USING BUFFERS AND WORK-SHARING FOR MINIMIZING MAKESPAN OF SMALL BATCHES IN ASSEMBLY LINES UNDER LEARNING EFFECTS

Y. Bukchin

E. Wexler
Tel Aviv University

Abstract

The effect of workers' learning curve on production rate in manual assembly lines is significant when producing relatively small batches of different products. This research analyzes this effect and suggests applying work-sharing among the workers in such an environment to improve the time to complete the batch, namely, the makespan. Work-sharing refers to a situation where adjacent workers help each other in order to reduce idle times caused by blockage and starvation. The effect of work-sharing and existence of buffers on the makespan is examined and compared to a baseline situation, with no work-sharing and buffers. We present mixed-integer linear-programing (MILP) formulations, which minimize the makespan and provides optimal work allocation. A numerical study is conducted and the results along with some operational insights are presented.

1 INTRODUCTION

The effect of workers' learning curve on production rate in manual assembly lines is significant when producing relatively small batches of different products. The worker's learning curve is relatively steep at this stage and highly affects the time to complete the batch, namely the makespan. The importance of the makespan performance measure stems from its equivalence to the throughput rate of the line. Another factor, other than the learning curve, that may affect the line performance is the buffers between stations. Buffers diminish blockage and starvation phenomena, as the upstream station has more room for locating the finished products and the downstream station can consume items from the input buffer when the upstream station is busy. In order to improve the line performance, we suggest applying work-sharing between the workers. Work-sharing is a concept that was initially defined by Ostolaza et al. [1] and its purpose is “helping your neighbor when they fall behind”. Each station may consist of ‘fixed’ and ‘shared’ tasks; the ‘fixed’ tasks are performed by the same station during all the production run, while the ‘shared’ tasks are performed for some cycles in one station and for the other cycles in the downstream or upstream adjacent station. As a result, the line can be dynamically balanced due to load transfer between the stations and consequently reduces the makespan.

In this research we study assembly lines of small batches under learning effects. We analyze different environment sets that strive to balance the line and reduce idle times in the stations. The main studied environment parameters are (a) existence of buffer between the stations and (b) applying work-sharing between stations.

The proposed approach is based on mixed-integer linear-programing (MILP) which minimizes the makespan and finds optimal work allocation. Variations of the MILP formulations are developed for solving lines with buffers or without, with work-sharing or without and any combination of the above.

2. LITERATURE REVIEW

The effect of learning in assembly line is mainly relevant in two cases: assembly line of small batches and high labor turnover. In oppose to mass production, in lines of small batches, the learning period becomes a more substantial part of the production. Therefore, learning should be considered in the design process, or in particular, in the task assignment to stations/workers [2]. The balancing problem of assembly lines of small batches, when learning effects are considered, has been studied by Thongsanit et al. [2], Karni and Herer [3] and Cohen et al. [4]. Karni and Herer [3] claimed that in addition to the learning effects, assembly lines of small batches are highly affected by ‘end effect’: due to the tasks sequence through the stations, at the beginning some stations are idle, and so happens at the end as well. Learning significantly affects assembly lines also in the case of high labor turnover; new and inexperienced workers replace experienced workers, and as a result, there is a reduction of throughput rate until the new workers acquire the necessary skills and experience. Armbruster et al. [5], Montano et al. [6] and Villabolos et al. [7] analyze different methods in order to analyze these lines.

In this research, we model the learning effect by the classic power learning curve model [8, 9], as $t_n = t_1 \cdot n^{-b}$, where t_n denote that the cycle time of the n^{th} cycle and b is the learning coefficient. The connection between b (learning constant) and $\emptyset$ (learning slope) is expressed by $b = -\log_2(\emptyset)$. This model is relatively simple and is widely used in Industry [6]. An assembly line balancing problem with learning effects assumes that the batch size to be processed is small and discrete, so the planning horizon is finite and the end effect must be considered [3]. The learning phenomenon may depend on the task

[3] or the worker [2, 5, 6]. When learning depends on the worker, one can assume that the learning slopes in the line are homogenous (same in all stations) or heterogeneous (different in stations). Cohen et al. [4] dealt with the homogenous problem. They developed the “upper envelope” and aimed to minimize the area under this line, which corresponds to the makespan. Typically, the learning slope of individual workers may vary in the range 65% to 95% [10]. The processing times in the stations, which are affected by the learning rate, can be referred to as discrete [3] or as continuous [4]. Cohen et al. [4] assume that tasks can be divisible whereas Karni and Herer [3] assume the tasks cannot be split. Most previous studies on the assembly line balancing problem considering learning ability assume no buffer space between workstations [4]. In that case the line is synchronous, as the tasks’ completion time at each workstation is determined by the bottleneck station.

Work-sharing refers to a situation where more than one worker is capable of performing the same task. Ostolaza et al. [1] were the first to coin the term of dynamic line balancing (DLB). This term refers to the operational side of work sharing as "allowing tasks to be assigned 'on the fly' based on the current state of system". The basic idea is shifting workload from high loaded to less loaded stations, by allowing different workers to perform the same task. A dynamic control on the buffer between every two stations can be used to determine in each cycle the identity of the station to perform the shared task. Another approach of DLB named Bucket Brigade (BB) was presented by Bartholdi and Eisenstein [11]. They offered a "line that balanced itself", for which workers must be cross-trained on the tasks of typically the whole line. Anuar and Bukchin [12] analyzed the reduction in the cycle time of a given line applying DLB and suggested analytical conditions for "line balanceability" in lines where forward sharing is allowed (a task can be performed by its current or the adjacent downstream station). Also, they proposed several tools for the design and operation of such assembly lines, while assuming a strict order of assembly sequence. Bukchin and Sofer [13] extended [12], as they addressed the problem of applying work-sharing in working assembly lines, with an initial assignment of tasks to stations and given technological precedence constraints among tasks. They analyzed work-sharing when tasks can be performed by the adjacent upstream or downstream station.

Applying work-sharing in assembly lines under learning effects was studied also in the case of high labor turnover. When the BB is applied in such an environment, it suffers from the fact that the slowest to fastest workers assignment does not always hold. Under this limitation, Armbruster et al. [5] studied the dynamics of a BB operating under workers' learning. They concluded that assembly line organized under the concept of BB may be robust if reordering of the workers is allowed. Montano et al. [6] showed that BB performs poorly when worker speeds are similar and proposed adding control buffers to avoid these harmful effects and to increase line's flexibility. They presented a Modified Work Sharing (MWS) methodology which uses the concepts of control buffers and work zones used on the traditional work sharing. They showed that MSW performed better than BB except in the case when one worker was significantly slower than the others. Villabolos et al. [7] analyzed the MWS and compared the performance of this line design to that of a line based on BB and traditional designs. They showed that throughput behavior of the models indicates that the MWS and the BB methods are superior to traditional methods of serial assembly lines and for longer and more realistic lines the MWS superior over the BB.

3. LINES WITH BUFFERS AND WORK-SHARING

3.1 Work-sharing model description & assumptions

Assembly lines quickly become unbalanced due to the learning of the workers that causes significant changes in the assembly time during the operation. As a result, idle times occur due to blockage and starvation between the stations. As an attempt to better balance the load we suggest applying work-sharing mechanism. Work-sharing aims to dynamically improve the balance of the line by transferring load between the stations. We define two types of assembly times/tasks: 'fixed' that is performed by the same station for all the items/cycles and 'shared' which can be performed by different successive stations for different items. This task shifting between the stations during the line operation enables a dynamic change of the stations load and, as a result may reduce the makespan.

In the proposed model, we assume that the identity of the shared tasks (and by the definition, the identity of the fixed task as well) remain the same for the whole batch. This is for two reasons: (1) to simplify the work-sharing mechanism; and (2) to enable previous training of the workers to perform the predetermined assigned tasks. Consequently, once the tasks of the first cycle were assigned to the workers, whether as fixed or shared, this assignment remains unchanged for the whole production run.

Both the fixed and shared task times follow the workers' learning curve. In the proposed model, we assume that all tasks are different (each requiring a different skill), and in particular, the fixed and shared tasks. Hence, no experience can be gained from performing the fixed tasks for the sake of the shared tasks. In other words, when the worker start performing the shared task, the first cycle time is not affected by the number of cycles the fixed task was performed by the same worker previously.

As mentioned above, we assume that each shared task is assigned to two adjacent stations. As such, it should be performed in one of these stations for every cycle. We define a 'switching point' as the cycle in which the identity of the station to perform the shared task is changed. We examined two cases of the number of the switching points: (1) smaller than or equal to one, and (2) not bounded. The former enables managing the line with a relatively simple control system, while the latter enables improving the solution by using the optimal number of switching points.

3.2 Model Formulation

Work-sharing principles are incorporated into the MILP formulation, which determines the work allocation to the stations. The model determines the fixed and shared time in each station and the identity of the station which performs the shared time in each cycle. For the latter, integer variables are needed. The notations used are given as follows. Although in this paper a 2-station line is analyze, the general form of the model, for an *M*-station line is given.

Indices:

i number of item (cycle) of the fixed task (*i*=1..*N*)

j number of station (*j*=1..*M*)

l direction of sharing: downstream or upstream, $l \in \{d, u\}$. If $l = u$, the shared task is performed in station j or $j - 1$, if $l = d,$ the shared task is performed in station j or $j + 1$

k temporary index that equals to the number of cycles done for the shared task of item i in station j in direction l

Parameters:

N batch size

M number of stations

b_j learning constant in station j (derived from the learning slope, $\emptyset_{\mathrm{j}}$)

T total workload of the first cycle

sw_j maximum number of switching points between stations j and j+1

R large number

Variables:

t_{ij} assembly time of item (cycle) *i* in station *j*

c_{ij} completion time of item *i* in station *j*

f_{ij} fixed tasks duration of item *i* in station *j*

s_{ij}^l shared task duration of item *i* in station *j* in direction *l*

$$sn_{ij}^l = \begin{cases} s_{ij}^l & \text{if shared task of item } i \text{ in direction } l \text{ is actually performed in station } j \\ 0 & \text{otherwise} \end{cases}$$

ds_{ij}^l shared task status: equals 1 if the shared task of station *j* in direction *l* is done at that station for item *i* , and 0 otherwise.

cs_{ij} switching point indicator: equals 1 if switching occurs in item i in station j in direction l and 0 otherwise.

x_{ijk}^l control boolean variable to determine the value of k of item i in station j in direction l

Model formulation

$$min\, c_{NM} \tag{1}$$

$s.t.$

$$c_{11} \geq t_{11} \tag{2}$$

$$c_{ij} - t_{ij} \geq c_{i-1,j} \qquad i\text{=2.., }j\text{=1..} \tag{3}$$

$$c_{ij} - t_{ij} \geq c_{i,j-1} \qquad i\text{=1.., }j\text{=2..} \tag{4}$$

$$c_{ij} - t_{ij} \geq c_{i-1,j+1} - t_{i-1,j+1} \qquad i\text{=2..}N\text{, }j\text{=1..}M\text{−1} \tag{5}$$

$$\sum_{j=1}^{M} f_{1j} + \sum_{j=1}^{M-1} s_{1j}^{d} = T \tag{6}$$

$$f_{ij} = f_{1j} \cdot i^{\left(-b_j\right)} \qquad i = 2..N, j = 1..M \tag{7}$$

$$t_{ij} \geq f_{ij} + sn_{ij}^{d} + sn_{ij}^{u} \qquad i = 1..N, j = 1..M \tag{8}$$

$$sn_{ij}^{l} \geq s_{ij}^{l} - R \cdot \left(1 - ds_{ij}^{l}\right) \qquad i = 1..N, j = 1..M, l \in \{d, u\} \tag{9}$$

$$s_{ij}^{l} \geq s_{1j}^{l} \cdot k^{\left(-b_j\right)} - R \cdot \left(1 - x_{ijk}^{l}\right) \qquad i = 2..N, j = 1..M, l \in \{d, u\}, k = 1..N \tag{10}$$

$$\sum_{k=0}^{N} k \cdot x_{ijk}^{l} = \sum_{o=1}^{i} ds_{oj}^{l} \qquad i = 2..N, j = 1..M, l \in \{d, u\} \tag{11}$$

$$\sum_{k=0}^{N} x_{ijk}^{l} = 1 \qquad i = 2..N, j = 1..M, l \in \{d, u\} \tag{12}$$

$$ds_{i,j}^{d} + ds_{i,j+1}^{u} = 1 \qquad i = 1..N, j = 1..M-1 \tag{13}$$

$$cs_{ij} \geq ds_{ij}^{d} - ds_{i-1,j}^{d} \qquad i = 2..N, j = 1..M-1 \tag{14}$$

$$cs_{ij} \geq ds_{i-1,j}^{d} - ds_{ij}^{d} \qquad i = 2..N, j = 1..M-1 \tag{15}$$

$$\sum_{i=2}^{N} cs_{ij} \leq sw_j \qquad j = 1..M-1 \tag{16}$$

$$sn_{i1}^{u} = 0; sn_{iM}^{d} = 0 \qquad i = 1..N \tag{17}$$

$$t_{ij} \geq 0;\ c_{ij} \geq 0 \qquad i = 1..N, j = 1..M \tag{18}$$

$$s_{ij}^{l} \geq 0; f_{ij} \geq 0;\ sn_{i}^{l} \geq 0 \qquad i = 1..N, j = 1..M, l \in \{d, u\} \tag{19}$$

$$ds_{ij}^{l} = \{0,1\} \qquad i = 1..N, j = 1..M, l \in \{d, u\} \tag{20}$$

$$cs_{ij} = \{0,1\} \qquad i = 2..N, j = 1..M \tag{21}$$

The model objective (1) minimizes the makespan, which is equal to the completion time of the last item in the last station. Constraint (2) sets the completion time of the first item in the first station. The next three constraint sets, (3)-(5), refer to the start time of each item *i* in each station *j*, $(c_{ij} - t_{ij})$, reflecting the relations between the stations; constraint (3) and (4) assure that item *i* starts processing in station *j* after the previous item completed processing in station *j* (constraint (3)) and item *i* completed processing in the previous station (constraint (4)). Constraint (5) is a no-buffer constraint and it ensures that item *i* can start processing in station *j* only after the previous item is uploaded in the next downstream station. Note that constraint set (5) can be easily modified to capture the buffer case. Constraint (6) assures that the sum of the fixed and shared times of the first item, allocated to the different stations, is equal to the predetermined total assembly time. The effect of the learning slope on the fixed task duration is captured by constraint set (7). Constraint (8) assures that the workload of item *i* in station *j* is larger than or equal to the fixed time plus the actual shared tasks duration to the total assembly time for item *i* in station *j*. Constraint set (9) addresses the shared time allocation of item i in station j. $\mathrm{sn}_{\mathrm{ij}}^{\mathrm{l}}$ gets the minimum value due to the objective functions. The effect of the learning slope on the shared task duration is captured by constraint set (10). The number of cycles considered for the shared tasks is calculated in constraint sets (11)-(12): control variable, $\mathrm{x}_{\mathrm{ijk}}^{\mathrm{l}}$, is used to set the number of cycles considered for the shared task. Constraint (13) ensures that a specific shared task is performed in only one station: downstream or upstream station. Constraint (14) and (15) set the value of 1 to $\mathrm{cs}_{\mathrm{ij}}$ if a switching occurs between station *j*-1 and *j*. Constraint (16) ensures that the number of switches is lower than the predefined limit. Constraint (17) refers to boundary values. We conclude the model with additional inequality constraint, constraints (18) and (19) and Boolean definition, constraints (20) and (21).

4. NUMERICAL STUDY

4.1 Experimental Design

A numerical study was performed to examine the makespan improvement for a different set of environment parameters, for 2-station lines with a batch size of 25 items. The workers learning slopes were assumed to be in the range 70%-90% and they were ordered in decreasing, increasing and identical learning slopes. The effect of work-sharing and existence of buffer on the investigated objective was examined. Due to scalability issues, the optimality of some results has not been verified. Based on the examined problem parameters, a general natation for the problem was developed: N/M/B/S/E/SW/$\emptyset_1\emptyset_2 \ldots \emptyset_\mathrm{m}$. N is the batch size, M is the number of stations, B refers to the existence of buffer between the stations (B=buffer exists, NB=no buffer), S refers to the existence of work-sharing (S=with work-sharing, NS=without work-sharing), E refers to the experience factor (E=experience is considered, NE=no experience is considered), SW refers to the switching point (1 or unbounded), and $\emptyset_1\emptyset_2 \ldots \emptyset_\mathrm{m}$ are the workers' learning slopes in all stations. If the factor is irrelevant to the problem, it was denoted by '/-/'. For example, the notation '10/2/NB/S/NE/O/9070' refers to a batch of 10 items assembled in a 2-station line, with no buffer space between stations, work-sharing is applied, no previous experience is considered, the number of switching points is unbounded and the learning slopes of the workers in the first and second station is equal to 90% and 70%, respectively. Another example, '25/3/B/NS/-/-/758085', refer to a batch size of 25 items assembled in a 3-station line, with infinite buffer space between stations, work-sharing is not applied, for workers with learning slopes of 75%, 80% and 85% in stations 1, 2 and 3, respectively. Since work-sharing is not implemented in this problem, previous experience and number of switching points are signed as irrelevant. Recall that in these experiments, only the case of 'no experience' (different tasks) was studied.

4.2 General Results

In general, results show that both applying work-sharing and adding buffers to the line are able to reduce the makespan. Allowing multiple switching points (versus one) is effective only in the non-buffered problems. Figures 1 presents the makespan values for the different environments in a 2-station line and a batch size of 25 items. The graphs look somewhat as a concave function in which the best makespan values are located in the edges. In other words, the makespan is lower, in general, in cases where there is a large difference between the workers learning slopes. This can be explained by the fact that although the average learning slop remains the same for all problems, in the cases of non-identical learning slopes, the workers with the lower slope are utilized much more than the other workers, resulting in an improved performance.

We can see that the combination of buffers and work-sharing with optimal number of switching points provides the lowest makespan in all experiments. However, the effect of these factors is dependent on the order of the workers. In the decreasing learning slopes cases (slow learner to fast learner), shown in the left hand side of the graphs, the main makespan reduction is obtained due to the existence of buffers. In the increasing learning slopes cases (fast learner to slow learner), shown in the right hand side of the graphs, applying work-sharing has the main impact on the makespan. Allowing optimal number of switching points improves the makespan in the non-buffered problems. In all cases of the buffered problem, allowing multiple switching point does not provide any improvement, meaning that the optimal number of switching point is one.

Table 1 presents the makespan improvement of all cases over the baseline case (no buffer, no work-sharing). We can see that the largest improvements of up to 9.5% are obtained for the decreasing cases with buffers. In these cases, most of the improvement refers to the buffers and the contribution of work-sharing is relatively small. The last result is typical also to the homogeneous cases. The increasing cases show a different pattern. Placing buffers with no work-sharing provides no makespan improvement. However, the buffers are highly relevant when work-sharing is applied. The contribution of work-sharing combined with the existence of buffers reaches up to 5.4%, and it is much higher than the contribution of work-sharing alone (up to 2.5%).

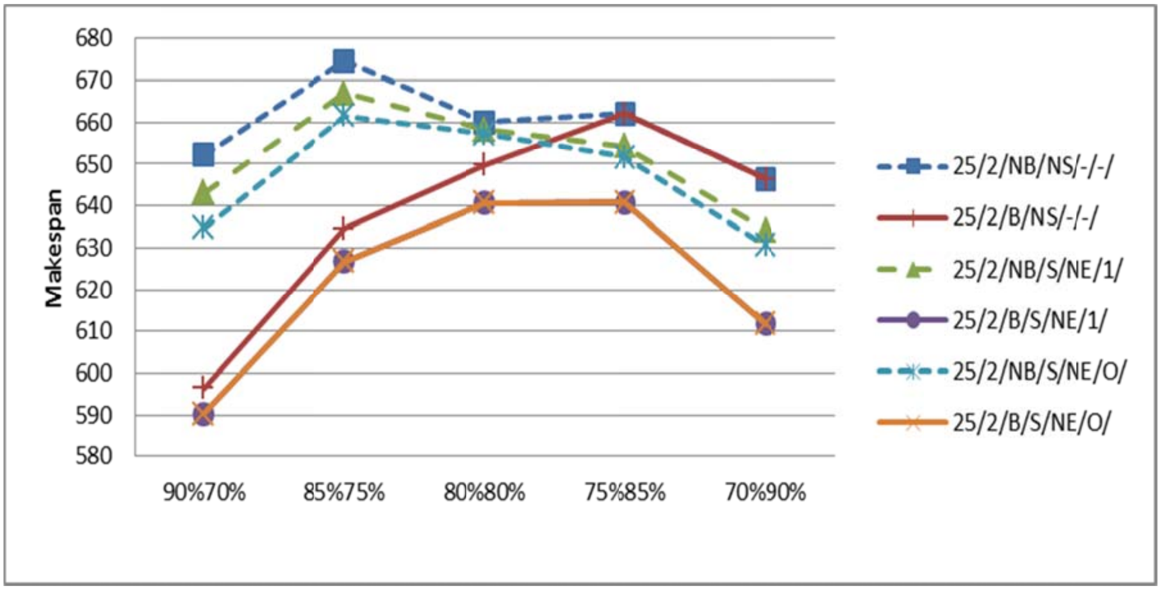

Figure 1 - Makespan of batch size of 25 items of 2-station lines

Table 1 - Makespan improvement for 2-station lines compared to the basic model [25/2/NB/NS/-/-//]

$\emptyset_1$ $\emptyset_2$	90% 70%	85% 75%	80% 80%	75% 85%	70% 90%
[25/2/B/NS/-/-/]	8.5%	5.9%	1.5%	0.0%	0.0%
[25/2/NB/S/NE/1/]	1.4%	1.1%	0.3%	1.2%	1.9%
[25/2/B/S/NE/1/]	9.5%	7.1%	2.9%	3.2%	5.4%
[25/2/NB/S/NE/O/]	2.7%	1.9%	0.4%	1.6%	2.5%
[25/2/B/S/NE/O/]	9.5%*	7.1%*	2.9%*	3.2%*	5.4%*

5. WORK-SHARING EFFECT ON MATERIAL HANDLING

Work-sharing via cross-trained workers is a win-win growing trend in modern industry. The socio-economic conditions of the workers improve over the years, and modern employees ask for enriched work content and more challenging tasks. The improved capabilities of workers in modern industry can be exploited for improving the system performance, by enhancing their training. As a result, more and more cross-trained workers can be found in industrial and service organizations, and much research is conducted on this topic.

The potential of using cross-trained workers in assembly lines is very high due to the lack of WIP inventory and the need to avoid blockage and starvation. As shown above, work-sharing may significantly improve the line operational performance measures. However, the implementation of work-sharing in assembly lines may affect the MH systems, and possibly, traditional MH systems should be redesigned to comply with the new dynamics of the systems. Assembly lines, which are usually characterized by continuous flow of items (usually a transfer batch of a single item), require expensive, possibly automated, MH system, consisting of conveyance system, fixtures, turning/lifting/positioning devices, manipulators, cranes, feeders etc. Traditional MH systems implicitly assume no sharing between workers/stations, and that predefined fixed work content is repeatedly performed in each station by the worker assigned to this station. However, the implementation of work-sharing adds complexity to the system, as some work elements have to be performed by different workers (at different stations) for different items. Since the work content of each station is no longer fixed, the MH equipment should support the new situation, which may require different orientation of the item in different cycles, more flexible transfer equipment due to changing station times, etc. in some cases, the assembly equipment, which is sometimes attached to the MH system, should be mobile between stations to support the operation. We believe that the implementation of work-sharing may establish the basis for more specific research on MH equipment to support the new operational features in assembly lines.

6. SUMMARY AND CONCLUSIONS

In this paper, assembly lines producing small batches of items were studied, when work-sharing is applied and workers production rate follows learning curve effect. The effect of work-sharing was analyzed with and without the existence of buffers between the stations. To this end, Mixed-integer linear-programing (MILP) formulations were developed, for the different problem combinations.

Experimental results showed that work-sharing and buffers help to reduce the makespan in different environment settings. In general, the buffers are more effective when the workers are ordered in a decreasing learning slopes (slow learner to fast learner) while the work-sharing is more effective in the opposite order of the workers (fast learner to slow learner).

References

[1] Ostolaza, J., McClain, J.O., and Thomas, J., “The use of dynamic (state-dependent) assembly line balancing to improve throughput”, *Journal of Manufacturing Operations Management*, 3, 105–133, (1990).

[2] Thongsanit K., Tharmmaphornphilas W. and Boondisakulchok R., “Heuristic For Task-Worker Assignment With Varying Learning Slopes”, *Engineering Journal*, 14, 2, 1-14, (2010).

[3] Karni, R. and Herer, Y., “Allocation of tasks to stations in small-batch assembly with learning: basic concepts”. *International Journal of Production Research*, 33,11, 2973-2998 (1995).

[4] Cohen, Y., Vitner, G. and Sarin, S.C., “Optimal allocation of work in assembly lines for lots with homogeneous learning”, *European Journal of Operational Research*, 168, 3, 922–931 (2006).

[5] Armbruster D., Gel E. S., Murakami J., “Bucket brigades with worker learning”, *European Journal of Operational Research*, 176, 264–274 (2007).

[6] Montano, A.,Villalobos, J. R., Gutierrez, M. A., Mar, L., “Performance of serial assembly line designs under unequal operator speeds and learning”, *International Journal of Production Research*, 45, 22, 5355-5381 (2007).

[7] Villalobos, J.R, M.A. Gutierrez, L. Mar, and O. Sanchez, “The Use of Dynamic Work Sharing Production Methods to Adapt to Labor Turnover in Serial Assembly Lines”, *International Journal of Manufacturing Technology and Management*, 23, 1-2, 34-53 (2011).

[8] Wright, T.P., “Factors affecting the cost of airplanes”, *Journal of Aeronautical Sciences*, 3, 4, 122–128 (1936).

[9] Yelle, L.E., “The learning curve: Historical review and comprehensive survey”, *Decision Sciences*, 10, 302–328 (1979).

[10] Dar-El, E.M., *Human learning: from learning curves to learning organizations*, Kluwer: Academic Publishers (2000).

[11] Bartholdi, J.J. and Eisenstein, D.D., “A production line that balances itself”, *Operations Research*, 44, 1, 21–34 (1996).

[12] Anuar R., Bukchin Y., “Design and operation of dynamic assembly lines using work-sharing”, *International Journal of Production Research*, 44, 18-19, 4043-4065 (2006).

[13] Bukchin, Y. and Sofer, T., “Bi–Directional Work-Sharing in Assembly Lines with Strict and Flexible Assembly Sequences”, *International Journal of Production Research*, 49, 14, 4377-4395 (2011).