponds to minimum difference in the training costs. If there is a tie, select the one which has minimum difference in the processing times. For further ties, select any one of the task randomly. This task is denoted by O_s.

Step 7 Now compare if d_t is greater than O_s.

Case 1: If d_t is greater than O_s, reallocate the task O_s from W_{max} to W_{min}. Continue this process and obtain at new feasible plan. Then again go to Step 5.

Case 2: If d_t is less than the time required for O_s, go to Step 8.

Step 8 From the above steps, collect all feasible training plan. All these plans are efficient task-to-worker training plans.

3.1.3 Stage 3: to determine the final satisfied task-to-worker training plan(s)

Step 9 We get alternate assignment plans from all the above steps. All these are feasible and are equally efficient. One or more worker-to-seru assignment plans can then be chosen from these alternative plans by applying one of the measurement performances following a predetermined rule.

3.2 Phase 2: to minimise the total throughput time

This phase of the algorithm focuses on decreasing the throughput time by reducing the number of workers in a seru and in turn reduces the labour force. The objective is to find a Pareto-optimal solution in the direction towards decreasing W which is the number of workers in the line cell conversion process. The steps involved are:

Step 1 This is the initialisation step and begins by setting $G = \Phi$ (the set of non-empty proper subsets of the set $\{1, 2, \ldots, W\}$), $F = \Phi$ (the set of solution with the minimum TTPT in every non-empty proper subset) and $N = \Phi$ (the set of solutions attending final non-dominated sorting).

Step 2 Generate $(2^W - 2)$ number of non-empty proper subsets (G_i) of the set $\{1, 2, \ldots, W\}$ by using the recursive algorithm. The cardinality of G, $|G| = 2^W - 2$.

Step 3 For each subset G_i belonging to the set G, produce the set of ordered set partitions (S_i) of G_i as the set of feasible solutions. Follow this by initialising the minimum TTPT (mTS_i) of Si such that it corresponds to the equation (7).

$$mTS_i = \infty(\text{infinity}) \tag{7}$$

For each *sj* belonging to *Si* there arise two subcases:

Case 1. If the TTPT of $S_j < mTS_i$ then

mTS_i = TTPT of S_j

Case 2. If not then,

j = j + 1

After performing the calculations in subcases 1 or 2 add mTS_i into F.

Step 4 According to the number of workers, partition F into (W – 1) number of subsets Fs. The main intension to make a subset is to assign the workers equally for all the products i.e., each product has the same number of workers to perform the task.

Step 5 For each Fs which is equal to F initialise the minimum TTPT (mTFs) of $FsmTFs$ to infinity. Similarly for each fj belonging to Fs there are two subcases:

Case 1. If the TTPT of $f_j < mTFs$ then

$mTFs$ = TTPT of fj

Case 2. If not then,

J = j + 1

Following the implementation of either of the subcases add $mTFs$ to N.

Step 6 We obtain the non-dominated solutions of N as output in this final step of the algorithm. There are utmost (W – 1) solutions undergo the non-dominated sorting.

4 Experimentation

In Table 1 the standard processing time of each task of each product of the worker before he is trained is given. The products can be a link or a roller and a roller which go through a series of operations such as induction hardening, tempering, MPI, welding, turning, brush pressing, assembly, oiling and washing. Each of the products may or may not require the same operations and may also follow a different sequence accordingly. The sign of '/' means the task does not exist in the corresponding product type. In Table 2, '0' indicates that the worker is already skilled at the corresponding task, and the sign of '/' means that the following task is not required in the product and thus training is not necessary. This computational case has been solved using MATLAB programming.

Table 1 The standard processing time of each task of each product type before training the worker

Product type	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*	*Task 9*	*Task 10*
Product A	81	50	95	/	87	67	/	74	80	95
Product B	/	73	/	78	69	95	/	93	54	84
Product C	84	82	63	/	69	/	/	/	56	/
Product D	86	/	70	66	71	54	89	58	65	91
Product E	92	88	88	63	50	63	56	79	/	94
Product F	82	63	86	95	/	67	61	80	85	91

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

Product type	*Worker*	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*	*Task 9*	*Task 10*
Product A	1	0	30	56	/	20	62	/	36	113	117
	2	62	0	158	/	109	92	/	54	7	80
	3	53	1	0	/	50	49	/	90	17	114
	4	102	145	151	/	62	87	/	96	95	47
	5	80	90	43	/	0	132	/	77	21	94
	6	15	67	67	/	122	0	/	118	41	74
	7	96	146	32	/	167	135	/	55	100	142
	8	49	10	101	/	169	123	/	0	69	163
	9	147	136	23	/	83	104	/	132	0	99
	10	61	81	31	/	140	161	/	105	25	0
Product B	1	0	11	105	9	70	/	176	84	/	42
	2	48	0	179	90	106	/	6	122	/	55
	3	63	167	0	64	146	/	68	67	/	154
	4	110	39	23	0	123	/	133	8	/	117
	5	130	109	62	55	0	/	87	136	/	106
	6	86	60	60	45	86	/	78	46	/	144
	7	103	176	73	83	139	/	0	175	/	52
	8	91	33	18	130	167	/	76	0	/	60
	9	60	64	56	12	55	/	85	50	/	68
	10	55	153	120	2	13	/	10	21	/	0
Product C	1	0	/	95	71	53	/	134	141	115	93
	2	55	/	94	125	107	/	79	60	161	53
	3	86	/	0	38	52	/	9	73	15	71
	4	66	/	138	0	65	/	72	110	62	62
	5	82	/	89	113	0	/	40	80	149	54
	6	167	/	61	80	71	/	101	106	103	24
	7	65	/	1	85	52	/	0	179	58	3
	8	128	/	131	90	31	/	60	0	118	11
	9	73	/	2	6	104	/	101	137	0	78
	10	131	/	169	12	31	/	119	21	146	0
Product D	1	0	132	117	/	14	133	23	/	59	50
	2	49	0	94	/	19	163	71	/	118	154
	3	112	173	0	/	161	87	114	/	179	70
	4	154	115	77	/	8	131	7	/	23	160
	5	97	113	174	/	0	78	163	/	81	10
	6	5	119	117	/	100	0	45	/	158	7
	7	173	40	118	/	12	146	0	/	24	26
	8	43	178	19	/	137	75	63	/	110	148
	9	3	164	164	/	43	73	162	/	0	62
	10	55	58	46	/	18	109	40	/	41	0

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

Product type	*Worker*	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*	*Task 9*	*Task 10*
Product E	1	0	/	4	6	11	/	2	112	88	10
	2	120	/	26	160	150	/	126	64	32	146
	3	5	/	0	104	35	/	121	130	136	14
	4	107	/	59	0	5	/	76	77	67	0
	5	139	/	177	157	0	/	127	29	169	27
	6	61	/	48	105	18	/	112	154	47	5
	7	5	/	17	63	91	/	0	72	6	87
	8	80	/	123	111	104	/	21	0	6	163
	9	81	/	25	177	84	/	128	57	0	94
	10	129	/	20	166	63	/	26	33	50	0
Product F	1	0	130	74	/	131	80	/	129	172	3
	2	136	0	127	/	5	74	/	126	61	86
	3	148	99	0	/	116	9	/	58	97	112
	4	15	72	62	/	131	165	/	4	160	79
	5	105	68	116	/	0	30	/	44	97	36
	6	43	163	112	/	92	0	/	5	137	95
	7	5	2	23	/	161	102	/	159	80	7
	8	152	30	26	/	140	106	/	0	46	43
	9	141	133	128	/	167	18	/	141	0	14
	10	21	61	146	/	57	145	/	29	60	0

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)*	*Product type 5 (seru 5)*	*Product type 6 (seru 6)*
1	434	497	702	528	233	719
2	562	606	734	668	824	615
3	374	729	344	896	545	639
4	785	553	575	675	391	688
5	537	685	607	716	825	496
6	504	605	713	551	550	647
7	873	801	443	539	341	539
8	684	575	569	773	608	543
9	724	450	501	671	646	742
10	604	374	629	367	487	519

From Table 2, we calculate the total training cost of the worker for each worker. This is shown in Table 3. We get the worker-to-seru assignment plan as in Table 4 by allocating the workers to the respective serus depending their training costs. In Table 4 the number of '1' means the corresponding assignment exists, '0' means that the corresponding assignment does not exist.

Table 4 The worker-to-seru assignment plan

Worker	*Seru 1*	*Seru 2*	*Seru 3*	*Seru 4*	*Seru 5*	*Seru 6*
1	0	0	0	0	1	0
2	1	0	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	1	0	0	0	0	0
7	0	0	0	0	1	0
8	0	0	0	0	0	1
9	0	1	0	0	0	0
10	0	0	0	1	0	0

Table 5 The difference of total training cost for worker *i*

Worker	*Seru 3*	*Seru 4*	*Seru 2*
1	469	295	264
2	172	106	44
4	184	284	162
5	111	220	189
6	209	47	101
7	102	198	460
8	26	7	32

Table 6 The data of worker's level of skill before training the workers (β_{ni})

Workers	*Product's*					
	1	*2*	*3*	*4*	*5*	*6*
1	5	6	7	10	9	8
2	5	9	5	5	8	5
3	9	6	9	6	5	10
4	6	8	8	9	5	10
5	5	6	9	6	7	7
6	10	5	10	7	6	5
7	10	9	10	10	9	8
8	6	5	7	10	6	5
9	9	10	6	6	5	8
10	10	7	7	8	6	9

Table 4 shows that the number of workers in *serus*1 and 4 exceeds the predetermined *seru* size 2. Hence, we have $S_s = \varphi$ and $S_w = \{1, 5, 6\}$, where 1, 5 and 6 mean *seru* 1, *seru* 5 and *seru* 6, respectively. Now, we remove some workers from *serus* 1, 5 and 6 to *serus* 3, 4 and 2. According to the worker-to-seru assignment plan Table 4, from step 2 in stage 3 the difference of total training cost can be calculated and is represented in Table 5. Therefore workers are shifted to those serus which have high workloads to minimise the stress on it. Table 6 represents the skill levels of workers before they were trained and also show us where they need to be trained.

Table 7 The training cost of each task of each product type for each worker

Product type	*Worker*	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*
Product A	1	0	23	2	/	/	/	/	/
	2	45	0	13	/	/	/	/	/
	3	111	62	0	/	/	/	/	/
	4	63	43	62	/	/	/	/	/
	5	73	33	43	/	/	/	/	/
	6	26	49	97	/	/	/	/	/
	7	79	39	81	/	/	/	/	/
	8	77	65	48	/	/	/	/	/
Product B	1	0	54	76	12	32	56	90	30
	2	27	0	12	16	81	51	39	42
	3	30	46	0	43	72	86	68	56
	4	32	12	19	0	52	29	43	71
	5	96	47	32	37	0	39	21	23
	6	55	51	75	34	54	0	10	43
	7	34	65	70	27	83	8	3	102
	8	43	44	32	76	33	87	12	0
Product C	1	0	44	35	24	12	/	/	/
	2	67	0	3	49	95	/	/	/
	3	45	33	0	45	31	/	/	/
	4	43	39	18	0	87	/	/	/
	5	31	95	66	87	0	/	/	/
	6	44	42	54	87	54	/	/	/
	7	40	74	27	26	26	/	/	/
	8	72	19	66	67	39	/	/	/
Product D	1	0	49	43	12	/	/	/	/
	2	99	0	82	11	/	/	/	/
	3	54	93	0	68	/	/	/	/
	4	68	81	85	0	/	/	/	/
	5	30	5	47	29	/	/	/	/
	6	22	88	7	62	/	/	/	/
	7	57	91	81	40	/	/	/	/
	8	42	97	97	21	/	/	/	/

Table 7 The training cost of each task of each product type for each worker (continued)

Product type	*Worker*	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*
Product E	1	0	65	81	56	/	/	/	/
	2	68	0	7	31	/	/	/	/
	3	8	96	0	95	/	/	/	/
	4	73	44	88	0	/	/	/	/
	5	1	18	63	20	/	/	/	/
	6	48	37	73	25	/	/	/	/
	7	86	43	54	3	/	/	/	/
	8	91	74	94	8	/	/	/	/
Product F	1	0	59	81	23	81	98	/	/
	2	91	0	87	22	71	56	/	/
	3	1	63	0	91	60	27	/	/
	4	60	10	9	0	54	15	/	/
	5	22	57	45	100	0	90	/	/
	6	19	90	80	60	63	0	/	/
	7	36	63	12	96	9	37	/	/
	8	16	47	39	88	96	41	/	/
Product G	1	0	57	11	39	62	/	/	/
	2	51	0	31	0	26	/	/	/
	3	58	68	0	97	30	/	/	/
	4	62	22	75	48	81	/	/	/
	5	24	58	82	76	0	/	/	/
	6	28	71	55	56	54	/	/	/
	7	50	65	59	87	60	/	/	/
	8	23	34	21	1	3	/	/	/
Product H	1	0	92	15	34	74	/	/	/
	2	13	0	9	63	86	/	/	/
	3	55	55	0	84	69	/	/	/
	4	92	82	87	0	41	/	/	/
	5	12	76	37	52	0	/	/	/
	6	33	31	56	64	44	/	/	/
	7	17	93	44	41	28	/	/	/
	8	85	24	63	88	53	/	/	/

The same heuristic algorithm was also applied to 8 workers and 8 serus problem. Table 7 represents the training cost of each worker for each task of each product while Table 8 shows the consolidated training cost of a worker for a product.

Table 8 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)*	*Product type 5 (seru 5)*	*Product type 6 (seru 6)*	*Product type 7 (seru 7)*	*Product type 8 (seru 8)*
1	25	350	115	104	202	342	169	215
2	58	268	214	192	106	327	108	171
3	173	401	154	215	199	242	253	263
4	168	258	187	234	205	148	288	302
5	149	295	279	111	102	314	240	177
6	172	322	281	179	183	312	264	228
7	199	392	193	269	186	253	321	223
8	190	327	263	257	267	327	82	313

Table 9 The worker-to-seru assignment plan

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

The worker to seru assignment plan is represented in Table 9 by using the algorithm. The serus having workers more than the specified worker limits are selected and the workers are shifted to worker deficient serus. This is done by computing the difference in the training cost in workers as shown in Table 10.

Table 10 Training cost of the workers

Worker	*Seru 2*	*Seru 4*	*Seru 8*
1	325	79	190
2	210	134	113
5	193	9	75
6	150	7	56
7	206	83	37

5 Results of the computed experiments

Table 11 is the adjusted worker to seru assignment plan which is obtained after optimising the number of workers in each seru according to the predetermined size In Table 12, *T*c represents the total training cost for all workers in the corresponding *seru*, and *Dj* represents the sum of squares of deviations from mean of processing times among the workers in the corresponding *seru*. After the training of the workers is completed, there is a decrease in the processing time of workers for each product as compared to the time before they were trained. Table 13 represents the final processing times of products after the adjusted worker to seru assignment plan is prepared.

Table 11 The adjusted worker-to-seru assignment plan

Worker	*Seru 1*	*Seru 2*	*Seru 3*	*Seru 4*	*Seru 5*	*Seru 6*
1	0	0	0	0	1	0
2	1	0	0	0	0	0
3	0	0	1	0	0	0
4	0	1	0	0	0	0
5	0	0	0	0	0	1
6	1	0	0	0	0	0
7	0	0	0	0	1	0
8	0	0	0	0	0	1
9	0	1	0	0	0	0
10	0	0	0	1	0	0

Table 12 All feasible task-to-worker training plans

	Worker	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*	*Task 9*	*Task 10*	*Total cost*
Seru 1	2	0	1	0	0	1	0	0	1	1	0	326
	6	1	0	1	0	0	1	0	0	0	1	
Seru 2	4	0	1	1	1	0	0	0	1	0	0	338
	9	1	0	0	0	1	0	1	0	0	1	
Seru 3	3	1	0	1	1	1	0	1	1	1	1	344
Seru 4	10	1	1	1	0	1	1	1	0	1	1	367
Seru 5	1	1	0	1	1	1	0	0	0	0	1	109
	7	0	0	0	0	0	0	1	1	1	0	
Seru 6	5	1	0	0	0	1	1	0	0	0	0	280
	8	0	1	1	0	0	0	0	1	1	1	

Table 13 The standard processing time of each task of each product type after training the worker

Product type	*Task 1*	*Task 2*	*Task 3*	*Task 4*	*Task 5*	*Task 6*	*Task 7*	*Task 8*	*Task 9*	*Task 10*
Product A	33	38	41	/	24	39	/	38	45	10
Product B	/	13	/	16	27	14	/	34	47	41
Product C	25	22	46	/	29	/	/	/	9	/
Product D	42	/	25	12	36	21	11	40	39	34
Product E	27	32	26	33	5	28	21	17	/	39
Product F	21	32	18	12	/	30	33	38	45	20

Table 14 represents the adjusted worker to seru assignment plan for 8 workers to 8 products case.

Table 14 The adjusted worker-to-seru assignment plan

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

The experimental data used for calculation TTPT is describes below. Thus, Table 15 shows the distribution of coefficient of influencing level of doing multiple tasks for each worker. The detailed data of ε_i is given in Table 16. In Table 17 the mean skill of each worker for product type n (β_{ni}) has a range from 6 to 10 before training and 0 to 5 after training the worker. "0" indicates the worker is completely trained. The detailed data is given in Table 18.

Table 15 The parameter of the example of the multi-objective line cell conversion.

Factor	*Value*
Product type	6
Batch size	N (50, 6)
ε_i	N (0.2, 0.05)
SL_n	2.2
SC_n	1.0
T_n	1.8
η_i	10

Table 16 The coefficient of influencing level of skill for workers (ε_i)

Worker	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
ε_i	0.18	0.19	0.2	0.21	0.2	0.2	0.2	0.22	0.19	0.19

Table 17 The data distribution of worker level of skill (β_{ni})

Training of worker	*Product type*					
	1	*2*	*3*	*4*	*5*	*6*
Before	N(10, 0.1)	N(8.05, 0.1)	N(10, 0.1)	N (9.2, 0.1)	N(9.1, 0.1)	N(10, 0.1)
After	N(1, 0.1)	N(2.05, 0.1)	N(1.15, 0.1)	N (3.2, 0.1)	N(1.1, 0.1)	N(2.5, 0.1)

Table 18 The data of worker's level of skill after training the workers (β_{ni})

Workers	*Product's*					
	1	*2*	*3*	*4*	*5*	*6*
1	1	4	3	2	3	4
2	1	4	2	0	0	3
3	0	2	4	3	3	4
4	4	1	3	1	3	4
5	0	4	0	2	3	1
6	3	2	3	4	3	2
7	1	1	1	4	3	1
8	4	4	0	2	0	0
9	4	3	3	4	4	1
10	0	3	0	1	4	1

Figure 1 Effect on TTPT before and after training of workers (see online version for colours)

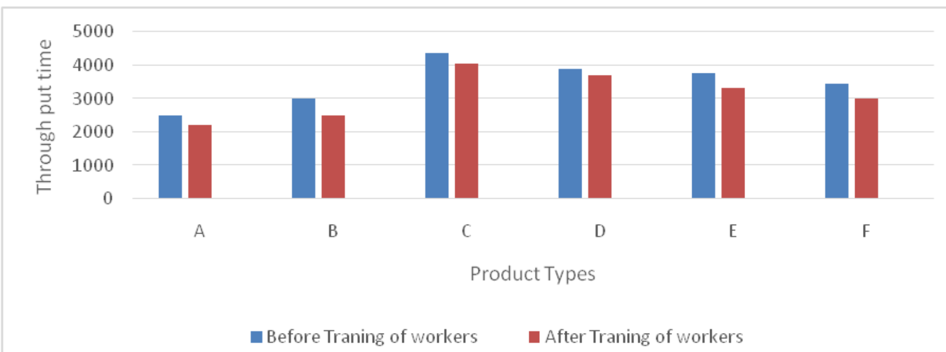
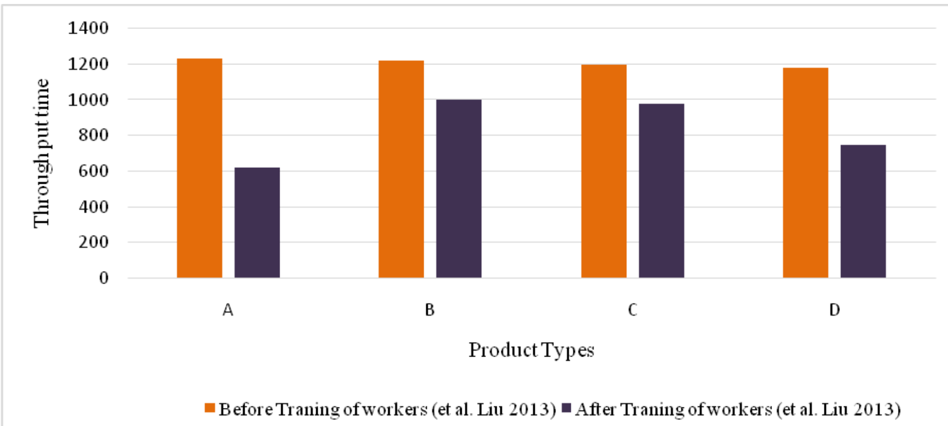


Table 19 Through time and computational for the proposed problem

	Throughput		*Computational time*	
Product type	*Before training of workers for proposed problem*	*After training of workers for proposed problem*	*Before training of workers (sec) proposed problem*	*After Training of workers (sec) proposed problem*
A	2,500	2,200	150	132
B	3,000	2,500	180	150
C	4,353	4,043	272.813	242.58
D	3,879	3,714	232.74	222.84
E	3,780	3,336	226.8	200.16
F	3,447	2,998	206.82	179.88

From Figure 1, it is observed that reduction is the number of workers lead to decrease in TTPT. However, in the line-cell conversion, with the increase in the skill levels of workers, there is chance to find multiple optimal solutions. Moreover, the performance of worker(s) is also affected by other operating factors, such as workers' skill level, batches and lot sizes, however, the factors that have higher influence are not known. Clarifying such relationship and information on how to format cell and load cell are key issues in success of full line cell conversion towards reducing worker(s) and TTPT. In Table 19, the throughput time and computational time for completing the products have been shown. It is observed that the throughput time has decreased after training of the workers due to their increase in skill level, simultaneously the computational time for conducting the analysis also decreased substantially.

Figure 2 Effect on TTPT before and after training of workers (see online version for colours)



Source: Liu et al. (2013)

To prove the effectiveness and efficiency of the proposed model, we have also obtained the data sets from Liu et al. (2013) and then tailored the data sets as 4 jobs 6 workers case by maintaining the same parameters as mentioned in the above experimentation for further analysis. Figure 2 shows that the results obtained after conducting the experimentation follows the same pattern where workers' skills after training have been significantly improved and it further lead to improvement in their process and performance. Here, for all the 4 jobs and 6 workers the throughput time has significantly decreased with trained workers. In Table 20 analysis of a similar experimentation conducted are presented, where similar pattern proves the model validation. In Table 21 by varying different parameters i.e., jobs, workers and number of seru's with different instances the sensitivity analysis is performed. Initially, for four different scenarios by keeping the total number of workers as constant and by varying the workers and jobs the analysis is performed. It has been observed that though the number of jobs increasing for these instances the performance measures such as through put time, total cost and computational time shown no difference. Similar kind trend observed for instances 5, 6, 7 and 8 even after increasing the seru's. Similar analysis has done for instances 9, 10, 11 and 12 where the number of workers are increased from 10 to 12 by keeping the serus as constant 6 and by varying the products incrementally. It has been observed from these instances that the performance measures shown their poor performance. This trend clearly shows the decrease in serus impact the performance of the system than the increase of workers. One can observe these trends highlighted with red and green boxes shown in Table 21.

Table 20 Through time and computational for the problem

	Throughput		*Computational time*	
Product type	*Before training of workers (Liu et al., 2013)*	*After training of worker (Liu et al., 2013)*	*Before training of workers (sec (Liu et al., 2013)*	*After training of workers (sec) (Liu et al., 2013)*
A	1,230.5	621.5	73.83	37.29
B	1,222.5	1,003	73.35	60.18
C	1,198	981.5	71.88	58.89
D	1,182	746.5	70.92	44.79

Source: Liu et al. (2013)

Table 21 Sensitivity analysis (see online version for colours)

Parameters / *Scenarios*	*Workers*	*Products*	*Seru's*	*Through put time*	*Total cost*	*CPU time(sec)*
1	10	4	8	187.91	3,131	294
2	10	5	8	188.28	3,131	294
3	10	6	8	190.16	3,133	294
4	10	7	8	191.19	3,133	294.2
5	10	4	10	187.06	3,131	294
6	10	5	10	188.12	3,131	294
7	10	6	10	190.20	3,133	294
8	10	7	10	191.18	3,133	294.2
9	12	4	6	192	3,134	295
10	12	5	6	197	3,139	295
11	12	6	6	203	3,145	296
12	12	7	6	210	3,152	297
13	12	4	7	192	3,134	295
14	12	5	7	194	3,136	295
15	12	6	7	198	3,138	295
16	12	7	7	200	3,140	296
17	15	4	8	185.02	3,125	291
18	15	5	8	186	3,126	291
19	15	6	8	190	3,130	294
20	15	7	8	190.6	3,130	294
21	15	4	10	187	3,131	294
22	15	5	10	188	3,131	295
23	15	6	10	189	3,132	294
24	15	7	10	190	3,133	294

6 Conclusions and future work

This paper mainly focused on two important objectives, primarily, in phase one making the workers multi-skilled by training them for aiming to reduce the total training cost and secondarily, minimising the processing time of the workers. In line of the above objectives a mathematical model has been developed. However, based on the processing times of the workers, number of operations and other related data to develop the worker to seru assignment plans a heuristic algorithm has been proposed. In second phase, one of the important performance measure i.e., minimisation of total throughput time of workers is considered to find the performance of seru systems. This reduces the labour force that has been detailed in the second phase. Later, with several instances the proposed model has been tested and found from the results that training of workers into multi-skilled and then implemented their skill in seru systems helps in improving the performance of the system. In addition, sensitivity analysis also conducted by varying the products, workers and seru systems to validate the proposed model performance. It has been observed from the results that reduce in number of seru's by parallel increasing the jobs leads to poor performance of the system. Consequently, though workers skill level improved their performance will satisfy if the number of jobs is less in number. In future work, the sustainability parameters can be identified and with different heuristics, meta-heuristics these problems may be tested.

References

Yin, Y., Stecke, K.E. and Kaku, I. (2008) 'The evolution of seru production systems through out Canon', *Operations Management Education Review*, Vol. 2, pp.27–40.

Shinobu (2003) *Post-Lean Production Systems:Towards an Adaptable Enterprise in the Age of Uncertainity*, Bunshin-Do,Tokyo.

Endou, S. (2004) 'Construction of *seru* production system and improvement of performance', *IE Review*, Vol.45, No. 2, pp.22–28.

Kaku, I. (2017) 'Is seru a sustainable manufacturing system?', *Procedia Manufacturing*, Vol. 8, pp.723–730.

Kaku, I., Gong, J., Tang, J. and Yin, Y. (2009) 'Modeling and numerical analysis of line-cell conversion problems', *International Journal of Production Research*, Vol. 47, No. 8, pp.2055–2078.

Liu, C., Lian, J., Yin, Y. and Li, W. (2010) 'Seru seisan-an innovation of production management mode in Japan', *Asian Journal of Technoiogical Innovation*, Vol. 18, No. 2, pp.89–113.

Liu, C.G., Dang, F., Li, W.J., Lian, J., Evans, S. and Yin, Y. (2015) 'Production planning of multi-stage multi-option seru production systems with sustainable measures', *Journal of Cleaner Production*, Vol. 105, No. 1, pp.285–299.

Liu, C.G., Yang, N., Li, W.J., Lian, J., Evans, S. and Yin, Y. (2013) 'Training and assignment of multi-skilled workers for implementing *seru* production systems', *International Journal of Advanced Manufacturing Technologies,* Vol. 69, Nos. 5–8, pp.937–959.

Shao, L., Zhang, Z. and Yin, Y. (2016a) 'A bi-objective combination optimisation model for line-seru based on queuing theory', *International Journal of Manufacturing Research*, Vol. 11, No. 4, pp.322–338.

Shao, L., Zhang, Z. and Yin, Y. (2016b) 'Production system performance improvement by assembly line-seruconversion', *Proceedings of the Tenth International Conference on Management Science and Engineering Management*, Baku, Azerbaijan, 30 August to 2 September.

Shimizu, K. (1995) 'Humanization of the production system and work at Toyota Motor Co. and Toyota Motor Kyushu, In Enriching Production', *Perspectives on Volvo's Uddevalla Plant as an Alternative to Lean Production*, Vol. 4, pp.383–403.

Stecke, K.E., Yin, Y., Ikaku, I. and Murase, Y. (2012) 'Seru: the organisational extension of JIT for a super talent factory', *International Journal of Strategic Decisions Sci.*, Vol. 3, No. 1, pp.106–119.

Xiao, Z.L., Liu, C.G., Li, W.J., Evans, S. and Yin, Y. (2017) 'Effects of key enabling technologies for seru production on sustainable performance', *Omega*, Vol. 66, No. 1, pp.290–307.

Ying, K.C. and Tsai, Y.J. (2017) 'Minimising total cost for training and assigning multiskilled workers in seru production systems', *International Journal of Production Research*, Vol. 55, No. 10, pp.2978–2989.

Yong, Y., Stecke, K., Swink, M. and Kaku, I. (2017) 'Lessons from Seru production on manufacturing competitively in a high cost environment', *Journal of Operations Management*, Vols. 49–51, No. 1, pp.67–76.