Management and Production Engineering Review
Volume 12 • Number 4 • December 2021 • pp. 53–70
DOI: 10.24425/mper.2021.139995

Seru Production System: A Review and Projections for Future Research

Yung Chin SHIH

Federal University of Technology, Dept of Production Engineering, Brazil

Received: 07 November 2020
Accepted: 07 October 2021

Abstract
In mid-1992, Japanese consultant Yamada Hitoshi was tasked with modifying the production systems of Japanese companies as the existing configurations at manufacturing plants no longer satisfied unstable demands. He made improvements to the overall production system by dividing the long assembly lines into several short ones called cells or *seru*. Although of the advantages, it is still unclear about how to manage this new production system, and what variables really promoted the desired benefits. We identify in total 39 articles from 2004–2020 about the progress of the *seru* production system, and we observe some possibilities to improve the effectiveness of this type of the production system. The first is the possibility of manufacturing the product in flexible sequence, in which the operations are independent among them. We show through the developed example that the makespan may be different. We noted when converting the in-line production system to one pure *seru*, the makespan tend to increase. Nevertheless, when analyzing the effectiveness of *serus* working concomitantly considering splitting the same lot, makespan and the cost may be reduced. And finally, when converting to one of pure *serus*, the performance may be similar to that obtained when *serus* working concomitantly.

Keywords
Seru production systems, Flexible and reconfigurable manufacturing systems, Production activitycontrol, Assembly and disassembly, Manufacturing plant control.

Introduction

In mid-1992, Taiichi Ohno's consultant Yamada Hitoshi was tasked with innovating the production systems of companies such as Sony and Canon as the configurations in their manufacturing plants – with long assembly lines moved by conveyors – no longer satisfied the new demand profiles characterized by low volumes, large variations, and high product varieties. Products with such profiles include electronic devices, such as digital cameras and cell phones. Hitoshi divided the assembly lines into smaller lines called the cell, or *seru* in Japanese. This approach improved the productivity, reduced the production costs, and decreased the idling of operators, to name a few advantages. As per Kaku (2016), although of the advantages, it is still unclear about how to manage this new production system, and what variables really promoted the desired benefits.

According to Yu & Tang (2019), the typical variables involved in the implementation of this system include layout, type of *seru*, variety of products, objective to be analyzed, production volume, relationship among employees, difference between skills, and batch dispatching rules. The typical objective functions to minimize that are: makespan, workload attributed to employees, number of employees, total cost, and delay minimization.

Due to the complexity of involving these variables, many of them are omitted or simplified. This article aims to review the literature discussing this type of production system, the difference with the traditional cell, the classifications, the factors involved in the design, the performance indicators, the developed algorithms and future directions.

The rest of the article is structured as follows: the first heading explains the methodology. The second heading summarizes traditional assembly systems, explaining the motivation of companies to conceptualize a new production system. In sequence, the configurations of *serus* are presented. The third section

---

***Corresponding author**: Yung Chin Shih – Federal University of Technology, Dept of Production Engineering, Brazil, phone: +36 465 65 111/17-56, e-mail: chin@utfpr.edu.br*

presents a survey on the experiences of authors and companies that implemented *seru*. The importance of operator flexibility and desired *seru* configurations when handling expensive equipment are discussed. Finally, we present examples to show some possibilities to improve the performance of *seru* production system beginning with the traditional convertion in-line production into pure *seru* production, following by considering *serus* working concomitantly. Next, it is incorporated in the example analysis the rigid and flexible production sequence. The splitting the lot is also analyzed. Finally, since the level of skill is defined in the literature, the last example illustrates the situation when the skills are subjected to variation (and consequently affecting the duration of operation).

Research methodology

We perform exploratory research, in which a specific subject is investigated, the technical procedure involves bibliographic research, and the data acquired corresponds to the number of published papers.

For research on *seru* production systems, an open-access noncommercial database was used to retrieve the most relevant and cited articles of commercial publishers. The rest of the articles were obtained directly from the authors through personal request.

Fig. 1 illustrates the number of locally published articles since 2004, totaling 39 articles until 2020.

The literature on *seru* production systems is not exhaustive. The total number of locally published articles on this topic is approximately equal to that found by Yu & Tang (2019), i.e., 30 articles. Despite these limitations, the reviewed articles help provide an overview of the approaches to this subject.

Table A.1 summarizes the classifications of published articles in relation to the general variable decision involved in the development and correct functioning of *seru* production systems.

Background

To understand *seru*, it is important to introduce in-line production systems and the Toyota Production system, and the motivation behind the proposal of a new production system.

In in-line production – heavily used in North America – workers in the line are tasked to execute one or many operations. The principle of this system is based on the division of labor, documented by Adam Smith, with the objective of achieving low cost per unit. This approach demands a high reproduction rate, has a low diversity of product yield, and other reported problems such as repetitive strain injury, removal of the decision-making power of the operators, etc. The tooling installation (Womack et al., 2004) for this approach takes 2 to 3 days. This shows the challenge in achieving the desired "flexibility" with this method. Therefore, workers are assigned simple activities. Some companies in the West have adopted this system with conveyors to support the movement of bulky parts, as in the case of automotive components.

According to Ohno (1997), Kiichiro Toyoda could not replicate the North American production system in the Japanese post-war scenario where all waste must be conserved due to the scarcity of raw materials.

The just-in-time system (JIT) in Japan was aimed at reducing the need for space for material inventory. According to Womack & Jones (2004), to achieve

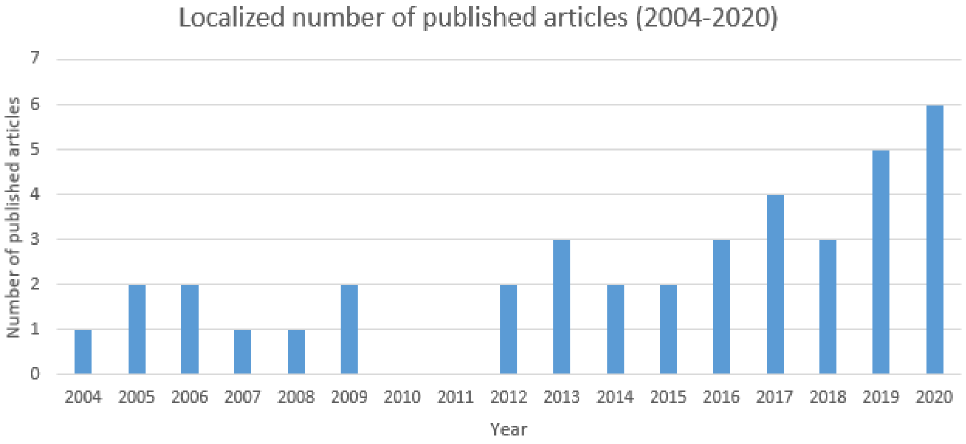

Fig. 1. Number of published articles (2004–2020)

JIT, companies must produce at the right time and in the right quantity, i.e., only when there is a definitive order.

According to Chaves Filho (2010), the JIT had three key operational elements: the pulled system, the takt time, and the continuous flow. In other words, occasionally, the end customer "pulls" back the product according to the quantity in order. The final products must be replaced by communicating the previous steps to deliver/produce the demanded quantity. This process, which is a repeated cycle, is called continuous flow.

The lean production system is an extension of the JIT system from the value perspective, i.e., the amount of money the customer is willing to spend by eliminating unnecessary components or activities.

Seru production system

Although Toyota's production system is an overall efficient management system, it is inefficient in unstable scenarios, e.g., electronics (Villa & Taurino, 2013). In this sector, the product life cycle is very short, with uncertain demands and large fluctuations in production volume. The meaning of production fluctuation in Villa & Taurino (2013) is vague. We assumed that it might be referring to scenarios where production may be interrupted, such as unexpected machine breakdown, unprepared operators working below their normal pace, unknown product profile that requires additional operator training, etc.

According to Shah & Ward (2007), to be able to reach a greater mix of products and low volume, companies must show swift response to the market with rapid adaptation.

Subsequently, a new production system emerged in Japan in the mid-1990s, with *serus* instead of long conveyors.

The expression "*seru* production" is relatively new. The difference between traditional cellular manufacturing and cell production is not clear. For example, Hasegawa & Fukuta (2009) stated that, similar to the traditional cell, employees work in an upright position within each *seru*. According to Zwierzyński & Ahmad (2018), a *seru* production concerns handling of employee skills. Each operator in a *seru* must have the competency to work in different *serus*. Miyake (2006) and Sakazume (2005) conducted a comparative study between the traditional cellular physical arrangement and *seru*. However, Kaku (2016) showed that the comparison was not valid both the systems used similar technologies. However, Kaku (2016) did not specify the type of technology, whether it is related to equipment or information technology, or the work philosophy (the production technique itself).

Machines that compose a classical cell are not very mobile. The principle of a *seru* emphasizes the necessity of human resources and the simplicity of the mechanism to move the machines (Yin et al., 2017).

Villa & Taurino (2013) made a more precise observation regarding this, when declared handling light products, as seen in the electronics industries, does not justify the use of large conveyors. The *seru* system was conceived precisely to offer greater production flexibility through training and transport of employees in workstations.

According to Kaku (2016), *seru* also has a typical *U* shape, favoring the: a) rational movement of materials in the process (unitary flow); b) execution of activities by multifunctional workers. This suggests that further configurations of the *seru* production system need not be explored.

Merits and demerits of *seru* production system

As per Kaku (2017), Canon estimated that the implementation of *serus* could reduce CO_2 emissions by more than 50%. In 2001, Ricoh reduced its emissions by approximately 14%.

As per Kaku (2017), *serus* could help reduce the energy consumed by production owing to the lack of conveyors. This reduction, according to Liu et al. (2015), was equivalent to a reduction in the CO_2 emission. Nevertheless, this statement does not hold true for countries that already utilize abundant renewable energy sources. In this case, is the implementation of *seru* production system truly beneficial?

To address this speculation, Kaku (2016) proposed a method that unified the experiences of companies that implemented the system and the benefits observed by most of them. The author named it the "S–F scheme" and exemplified the grouping method for a more simplified scenario, analyzing the responses of 7 companies using 26 keywords. For instance, if there is a 57% similarity in productivity among the companies, then 57% of the companies observed a gain in productivity after implementing *seru* production. The author claimed choosing such keywords is not an easy task because it involved cultural issues. However, companies may not be able to avail the full benefits of *serus* due to the lack of knowledge about them (Liu et al., 2014). Kaku (2017) claimed that the *seru* production system was not suitable for all types of products. Another study (Sengupta & Jacobs, 2004) applied the *seru* system for assembling televisions and showed that assembly lines still outperformed the *seru* system. Therefore, the factors and scenarios that warrant the implementation of *serus* must be understood. Kaku (2017) observed that the approach of designers had shifted to developing smaller, lighter, more compact products.

Types of *seru* production systems

According to Yu & Tang (2019), the types of *serus* are – divisional *seru*: where each cell consists of more than one operator, each of who performs one or more operations; rotational *seru*: each cell has one or many operators, but the operators perform all the necessary operations on the product. The level of qualification and skill required in this type is higher than in the divisional *seru*; and *Yatai*: only one employee works in one *seru* and he/she is responsible for all product operations.

According to Beber (2019), a *Yatai* is suitable for products that require difficult and high-precision techniques, and highly proficient operators. For a rotational *seru*, this format performs well in environments with a certain variety of production volumes, moderately complex product structure, and fully trained operators who have approximately the same levels of proficiency. Divisional *serus* are generally adopted in the initial stage, with a low level of multifunctional operator training.

Additionally, in the case of divisional *seru*, it is important to note that the operator may execute the same activity in other *serus*.

Fig. 2 illustrates a typical example where a traditional assembly line composed of seven employees is divided into three *serus*. Each *seru* has a *U* shape. The first *seru* on the left has three employees, the first being responsible for operations 1 and 2, the second responsible for operations 3, 4, and 5, and the last for operations 6 and 7. Note that the employees are not responsible for all operations. Given these features, we classified this line under the divisional type. The second *seru* has the same configuration as the first. Although not represented, each employee is responsible for all product operations. This is the rotational type *seru*. In the third *seru*, only employee performs all the operations. This comes under the *Yatai* type.

Good practices for correct functioning of *seru* production systems

As per Stecke et al. (2012), each *seru* has its own characteristics, which may be broadly divided into three based on the literature: 1) *Kanketsu*: All operations are performed within the *seru* without the need to complete the remaining product operations in another *seru*; 2) *Majime*: the position of the equipment; 3) *Jiritsu*: the ability of *seru* employees to learn, organize, and evolve. As a prerequisite for *Jiritsu*, the employees must be trained according to the needs of production (Liu et al., 2014) at a steady pace. Yin et al. (2017) reinforced this point by stating that employees must be provided with autonomy, self-management, learning, and self-improvement.

This characteristic was inherited from total quality management, in which the employee is given the

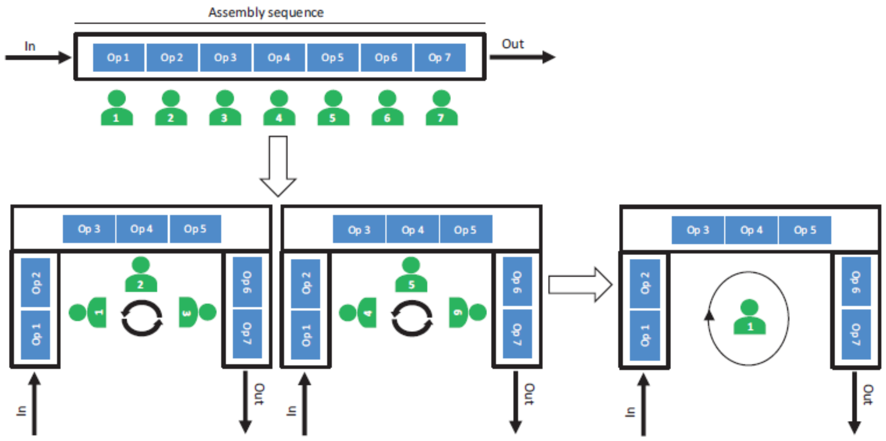

Fig. 2. Conversion of the traditional line into three *serus* (Source: Zwierzyński & Ahmad (2018)). Although apparently the authors were wrong in representing the last *seru* with the operator 1 instead of 7, it may be interpreted that the authors intended to replace the entire assembly line with 7 operators with a *Yatai*-type composed by only one operator

freedom to decide, learn, and define the best course of action to execute operations without relying on a centralized form of management.

According to Yu & Tang (2019), a *seru* can produce not only one but several types of products, which is called parallel production. In addition to employee skills to manage, machines may require changing the position on the floor and also, calibration of them. They also proposed the idea of a hybrid production system, i.e., a *seru* but combined with small assembly lines. Yu et al. (2014) claimed that the hybrid type was more appropriate in a scenario with expensive equipment and simple operations, requiring little to no skills.

Yu et al. (2017) showed that most studies covered pure *serus*. They also claimed that it was more practical to implement a hybrid system with pure *serus* and a short assembly line. In this case, the expensive equipment are installed in the line instead of the *serus*. The operator may not always have the skills required to perform all operations or low-complexity operations. In this case, it is also convenient to position him/her in the line.

Van der Zee & Gaalman (2006) observed that the performance of the entire production system depended on the positioning of the small assembly line, either before or after the *serus*. As per Yu & Tang (2019), the optimization of a hybrid *seru* production system is significantly more complex. Therefore, most research studies around this topic are focused on pure *serus*.

Mathematical models and algorithms

In this section, we discussed the main published studies from the 39 articles identified in the literature and problem formulation.

Liu et al. (2015) proposed a mathematical model to minimize CO_2 emissions and the makespan. The model can choose the most appropriate *seru* and allocate the part for assembly. As per the authors, in problems involving multiple objectives, all the objectives must be satisfied. The conventional method to satisfy all the objectives is to assign a weight to each objective and transform them into a single function, ignoring the differences between the units of measurement. *Fast nondominated sorting* was proposed to overcome this difficulty.

As per Ghosh & Das (2008), multiple objective functions are usually conflicting, i.e., the solution obtained from one may not be beneficial to the other. Because one function is not more important than the other, a set of solutions that benefits all objective functions is identified. Once all feasible solutions have been plotted on a graph, it would be possible to identify a boundary (limit). Therefore, the solutions are called "nondominated solutions" (for one objective function not to dominate the other). This set of solutions is also known as the Pareto optimal frontier. The *fast nondominated sorting* algorithm works as a type of layer or frontier of Pareto, with the first boundary being formed by nondominated solutions and the second boundary from solutions dominated by the first layer. This approach is aimed to direct the search to obtaining better solutions with less effort. However, the authors did not consider that the batch might be dispatched (divided) to several *serus*.

Yu et al. (2013) proposed a mathematical model to convert the line into *serus*, which determined the appropriate number of *serus*, minimized the makespan and the number of employees, as well as assigned employees to cells. The algorithm to solve the model was improved from the exact algorithm, aiming to obtain the Pareto optimal solutions. Research studies including Yu et al. (2012), Yu et al. (2013), and Yu et al. (2014) used the first-in-first-out (FIFO) approach to assign "lots" to *serus*. As per Yu et al. (2016), the batch dispatching rules for cells were of the NP-hard (nondeterministic polynomial time) type.

According to Yu et al. (2016), these rules must be investigated, thus justifying the contribution of their study, which they considered in addition to FIFO and 9 dispatching rules. The authors also proposed two exact improved algorithms to obtain the shortest total production time and the lowest workload attributed to the operators. They considered N types of products and M lots, and assumed that all products had the same sequence of operations. This led us to believe that, from an operational perspective, the differences among the durations of operations times distinguish the products from one another (although the authors did not make this evident).

Studies have demonstrated that the formation of production *serus* is also an *NP*-hard problem (Yu et al., 2013). Yilmaz (2020a) concurred with existing research opinion that the operator was central to the *seru* production system as he/she helped the system achieved the desired flexibility. The author proposed a mathematical model that minimized two objective functions, makespan and workload balance between collaborators working in each *seru*. To this end, he suggested using a nondominated-II genetic classification algorithm (NSGA-II). Although the author did not explain it, NSGA-II shares certain characteristics with the traditional GA. The steps of the GA are: 1) Randomly generate individuals from a population; 2) Calculate the fitness (objective function); 3) Choose the most suitable individuals for a crossover; 4) Apply the mutation to the individual child; and 5) Repeat the procedure to obtain the population of an individual child.

The objective function is not a solution, but a result of the solution. The solutions obtained by the "child" individuals generate an unbound frontier called the first dominance frontier. NSGA-II differs from the traditional one in that it classifies solutions in relation to borders, aiming, in essence, to direct the search and thus produce the best solution in a short time. To this end, it is required to 1) exclude the nondominated solutions, where only a type of agglomeration of solutions from the individual child remains, which, in turn, generate another frontier of dominance, and 2) verify from the remaining solutions the proximity of the nondominated solutions to the second frontier. The farther they are, the worse the solution, and the less likely the corresponding individual child is chosen for crossover to the next generation. For small problems, the author suggested the increased ε-restricted method (AUGMECON2), which is particularly important for problems with several objective functions. One way to solve these problems is to maintain only one of the objective functions, while transforming the remaining into restrictions. These restrictions must be less than the threshold (Filho & Florentino, 2018).

Yu et al. (2017) proposed a mathematical model that aimed to establish a hybrid production system. They further suggested decomposing the hybrid system into simpler systems to obtain the optimal solution. They considered the skill level required for each type of product. The skill indices were defined a priori. They did not consider the influence of the indices on the duration of each operation, because, as stated by Anzanello & Fogliatto (2007), the greater the number of units produced, the more the training, resulting in a shorter operation time. This limitation warrants further investigation. In addition, in Yu et al. (2017), the amplitudes of the established indices were not established, i.e., whether a higher index meant a higher number of skills or vice versa, or if these indices must be updated over time. Moreover, the authors considered practically fixed lots (approximately 50 units), which seemingly contradicted the purpose of a *seru* production system, i.e., to deal with high variations in demand. Finally, they realized that, for small problems with no more than eight operators, the exact algorithm itself provided an optimal solution.

Johnson (2005) analyzed the performance of an assembly line after conversion into two *serus* in terms of the total production time and required number of hours of work. To conduct the study, he assumed that the batch (lot) with empty *serus* was assigned the workload upon arrival. If all cells were occupied, the workload would be assigned to the one that could finish the batch the fastest.

Zwierzyński & Ahmad (2018) conducted a comparative performance study between the traditional line and the *serus* through modeling and simulation. The benefits include reduced production costs, increased number of finished products, and reduced waiting time. However, the authors did not explain the meaning of waiting time. It is not clear if it is related to a part or to a collaborator. The authors also did not specify how they determined the number of employees, how the workforce was distributed in *serus*, and how operations were assigned to operators.

Wang et al. (2020) observed that, in practice, *serus* production systems were limited by the production capacity. They must be able to "choose" products appropriately based on the restrictions of operating time, setup time, delivery date, revenue, and penalty for delay. To assist in this difficult decision-making process, the authors proposed a model combining matrix crossover and GA (MCGA).

Another study (Gai et al., 2020) sought to minimize the makespan through the analysis of batch distribution in different *serus*. The study did not consider two crucial limitations: different *serus* cannot be used to concurrently execute the same batch; and the order in which the batches are dispatched to the cells. To help solve the problem, the authors developed an exact dimensional reduction algorithm for testing batch allocation to possible cells. Techniques for grouping information of variables were needed to reduce the computational effort without affecting the quality of the result. The authors did not explicitly reveal the clustering techniques they used. They did compare the exact algorithm proposed with the greedy algorithm and realized that the developed algorithm was remarkably robust in most situations. The greedy algorithm performed better when the production efficiency was considerably varied.

Examples of variations in production include introducing to the *serus* a new operator unfamiliar with the operation or a new type of product to the production line. The authors did not specify the type of *seru* they were addressing; however, they affirmed that the proposed algorithm could be extended to the rotational *seru* line. Particularly, unique to Gai et al. (2020), the algorithm would re-analyze the best *seru* option for the remaining batch quantity. As a mathematical proof, they demonstrated that the possible number of *serus* was in the order of O(n2), where n corresponds to the number of *serus*.

Deepthi et al. (2019) considered three objective functions: a) minimization of training costs; b) minimization of makespan in assembling the products; c) and minimization of carbon emissions. The heuristics were operated as follows: choose the *seru*, which is based on the one with the lowest training cost; and balance the employees. If the cell was overloaded, the employee was assigned to an emptier cell.

Wang & Tang (2018) developed a mathematical model aimed at developing a *seru* production system with a large capacity and respond to the stochastic demand, minimizing cost, and maximizing service level. The authors did not specify the type of cost, i.e., cost to form *serus* cells to meet the stochastic demand or operational cost as soon as the *serus* were formed. The authors improved NSGA-II for assistance in solving the model.

Another study (Ayough et al., 2020) proposed a nonlinear integer programming model for the line-to-*seru* conversion problem by analyzing the production schedule. The authors proposed the use of an invasive weed optimization algorithm. According to Josiński et al. (2014), the mechanism of this algorithm was very similar to the GA: 1) Randomly generate the population of individual parents; 2) Crossover these individuals according to their fitness (i.e., the objective function); 3) From the generated children, conduct random spatial dispersion, similar to that in the mutation.; 4) Carry out exclusion (competition) among individual children.

To explore the hybrid system, studies have developed simplified mathematical models (Yu et al., 2014), (Sun et al., 2016). For instance, Sun et al. (2019) developed a nonlinear mathematical model to minimize the total delay time, considering the formation and scheduling of *serus*. Considering the complexity of nonlinearity, the authors suggested to decompose the nonlinear models into linear ones. With simplified models, they could obtain the exact minimization of the desired indicator. This decomposition is called coevolution–cooperation. The authors proposed two co-evolutionary–cooperative algorithms: one which addressed the problem of *seru* formation, and another the problem of production scheduling.

The model developed in Kaku et al. (2009) considered the three types of *serus* and several other factors; however, according to Yu et al. (2012), the tradeoff relationship among the factors was not yet analyzed. Thus, they simplified Kaku et al. (2009)'s model and applied it to the development of rotational and Yatai *serus*, solved using the nondominated genetic classification algorithm-II. The *seru* was chosen based on the level of occupancy. If a *seru* was available, then a lot was assigned. Otherwise, it was assigned to that which could complete the batch earlier. In the modeling formulation, the authors considered the relationship between the operator's ability and the time he/she would take to perform the operation. They assumed that the learning curve resembled a V shape. However, they did not explain the procedure used to define the curve or the influence of the number of units produced on the production time.

According to Han et al. (2020), assembly lines rarely reach 100% of their production capacity. Thus, the authors presented a model aiming to establish *serus* by assigning machines to each of them, analyzing the problem of the stochastic capacity of each machine.

A study (Han et al., 2018) investigated the capacities of the operators, and concluded with justification that, in practice, the performances of employees were affected by their emotional state, absence from work, and working environment.

Liu et al. (2013) analyzed the employee assignment to *serus* in order to balance the workload. To this end, they proposed a mathematical model for the scenario in which an assembly line was completely transformed into *serus*. This model, in addition to addressing the problem of workload balance, minimized the cost of training through a proposed heuristics.

Another study (Gong et al., 2009) developed a model for the hybrid system to minimize the inventory. Although they considered inventory minimization important, it is a controversial objective. The *seru* production system does not have the characteristic of a push system, let alone forming intermediate inventory, because it follows unitary flow and the proximity between equipment prevents any type of accumulation.

Liu et al. (2015) conducted performance simulations considering four types of products, and each type had several *serus* where the product could be processed. Finally, they presented the performance including in the analysis 10 types of products. Although *serus* were proposed to manage diversity (flexibility) of products, the authors tested a maximum of 10 products. This appears to be inadequate because this type of production system operates based on the premise of a diverse range of products.

In a first, Yilmaz (2020b) studied the assignment of the workforce to different *serus* with the objective of minimizing the makespan by scheduling the operations of the employees. This issue of attribution is called *Shojinka* in Japanese. Studies before Yilmaz (2020b) assumed that the operator must remain in the same *seru* where he/she started the operation, and he/she must continue to perform the operations of the same product until the end of the assembly.

The author did not specifically address the training itself, assuming in his work, a priori that the worker had already received the expected training and can, without restrictions, be allocated to work in different *serus*.

Mouzon et al. (2007) highlighted a historical survey of a certain company, which reported that, on an 8-hours workday, a machine might consume energy while idling for 16% of the time. Based on this premise, they presented a mathematical model to minimize the energy consumption, considering the batch dispatching rules. With this model, it was possible to determine the machines that remained underused for the longest time, the required shutdown period, and the allowable "stand-by" time.

Further, Mouzon & Yildirim (2008) developed a model to minimize the electricity consumption as well as the delay.

Manupati et al. (2015) analyzed the possibility of implementing a *seru* production system from the perspective of Indian labor. They aimed to propose a model for optimizing a single objective function that minimized the cost of operator training.

Another study (Ren & Wang, 2019) pointed out the lack of studies from a customer's perspective. Subsequently, they proposed a method to determine the time of receiving products. To compare the traditional line and the *serus*, they used the $\mathrm{M}^x/\mathrm{G}/1$ queuing model, proposed by Briere & Chaudhry (1988), which, according to Chaudhry (1979), the exact result could be readily obtained without major difficulties. They added that, according to Little's law, the length of the queue had a strong positive correlation with the waiting time in that line. This type of model, according to Chaudhry (1979), was characterized as a Poisson arrival from entities in size X group, with service times of the generic type and single server. For the assembly line with m operations, according to the authors, the "temporary (or sojourn)" waiting time for a product had two time compositions. The first was the time at which the batches preceding the batch of the analyzed part ended, and the second was the one at which the batch of the analyzed part ended. The authors deduced the effective waiting time, and consequently the length of the queue (1):

$$Lq(A) = \frac{\lambda k(k+1)}{2(\mu - \lambda k)} - \frac{m\lambda k}{\mu}\,, \tag{1}$$

where: k represents the average lot size, A represents the assembly line.

The authors analyzed *Yatais*-type *serus*, with n *serus* and 1 collaborator for each *seru*. The service time of each *seru* behaved as a negative exponential. They stated that the suitable model applicable to this case would be $\mathrm{M}^x/\mathrm{M}/\mathrm{n}$ but claimed that it did not exist in the literature. According to them, the M/M/n model was similar to the first $\mathrm{M}^x/\mathrm{M}/\mathrm{n}$. Therefore, they chose the first model to deduce the second. The analyzed scenario included n *serus*, each with a negative-exponential service time. As soon as any batch k arrived at n *serus*, according to the Poisson distribution λk, and Xr batches in the system, the authors demonstrated mathematically that the output of the products in each *serus* behaved like a Poisson distribution. They added that, if the *seru* production system was in equilibrium, with all *serus* occupied, in addition to overlaps in the n Poisson processes of product output, which the authors called μ', it behaved like a Markov chain. The length of the queue was determined from this exit behavior:

$$Lq(C) = \frac{C\lambda^2 k^2}{(n\mu' - \lambda k)^2}\,, \tag{2}$$

where: Lq is the length of the queue and C is a constant that varies between [0, 1].

Hence, the difference in performance between the production systems was $\Delta Lq = Lq(A) - Lq(C)$.

Ying & Tsai (2017) presented a mathematical model aimed at minimizing the cost of training the employees to perform multiple tasks, from the perspective of balancing the assigned workloads, in the divisional or *Yatai serus*. To solve the model, the authors proposed an algorithm called the SAIG, which was composed of two phases. With the aim of efficiently allocating employees to *serus*, this phase was described as follows: 1) generate the first random solution by choosing employees and allocating them to the *seru*. If it produces better results than the conventional system, there was a 50% chance that the solution would be chosen again in the next iteration; 2) improve the solution through the *serus* employee exchange and insertion mechanism. To this end, a pair of *serus* was randomly chosen, and in each pair a collaborator to be exchanged was chosen. Once chosen, he/she was removed from the *seru* to which he belongs, along with the other chosen one; 3) finalize the algorithm, which was performed when the target number of iterations was reached or when a certain criterion was accepted. The second phase aimed at ordering the tasks according to the shortest operating time and allocating them to operators with the least assigned workload in the analyzed *seru*. Then, the assigned workloads were distributed among the employees of the same *seru* by removing the assigned load from the most overloaded employee and allocating it to the least overloaded one.

Examples

Conversion of a traditional assembly line to *serus*

Considering the example illustrated in Fig. 2, suppose that a product must undergo seven operations, spanning the entire traditional assembly line. The cycle time is set at 5 s, i.e., the line must produce 1 product every 5 s. The duration of each operation (in s), with one operation performed by one operator, is listed in (Table 1):

Table 1
Duration of each operation (in seconds)

	Op. 1	Op. 2	Op. 3	Op. 4	Op. 5	Op. 6	Op. 7
Duration of the operation	4	4.3	5	4.6	4.6	4.9	4.8

Note that the durations should not be dissociated from each other as that would cause the accumulation of parts between operations. Such a production system, where the proximity between the productive resources is regularized through the proper functioning of the unitary flow, has no space for product accumulation. Therefore, line-balancing techniques were proposed to group the operations (thus forming workstations) and optimize the time. The Gantt chart may be constructed using this information about the operation time (Fig. 3). Assuming only three units of product and no interruption to the operators, the makespan (which is the instant that all units are produced) is 42.2 s.

In Fig. 3, when executing the first operation of the first product, which required 4 s, Operator 1 forwards the product to Operator 2. Subsequently, Operator 1 initiates the first operation of the second product, and so on. We observed that, in some operations, while Operator 2 is preoccupied, the product must await the next operation. An important observation to be made is that the operation that requires more time, in this case operation 3, appears to have a strong influence on the makespan. Because of this bottleneck, if one product is delayed, the entire line is delayed.

The traditional line in one of the three *serus* must be modified as illustrated in Fig. 2. We obtained the makespan by constructing the Gantt chart for each *seru*.

Fig. 4 illustrates the allocation of tasks for each operator. It is important to note that the first operation of the second operator only starts at the instant (3.9 + 4.2) s, which is also the instant at which the first operation of the second product is performed by operator 1. The makespan is the point at which the last operation of the last product is finished, and it occurs at 56.3 s. The performance is inferior compared to the traditional assembly line, which was 42.2 s.

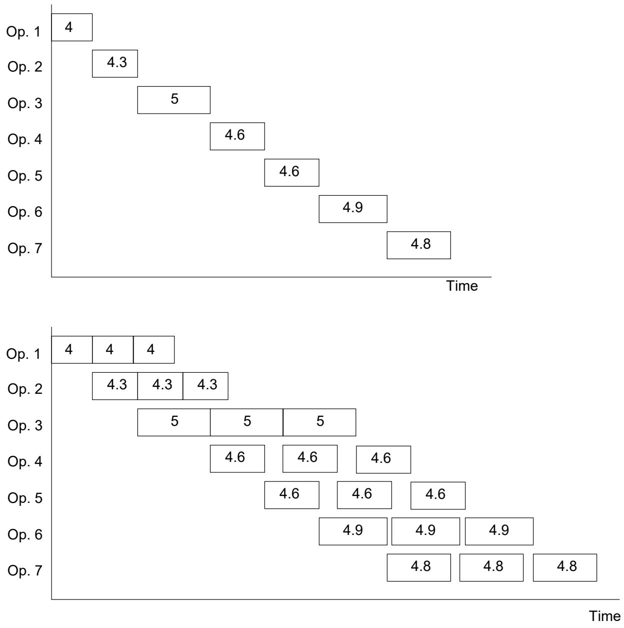

Fig. 3. Allocation of operations on the Gantt chart, from the first to the third product, respectively

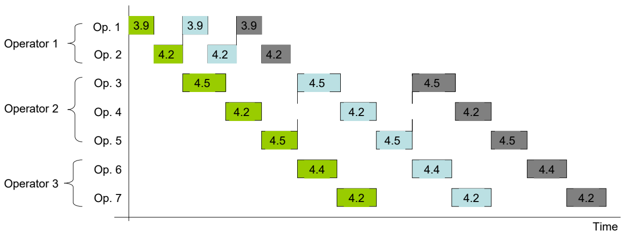

Fig. 4. Allocation of operations on the Gantt chart, from the first to the third product, respectively Makespan in divisional *seru* (3.9 + 4.2 + 4.5 + 4.2 + 4.5 + 4.5 + 4.2 + 4.5 + 4.5 + 4.2 + 4.5 + 4.4 + 4.2 = 56.3 s)

The next *seru* is the rotational type, where the operator must perform all operations of the same product, with more than one employee in the same *seru*. The makespan obtained when allocating the durations on the Gantt chart is 32.7 s, as shown in Fig. 5.

Finally, in *Yatai*, a makespan of 36 s was obtained when allocating the tasks of the three products to an employee, as observed in Fig. 6. Note that the employee only starts the operation of the next product as soon as all the operations of the previous product have been fully executed.

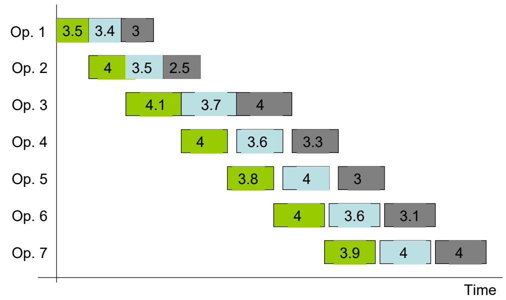

Fig. 5. Makespan in the rotational *seru* $(3.5+4+4.1+3.7+4+3.3+3+3.1+4=32.7\text{ s})$

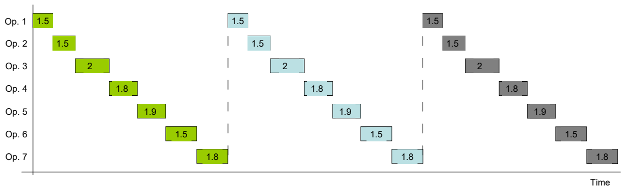

Fig. 6. Makespan in Yatai $3\times(1.5+1.5+2+1.8+1.9+1.5+1.8)=36$ s

According to the examples, the conversion of the traditional line to one of the three types of *serus* had worsened the performance in the makespan, reinforcing the claims of Kaku (2017), Liu et al. (2014), Sengupta & Jacobs (2004) and Adler & Cole (1993).

Conversion of a traditional assembly line to *serus* considering a flexible sequence

According to Gonçalves Filho & Gorgulho Junior (2003), Shih et al. (2008), Gorgulho Junior & Gonçalves Filho (2007), the same product might also have other production options. As seen in the literature review section about the *seru* production system (*Background*), production alternatives have not been explored yet.

Now, let us suppose a product P exists in an assembly line. A set of operations is performed, with a cycle time of 3 s. The arrival of two consecutive lots is 9 s, with 3 units in each lot. The durations of operations, as well as the operators responsible for the operation, are listed in (Table 2).

Table 2
Duration of each operation (in s) of each unit of product P

	A	B	C	D
Duration of operation	2.8	2.5	2.1	2.6
Operator	1	2	3	4

The rigid sequence, as shown in Fig. 7, resulted in a makespan of 24 s.

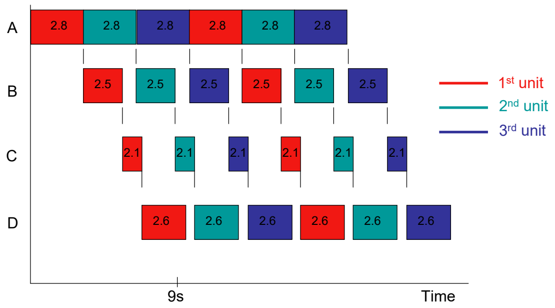

Fig. 7. Assembling according to the sequence A, B, C and D

Note that any possibility of execution of operations without a sequence is a factorial problem. For each possibility, there are makespans to be compared. To confirm that there are different makespans, let us take the following sequence as an example: D, B, A, and C. When plotting the tasks on the Gantt chart (Fig. 8), the makespan for this case is 24.6 s.

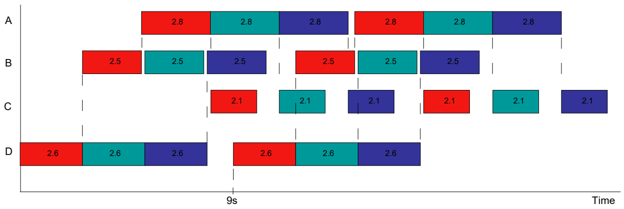

Fig. 8. Gantt chart with the allocation of tasks for one of the possible sequencesto be executed in product P

The results of the sequence (D, B, A, and C) worsened by 0.6 s from sequence (A, B, C, and D). If we consider sequence C, D, B, and A, the makespan would equal that of the rigid sequence, i.e., 24 s.

As mentioned, the focus of the *seru* production system is on the amount of skill training an employee requires. For the purpose of illustration, (Table 3) lists the relevant costs and time of training.

Note that any operator can execute any operation, albeit at different operating times and training costs. Moreover, the operators should possess three levels of skills to work in the *serus*, as shown in the table. The higher the qualification required, the higher the training cost. Likewise, the more prepared the operator, the shorter the time of operation, without compromising on the quality of products. However, these costs are generally excluded in the literature, as it is assumed that the operator manages the tasks with simple, easy-to-learn activities that do not require much

Table 3
Shows the duration of the operation, and the cost of training according to the degree of skill required

Skill	Operator		A	B	C	D
Low	1	Duration of operation (in s)	2.8	2.5	2.1	2.6
		Training cost per lot	0.3	0.2	0.2	0.1
	2	Duration of operation (in s)	2.5	2.7	2.4	2.5
		Training cost per lot	0.3	0.3	0.25	0.2
Median	3	Duration of operation (in s)	2.3	2	2	2.3
		Training cost per lot	0.6	0.4	0.5	0.23
High	4	Duration of operation (in s)	2	1.8	1.8	2
		Training cost per lot	1	0.5	0.7	0.34

qualification. We assumed that in the divisional, rotational and *yatai serus* have two, two and 1 operator(s), respectively. Besides, units from two lots are fully assembled in each type of *seru*. With sequence A, B, C, and D as the example, Table 4 defines the activities for each operator.

Note that (in Table 4) employees are allocated according to the required qualification, with the least skilled in the divisional *seru* (for performing part of the operations) and the most skilled in the *Yatai seru* (because all operations need to be performed). The Gantt charts in Fig. 9 depict the allocation of tasks from the first to third units of the considered lots.

In the first graph, referring to the divisional *seru*, Operator 2 performs the first two operations A and B on the product, finishing at 5.2 s. He/she forwards the task to Operator 1, who then executes operations C and D, ending at 9.9 s (5.2 + 4.7 = 9.9 s). When forwarding the first product to Operator 1, Operator 2 starts operations A and B on the second product, and so on. Note that the second batch had to wait for 11.8 s to start the operations. The makespan and training cost were 35.9 s and 1.8 currency units, respectively.

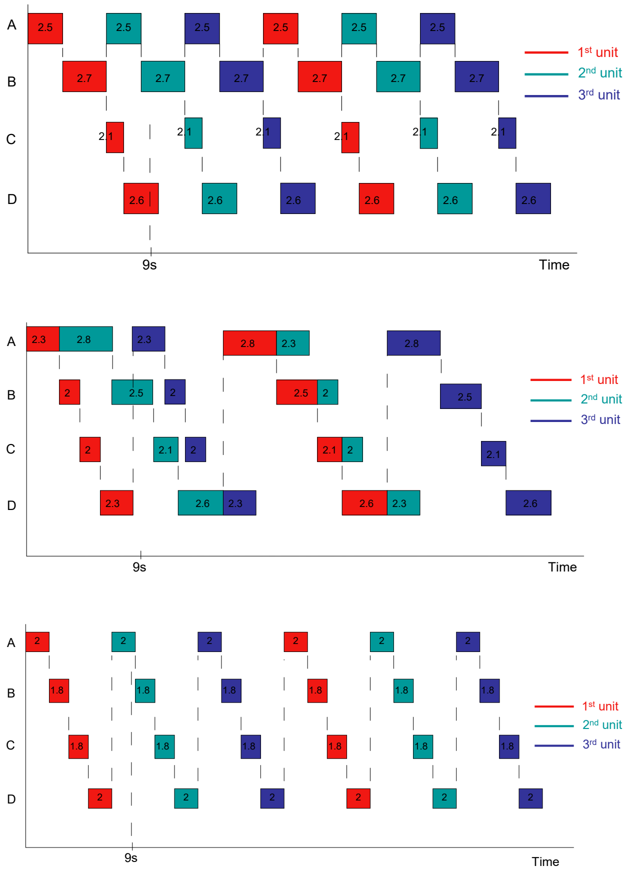

Fig. 9. Gantt charts for each type of *seru*

In the second graph, referring to the rotational *seru*, the operations are fully performed by an operator from start to end. As each lot has three units and two operators, the activities are interspersed. As soon as Operator 3 executes the first operation of the first product, Operator 1 immediately starts the first operation of the second product. The second batch must wait for 3.3 s to start processing. The makespan and training cost for assembling the products from the two

Table 4
Shows the division of tasks between the operators of each type *seru*

A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
Operator 2		Operator 1		Operator 3				Operator 1				Operator 4			
Divisional				Rotational								Yatai			

lots were 32.3 s and 5.06 currency units, respectively.

Finally, the *Yatai seru* has only one operator who performs from start to finish all the operations of the six units of the product. Meanwhile, when the second batch arrives, the operation was started after 13.8 s as the operator is preoccupied. The makespan and cost were 45.6 s and 5.08 currency units.

Comparing the assembly line for sequence A, B, C, and D with the makespans of the individual *serus*, i.e., conversion of the line to use only one of the types of *serus*, the performance in the makespan worsened (before 24 s), in addition to the increased cost. This result indicates that converting the system to a single type of *seru* does not yield the expected benefits.

Conversion of traditional assembly to *serus* considering the effect of lot splitting (division)

The following example considers the conversion of a traditional assembly line into two types of *serus*, and the last two studies concurrently. Chin (2013) stated that division of the same batch may be an important alternative that could help minimize the makespan. In that opportunity, the author analyzed for distributed layouts, and observed that balanced division of the same batch into virtual cells produced better results than balanced division. In our example, three units of each two lots were divided between the divisional and rotational *serus*, with 2 units for the divisional *seru* and one for the rotational *seru* (with Operator 3). The corresponding Gantt chart is illustrated in Fig. 10, with a makespan of 26.1 s and a training cost of 1.2 currency units (1 complete lot + 1/3 of the lot).

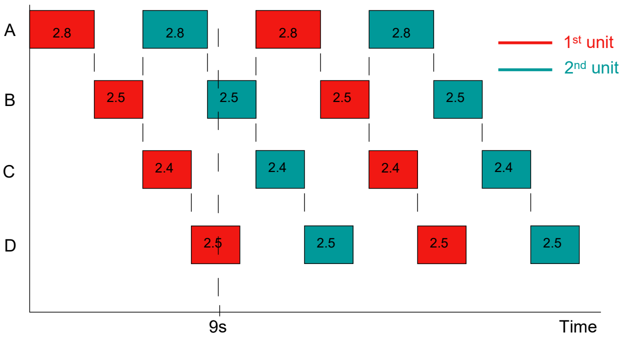

Fig. 10. Gantt chart for the divisional *seru*

In Fig. 11, when Operator 3 executes the third product of the first batch, he/she waits for 0.4 s to start the product operation in the second batch. This means that, if there were more operators working in the *seru*, they still would have to wait for the batch to arrive. Hence, Operator 1 is not needed for the execution. The makespan is 17.6 s, and the training cost is 1.15 currency units.

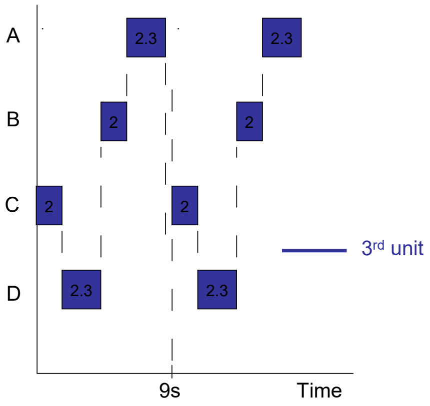

Fig. 11. Gantt chart for the production of one unit from each batch in the rotational

Therefore, if the option allowed to convert the assembly line into two *serus* working concurrently, the cost would be $1.15+1.2 = 2.35$ currency units and the makespan is $\max\{26.1; 17.6\} = 26.1$ s. Note that this result is more promising (makespan and cost) than converting the traditional line into one pure rotational or one *Yatai serus*. On the other hand, the cost obtained here is higher than converting to a pure divisional *seru* (2.35 versus 1.8).

Conversion of traditional assembly line to *serus* considering the learning effect

According to Deepthi et al. (2019), the duration of an operation is not a constant parameter. In addition to typical production variations, there is, for example, the factor of operator learning. The study showed that the more experienced the employee, the shorter the production time. The training costs tend to decrease with experience. Anzanello & Fogliatto (2007) used a learning curve to measure the performance of employees in repetitive work and described the effects of the relevant factors on the curve through mathematical models.

We supposed that the learning considerations were represented by a parameter f, which would be used for time and cost reduction for each operation (Tables 5 and 6).

Especially, we considered that the parameter f only updated the production time and cost for each batch. Because only two batches were analyzed, the variables time and cost were only updated once.

Tables 5 and 6 list the operation times of each product unit and their respective costs for two operators.

Following the same production sequence (A, B, C and D), in the divisional *seru*, Operator 2 performs the first two operations, and Operator 1 the remaining two. In rotational *seru*, Operator 3 initiates the execution of the operation, followed by Operator 1. Here, the difference that each operator in the rotational *seru* executes operations from beginning to end.

Table 5
Operating time and training cost information, per batch, for the divisional *seru*

	Divisional			
	Operator 2		Operator 1	
	A	B	C	D
First lot	2.5	2.7	2.1	2.6
Cost	0.3	0.3	0.2	0.1
Second lot	$2.5 \times f\mathrm{A}$	$2.7 \times f\mathrm{B}$	$2.1 \times f\mathrm{C}$	$2.6 \times f\mathrm{D}$
Cost	$0.3 \times f\mathrm{A}$	$0.3 \times f\mathrm{B}$	$0.2 \times f\mathrm{C}$	$0.1 \times f\mathrm{D}$

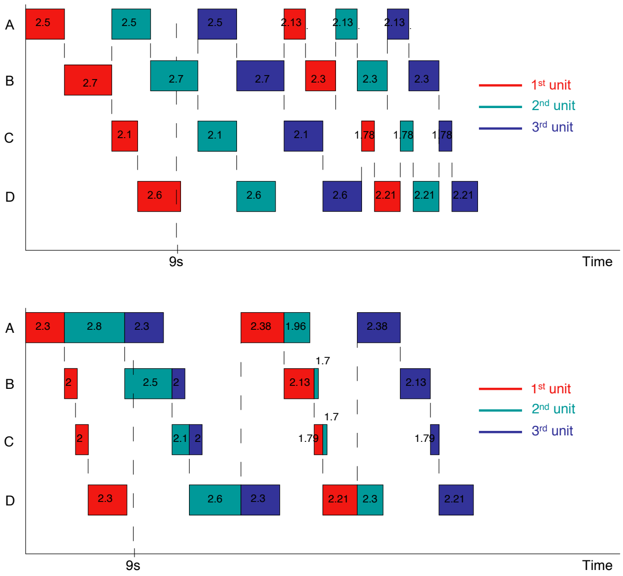

Fig. 12. Gantt chart for the divisional (left) and rotational (right) *serus* with the time and cost reduction factor

The Gantt charts for each *seru* under the assumption that $fA = fB = fC = fD = 0.85$ are presented in Fig. 12. Note that the second batch arrives at 9 s. In the first *seru*, it waited for 6.6 s before processing, and 3.3 s in the second. In the first type of *seru*, the allocation of tasks to the operators for the first batch is proceeded as planned. When processing the second batch, Operator 1 becomes the production bottleneck.

Each operator used different equipment, indicating that the delay in production was not caused by a conflict of use (awaiting the release of occupied equipment), but because of the duration of the operation. The makespan was 32.88 s for both operators. The training cost for Operator 2 was 1.11 currency units and 0.55 currency units for Operator 1, totaling 1.66 currency units. In the second type of *seru*, the equipment is shared by the operators as each operator performs all operations of the product. The bottleneck in this case is due to the conflict of use. That is, Operator 3, who had better training, awaited the arrival of equipment being used by Operator 1. The makespan in this case was 29.32 s, and the costs of training were 1.64 and 0.72 units for Operators 3 and 1, respectively, totaling 2.36 currency units.

When considering parameter f, the results show that converting the in-line production system to one of *serus* (divisional or rotational), the makespan and the costs are reduced.

In divisional *seru*, the reduction of makespan is from 35.9 s to 32.88 s and the reduction of cost is from 1.8 to 1.66.

In rotational *seru*, the reduction of makespan is from 32.3 s to 29.32 s and the reduction of cost is from 5.06 to 2.36.

These examples raise an important question: how can the performance of each *seru* be improved? As noted earlier, this question has several answers. Ap-

Table 6
Operating time and training cost information, per batch, for the rotational *seru*

	Rotational							
	Operator 3				Operator 1			
	A	B	C	D	A	B	C	D
First lot	2.3	2	2	2.3	2.8	2.5	2.1	2.6
Cost	0.6	0.4	0.5	0.23	0.3	0.2	0.2	0.1
Second lot	$2.3 \times f\mathrm{A}$	$2 \times f\mathrm{B}$	$2 \times f\mathrm{C}$	$2.3 \times f\mathrm{D}$	$2.8 \times f\mathrm{A}$	$2.5 \times f\mathrm{B}$	$2.1 \times f\mathrm{C}$	$2.6 \times f\mathrm{D}$
Cost	$0.6 \times f\mathrm{A}$	$0.4 \times f\mathrm{B}$	$0.5 \times f\mathrm{C}$	$0.23 \times f\mathrm{D}$	$0.3 \times f\mathrm{A}$	$0.2 \times f\mathrm{B}$	$0.2 \times f\mathrm{C}$	$0.1 \times f\mathrm{D}$

pendix 2 highlights the challenges that must be explored in the future.

Conclusions

The objective of this study was to conduct a bibliographic survey of *seru* production systems. The existing challenges with possible solutions, the relevant algorithms, the limitations of each model, and the performance indicators used were detailed. The possible research directions of these systems were also discussed.

Among the typical problems addressed in the literature are the formation of cells through the distribution of machines to *serus*, and the training and assignment of employees to the *serus*. The problem of workload balance between *serus* and employees was also addressed and the cost of production, training, and CO_2 emissions were minimized.

The workload balance and cost of machinery acquisition for hybrid *serus*, considered more complex than the pure *seru,* requires investigation.

The reviewed articles only considered the rigid sequence and none analyzed the flexible sequence. Further, none have verified the batch division of the same product. As we have just noted, both should be investigated since the obtained performances of *serus* are different.

The effects of employee training on the operating time and cost warrant further investigation. The review revealed that most study considered both time and cost as fixed (with unique value or with average and standard deviation). A few studies have demonstrated fluctuations in the input data for their models. Although the values of operators'skills are considered in the literature, these are defined, not calculated. Hence, the influence of learning curves on the skills should be explored.

Appendix 1

Table A.1. Classification according to Yu & Tang (2019)

Layout (*Seru* Shape)	U		Zwierzyński & Ahmad (2018), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Yilmaz (2020b), Manupati et al. (2015)
	Parallel		Zwierzyński & Ahmad (2018), Beber (2019)
Seru	Hybrid (seru plus line)		Beber (2019), Yu et al. (2017), Liu et al. (2013)
	Concomitant serus		Zwierzyński & Ahmad (2018), Liu et al. (2015), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Gong et al. (2009), Yilmaz (2020b), Manupati et al. (2015), Ren & Wang (2019)
	Reconfigurable serus?	No	Zwierzyński & Ahmad (2018), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Yilmaz (2020b), Manupati et al. (2015), Ren & Wang (2019)
		Yes	Liu et al. (2015), Gong et al. (2009)
Product	Variety	One	Zwierzyński & Ahmad (2018), Han et al. (2018), Ren & Wang (2019)
		Several	Liu et al. (2015), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Liu et al. (2013), Gong et al. (2009), Yilmaz (2020b), Manupati et al. (2015)
	Type of operations among different products	Same	Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Deepthi et al. (2019), Wang & Tang (2018), Sun et al. (2016), Gong et al. (2009), Ren & Wang (2019)
		Different	Liu et al. (2015), Ayough et al. (2020), Liu et al. (2013), Yilmaz (2020b), Manupati et al. (2015)

Table A.1. [cont.]

	Production sequence	Rigid		Zwierzyński & Ahmad (2018), Liu et al. (2015), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Gong et al. (2009), Yilmaz (2020b), Manupati et al. (2015), Ren & Wang (2019)
		Flexible		None
	Duration of each operation, per product type	Fixed		Zwierzyński & Ahmad (2018), Liu et al. (2015), Beber (2019), Han et al. (2018), Liu et al. (2013), Gong et al. (2009), Yilmaz (2020b), Ren & Wang (2019)
		Variable	Based on Learning curve	none
			Based on skills of operators	Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Gong et al. (2009), Manupati et al. (2015)
Operators	Work in different serus during processing the product	Yes		Yu et al. (2017), Yilmaz (2020b)
		No		Zwierzyński & Ahmad (2018), Liu et al. (2015), Beber (2019), Yu et al. (2014), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Gong et al. (2009), Manupati et al. (2015), Ren & Wang (2019)
Lot	Scheduling the lot to cells (FCFS, LCFS, SPT, etc)	No		Zwierzyński & Ahmad (2018), Deepthi et al. (2019), Ayough et al. (2020), Han et al. (2018), Liu et al. (2013), Manupati et al. (2015), Yu & Tang (2019)
		Yes	One Strategy	Yu et al. (2014), Yu et al. (2013), Wang & Tang (2018), Sun et al. (2016), Gong et al. (2009), Yilmaz (2020b)
			Several strategies	Yu et al. (2017), Yu et al. (2016)
	Splitting the lot	No		Liu et al. (2015), Yu et al. (2014),Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Gai et al. (2020), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Sun et al. (2016), Han et al. (2018), Liu et al. (2013), Gong et al. (2009), Yilmaz (2020b), Manupati et al. (2015), Ren & Wang (2019)
		Yes	Equally among serus, piece by piece	Zwierzyński & Ahmad (2018), Gai et al. (2020)
			Equal division of the lot among serus	none
			Unequally division of the lot among serus	none
	Sizing between same product	Fixed		Zwierzyński & Ahmad (2018), Liu et al. (2015), Gai et al. (2020), Deepthi et al. (2019), Han et al. (2018), Liu et al. (2013)
		Variable		Yu et al. (2014), Yu et al. (2017), Yu et al. (2013), Yu et al. (2016), Ayough et al. (2020), Sun et al. (2016), Gong et al. (2009), Ren & Wang (2019)

Table A.1. [cont.]

Objective function	Makespan	Liu et al. (2015), Yu et al. (2017), Gai et al. (2020), Deepthi et al. (2019), Sun et al. (2016), Han et al. (2018), Yilmaz (2020b)
	Workload of employees	Zwierzyński & Ahmad (2018), Beber (2019), Yu et al. (2014), Yu et al. (2017), Yu et al. (2016), Sun et al. (2016)
	Cost	Zwierzyński & Ahmad (2018), Liu et al. (2013), Manupati et al. (2015)
	Others	Zwierzyński & Ahmad (2018), Liu et al. (2015), Yu et al. (2014), Yu et al. (2013), Yu et al. (2016), Deepthi et al. (2019), Wang & Tang (2018), Ayough et al. (2020), Gong et al. (2009), Manupati et al. (2015), Ren & Wang (2019)

In Table A.1, "Relationship between employees" refers to the operator's ability to perform multiple tasks, which was already included.

Appendix 2

- Will changing the position of employees to perform other operation in different *serus* improve the performance?
- What if product operations are partially executed in one *seru* and continued operations in another?
- What if different types of products in the same batch are considered?
- How many *serus* per type are there? Would they be economically viable?
- What if several different lot sizes arrive? In this case, what type of *seru* would be applicable?
- How to define the number of units for each *seru* in case of splitting the lot?
- Could changes in the production sequence combined with the choice of the best collaborators and best lot size for each seru optimize the overall production process?

References

Adler P.S., Cole R.E. (1993). Designed for learning: a tale of two auto plants. *Sloan Management Review*, vol. 34, no. 3, pp. 85–94.

Anzanello M.J. and Fogliatto F.S. (2007). Learning curves: state of the art and research perspectives. *Gestao & Produç ao*, vol. 14, no. 1, pp. 109–123. DOI: 10.1590/S0104-530X2007000100010 (in Portuguese).

Ayough A., Hosseinzadeh M. and Motameni A. (2020). Job rotation scheduling in the seru system: shake enforced invasive weed optimization approach. *Assembly Automation*, vol. 40, no. 3, pp. 461–474. DOI: 10.1108/AA-07-2019-0126.

Beber J. (2019). Seru production system: assembly cells implementation in a home appliance factory. *Journal of Lean Systems*, vol. 4, no. 3, pp. 23–43 (in Portuguese).

Briere G., Chaudhry M.L. (1988). Computational analysis of single-server bulk-arrival queues Mx/G/1. *Computers & Operations Research*, vol. 15, no. 3, pp. 283–292. DOI: 10.1016/0305-0548(88)90041-X.

Chaudhry M.L. (1979). The queueing system $M^X/G/1$ and its ramifications. *Naval Research Logistics Quarterly*, vol. 26, no. 4, pp. 667–674. DOI: 10.1002/nav.3800260411.

Chaves Filho J.G.B. (2010). *Best Practices for Ensuring the Sustainability of Improvements Obtained Through Kaizen Events*, University of S ao Paulo, Brazil (in Portuguese).

Chin S.Y. (2013). Virtual cells: evaluation of different lot sizing splitting strategies. *International Journal of Manufacturing Research*, vol. 8, no. 1, pp. 18–42. DOI: 10.1504/IJMR.2013.051838.

Deepthi T.V., Ramakotaiah K. and Krishnaveni K. (2019). Research on performance of multi-skilled workers for sustainable production planning in seru production systems. *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 1–13. DOI: 10.35940/ijitee.J9172.0881019.

Filho A.A. and Florentino H. (2018). *Introduction to multiobjective optimization – scalarisation methods.* V ERMAC – Regional Meeting on Applied and Computational Mathematics, Brazil.

Gai Y., Yin Y., Tang J. and Liu S. (2020). Minimizing makespan of a production batch within concurrent systems: seru production perspective. *Journal of Management Science and Engineering*, in press. DOI: 10.1016/J.JMSE.2020.10.002.

Ghosh S. and Das M.K. (2008). Non-dominated Rank based Sorting Genetic Algorithms. *Fundamenta Informaticae*, vol. 83, no. 3, pp. 231–252.

Gonçalves Filho E.V. and Gorgulho Junior J.H. (2003). A computer simulation for automatic part generation with process flexibility for simulation. *Gepros*, vol. 2, no. 2, pp. 37–50 (in Portuguese).

Gong J., Li Q. and Tang J.F. (2009). *Improving performance of parts storage through line-cell conversion*, Chinese Control and decision conference, China.

Gorgulho Junior J.H. and Gonçalves Filho E.V. (2007). Performance analysis of distributed physical arrangements operating under parts routing with sequencing flexibility. *Revista de Gest ao Industrial*, vol. 3, no. 1, pp. 1–12. DOI: 10.3895/S1808-04482007000100001 (in portuguese).

Han X.Z., Zhang Z. and Yin Y. (2018). *Reliability analysis for a divisional seru production system with stochastic capacity*. IEEE International Conference on Industrial Engineering and Engineering Management, Thailand.

Han X., Zhang Z. and Yin Y. (2020). Reliability-oriented multi-resource allocation for seru production system with stochastic capacity. *International Journal of Manufacturing Research*, vol. 15, no. 4, pp. 371–386. DOI: 10.1504/IJMR.2020.110517.

Hasegawa K., Fukuta Y. and Saito M. (2009). A study on the workload in seru production. *The Japanese Journal of Ergonomics*, vol. 45, no. 4, pp. 219–225. DOI: 10.5100/jje.45.219 (in japanese).

Johnson D.J. (2005). Converting assembly lines to assembly cells at sheet metal products: insights on performance improvements. *International Journal of Production Research*, vol. 43, no. 7, pp. 1483–1509. DOI: 10.1080/00207540412331299594.

Josiński H., Kostrzewa D., Michalczuk A. and Switoński A. (2014). The expanded invasive weed optimization metaheuristic for solving continuous and discrete optimization problems. *The Scientific World Journal*, vol. 2014, pp. 1–14. DOI: 10.1155/2014/831691.

Kaku I. (2016). A Fundamental positive investigation into Japanese seru production systems. *IFAC-papers online*, vol. 49, no. 12, pp. 337–342. DOI: 10.1016/j.ifacol.2016.07.627.

Kaku I. (2017). Is seru a sustainable manufacturing system? *Procedia Manufacturing*, vol. 8, pp. 723–730. DOI: 10.1016/j.promfg.2017.02.093.

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. DOI: 10.1080/00207540802275889.

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, pp. 285–299. DOI: 10.1016/j.jclepro.2014.03.033.

Liu C.G., Stecke K.E., Lian J. and Yin Y. (2014). An implementation framework for seru production. *International Transactions in Operational Research*, vol. 21, no. 1, pp. 1–19. DOI: 10.1111/itor.12014.

Liu C., Yang N., Li W., Lian J., Evans S. and Yin Y. (2013). Training and assignment of multiskilled workers for implementing seru production systems. *The International Journal of Advanced Manufacturing Technology*, vol. 69, pp. 937–959. DOI: 10.1007/s00170-013-5027-5.

Manupati V.K., Deephti V.T., Ramakotaiah K. and Rao S.S. (2015). *Reconfiguration of networked seru production systems in an Indian Perspective*. International Conference on Industrial Engineering and Operations Management, United Arab Emirates.

Miyake D.I. (2006). The shift from belt conveyor line to work-cell based assembly system to cope with increasing demand variation and fluctuation in the Japanese electronics industries. *International Journal of Automotive Technology and Management*, vol. 6, no. 4, pp. 419–439. DOI: 10.1504/IJATM.2006.012234.

Mouzon G.C. and Yildirim M.B. (2008). A framework to minimize total energy consumption and total tardiness on a single machine. *Int.ernational Journal of Sustainable Engineering*, vol. 1, no. 2, pp. 105–116. DOI: 10.1080/19397030802257236.

Mouzon, G., Yildirim M.B. and Twomey J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, vol. 45, no. 18–19, pp. 4247–4271. DOI: 10.1080/00207540701450013.

Ohno T. (1997). *The Toyota Production System: Beyond Large-Scale Production*, Bookman, Brazil (in Portuguese).

Ren H. and Wang D.Y. (2019). Analysis of the effect of the line-seru conversion on the waiting time with batch arrival. *Mathematical Problems in Engineering*, vol. 2019, pp. 1–13. DOI: 10.1155/2019/4036794.

Sakazume Y. (2005). Is japanese cell manufacturing a new system? A comparative study between japanese cell manufacturing and cell manufacturing. *Journal of Japan industrial Management Association*, vol. 55, no. 6, pp. 341–349. DOI: 10.11221/jima.55.341.

Sengupta K. and Jacobs F.R. (2004). Impact of work teams: a comparison study of assembly cells and assembly line for a variety of operating environments. *International Journal of Production Research*, vol. 42, no. 19, pp. 4173–4193. DOI: 10.1080/00207540410001720421.

Shah R. and Ward P.T. (2007). Defining and developing measures of lean production. *Journal of Operations Management*, vol. 25, no. 4, pp. 785–805. DOI: 10.1016/j.jom.2007.01.019.

Shih C., Carvalho H., Filho E. and Porto A. (2008). Job-shop arrangements: comparative study of performance and proposal for the design of routing lists for employees. *Sistemas & Gest ao*, vol. 3, no. 3, pp. 262–282. DOI: 10.7177/sg.2008.SGV3N3A7 (in Portuguese).

Stecke K.E., Yin Y., Kaku I. and Murase Y. (2012). Seru: the organizational extension of JIT for a super-talent factory. *International Journal of Strategic Decision Sciences*, vol. 3, no. 1, pp. 106–119. DOI: 10.4018/jsds.2012010104.

Sun W., Li Q., Huo C., Yu Y. and Ma K. (2016). Formulations, features of solution space, and algorithms for line-pure seru system conversion. *Mathematical Problems in Engineering*, vol. 2016, pp. 1–14. DOI: 10.1155/2016/9748378.

Sun W., Yu Y., Lou Q., Wang J.W. and Guan Y. (2019). Reducing the total tardiness by seru Production: Model, exact and cooperative coevolution solutions. *International Journal of Production Research*, vol. 58, no. 21, pp. 6441–6452. DOI: 10.1080/00207543.2019.1680898.

Van der Zee D.J. and Gaalman G.J.C. (2006). *Routing flexibility by sequencing flexibility–exploiting product structure for flexible process plans*, Proceedings of the third international conference on group technology/cellular manufacturing, The Netherlands.

Villa A. and Taurino T., From JIT to seru, for a production as lean as possible. *Procedia Engineering*, vol. 63, pp. 956–965. DOI: 10.1016/j.proeng.2013.08.172.

Wang T. and Tang J.F. (2018). Cost and service-level-based model for a seru production system formation problem with uncertain demand. *Journal of Systems Science and Systems Engineering*, vol. 27, pp. 519–537. DOI: 10.1007/s11518-018-5379-3.

Wang Y., Zhang Z. and Yin Y. (2020). Workload-based order acceptance in seru production system. *International Journal of Manufacturing Research*, vol. 15, no. 3, pp. 234–251. DOI: 10.1504/IJMR.2020.108197.

Womack J.P. and Jones D.T. (2004). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Campus, Brazil (in Portuguese).

Womack J., Jones D. and Ross D. (2004). *The Machine that Changed the World*, Elsevier, Brazil (in Portuguese)

Yin Y., Stecke K. E., Swink M. and Kaku I. (2017). Lessons from seru production on manufacturing competitively in a high cost environment. *Journal of Operations Management*, vol. 49–51, pp. 67–76. DOI: 10.1016/j.jom.2017.01.003.

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. DOI: 10.1080/00207543.2016.1277594.

Yilmaz O.F. (2020a). *Operational strategies for seru production system: a bi-objective optimisation model and solution methods*, in Int. J. Prod. Res., vol. 58, no. 11, pp. 3195–3219.

Yilmaz O.F. (2020b). Attaining flexibility in seru production system by means of Shojinka: an optimization model and solution approaches. *Computers & Operations Research*, vol. 119, pp. 104917. DOI: 10.1016/j.cor.2020.104917.

Yu Y., Sun W., Tang J.F. and Wang J.W. (2017). Line-hybrid seru system conversion: Models, complexities, properties, solutions and insights. *Computers & Industrial Engineering*, vol. 103, pp. 282–299. DOI: 10.1016/j.cie.2016.11.035.

Yu Y. and Tang J. (2019). Review of seru production. *Frontiers of Engineering Management*, vol. 6, pp. 183–192. DOI: 10.1007/s42524-019-0028-1.

Yu Y., Tang J. F., Gong J., Yin Y. and Kaku Y. (2012). How to carry out assembly line–cell conversion? A discussion based on factor analysis of system performance improvements. *International Journal of Production Research*, vol. 50, no. 18, pp. 5259–5280. DOI: 10.1080/00207543.2012.693642.

Yu Y., Tang J. F., Gong J., Yin Y. and Kaku I. (2014). Mathematical analysis and solutions for multi-objective line-cell conversion problem. *European Journal of Operational Research*, vol. 236, no. 2, pp. 774–786. DOI: 10.1016/j.ejor.2014.01.029.

Yu Y., Tang J. F., Sun W., Yin Y. and Kaku I. (2013). Reducing worker(s) by converting assembly line into a pure cell system. *International Journal of Production Economics*, vol. 145, no. 2, pp. 799–806. DOI: 10.1016/j.ijpe.2013.06.009.

Yu Y., Wang S., Tang J., Kaku I. and Sun W. (2016). Complexity of line-seru conversion for different scheduling rules and two improved exact algorithms for the multi-objective optimization. *Springerplus*, vol. 5, no. 1, pp. 1–26. DOI: 10.1186/s4006401624455.

Zwierzyński P. and Ahmad H. (2018). Seru production as an alternative to a traditional assembly line. *Engineering Management in Production and Services*, vol. 10, no. 3, pp. 62–69. DOI: 10.2478/emj-2018-0017.