Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 2002–2008

The Assembly Line Balancing Problem with Task Splitting: A Case Study

W. Grzechca*, L. R. Foulds**

**Institute of Automation Control, Silesian University of Technology, 44-100 Gliwice ul. Akademicka 2A Poland (Tel: 0048322372198; e-mail: waldemar.grzechca@polsl.pl)*
***Institute of Informatics, Federal University of Goiás, Samambaia Campus, Goiânia, GO 74001–970 Brazil, e-mail: lesfoulds@inf.ufg.br*

Abstract: The assembly line balancing problem (ALBP) involves distributing the tasks needed to manufacture any unit of the products to be assembled among the work stations along a manufacturing line. It is usually assumed that the required tasks cannot be split, that is, each must each be performed at a single station. However, this is not always the case in practice where task splitting can sometimes lead to improved line balancing. Indeed, we report on a case study of the assembly line in an actual Polish factory where it is possible to split certain tasks among more than one station. We show how the precedence graph can be modified to allow for task splitting and discuss the application of some existing ALBP heuristics that lead to improved line time for the reported case study. We conclude that task splitting, where appropriate, has the potential to significantly improve assembly line performance.

Keywords: manufacturing systems, assembly line balancing problem, quality measures, heuristic searches, production control

1. INTRODUCTION

The manufacturing assembly line was first introduced by Henry Ford in the early 1900's. It was designed to be an efficient, highly productive way of manufacturing a particular product. The basic assembly line consists of a set of workstations arranged in a linear fashion, with each station connected by a material handling device. The basic movement of material through an assembly line begins with a part being fed into the first station at a predetermined feed rate. A station is considered any point on the assembly line in which a task is performed on the part. Once the part enters a station, a task is then performed on the part, and the part is fed to the next station. The time it takes to complete a task at each operation is known as the process time. The cycle time of an assembly line is predetermined by a desired production rate. This production rate is set so that the desired amount of end product is produced within a certain time period (Baybars 1986). If the sum of the processing times within a station is less than the cycle time, idle time is said to be present at that station (Erel et al. 1998). One of the main issues concerning the development of an assembly line is how to arrange the tasks to be performed. This arrangement may be somewhat subjective, but has to be dictated by implied rules set forth by the production sequence (Kao 1976). For the manufacturing of any item, there are some sequences of tasks that must be followed and these are represented in a precedence graph. The assembly line balancing problem (ALBP) originated with the invention of the paced assembly line for the mass production of standardized commodities. The idea of line balancing was first introduced by Bryton (1954) in his thesis as a medium term planning challenge. The first published study involving mathematics was reported by Salveson (1955) who formulated the problem as a linear program. Helgeson and Birnie (1961) were the first to propose the ALBP. However, during the first forty years of the assembly line's existence, only trial-and-error methods were used to balance the lines (Erel and Sarin, 1998). However, Gutjahr and Nemhauser (1964) showed that the ALBP falls into the class of NP-hard combinatorial optimization problems. Furthermore, practical ALBP instances can be extremely large, with thousands of tasks. Thus, planners often have to examine a huge number of alternative balancing plans to deal with uncertain model mix information and technological and logistical constraints. For these reasons, heuristic methods have become the most popular techniques for solving the problem (Fonseca et al. 2005). Priority based methods apply priority rules to rank tasks according to specific task weight (priorities). At each iteration, a task with the highest priority is chosen from a set of available tasks and assigned to a station (Scholl and Voss, 1996). Two different strategies for constructing this set may be distinguished (station-oriented procedures and task-oriented procedures). In the literature a great variety of priority rules has been proposed and compared. The most popular variants of the ALBP considered in the literature are:

- The Simple Assembly Line Balancing Problem-1 (SALBP-1): Given the cycle time *c,* minimize the number *k,* of stations.
- The Simple Assembly Line Balancing Problem -2 (SALBP-2): Given the number *k,* of stations, minimize the cycle time *c*.

10.1016/j.ifacol.2015.06.382

Most types of assembly line balancing problems are based on a set of limiting assumptions (Baybars, 1986:; Scholl, 1999; Scholl and Becker, 2006):

1. Mass-production of one homogeneous product.
2. All tasks are processed in a predetermined mode.
3. Paced line with fixed common cycle time according to market demand.
4. The line is considered to be serial with no feeder lines or parallel elements.
5. The processing sequence of tasks is subject to precedence restrictions.
6. Deterministic task times.
7. No assignment restrictions of tasks besides precedence constraints.
8. No task can be split among two or more stations.
9. All stations are equally equipped with respect to machines and workers.

Most of these assumptions have been relaxed or somehow modified by various model extensions considered in the literature. In this paper, Assumption 8 above is not accepted, that is, the case where some tasks may be split among more than one station is discussed. The remainder of the paper is organized as follows: in Section 2 selected assembly line structures are presented, heuristic methods are discussed more precisely in Section 3. Section 4 includes popular assembly line balancing problem quality measures of final results. A numerical example is reported in Section 5. Finally, our conclusions are given in Section 6.

2. SELECTED ASSEMBLY LINE STUCTURES

In practice we can find different structures of assembly lines (serial line, parallel workstations line, two-sided line, u-line, etc.). All of them can be classified into two general types: type I with a constant cycle time and minimization of workstations' number and type II with constant number of workstations and minimization of cycle time. All of them can be used in single or multi/mixed model production. A lot of methods and procedures were introduced for solving the assembly line balancing problem that can satisfy future production rates and the utilization of tools in real companies. The first described assembly line structure was the serial assembly line (Fig. 1). A traditional line organizes workstations and the tasks that comprise them sequentially along a straight line. Nowadays, many products are produced not only from simple parts but very often from complex elements (assembled earlier). It means that even complex products need only a limited number of assembly operations. Therefore serial layout is still popular in the assembly of final products.

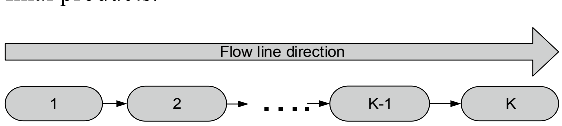

Fig. 1. Single assembly line

The U-line structure was introduced for the first time in 1994. In a U-line layout, workstations are arranged around a U-shaped line (Fig. 2). Operators work inside the U-line. The U-shaped assembly line has become an alternative for assembly production systems since the operator may perform more than one task located in different places of assembly line. Moreover, U-line disposition allows for more possibilities of how to assign the tasks to the workstations and therefore the number of workstations needed for U-shaped line layout is never more than the number of workstations needed for the traditional straight assembly line. In the traditional assembly line balancing problem for a given cycle time, the set of possible assignable tasks is confirmed by those tasks whose predecessors have already been assigned to workstations, whereas in the U-line balancing problems, the sets of assignable tasks is determined by all those tasks whose predecessors and successors have already been assigned (Miltenburg 1998, 2001). One of the important characteristics that make U-shaped structures different from straight assembly lines is that the entrance and the exit of these lines are at the same position (Shwetank et al., 2013).

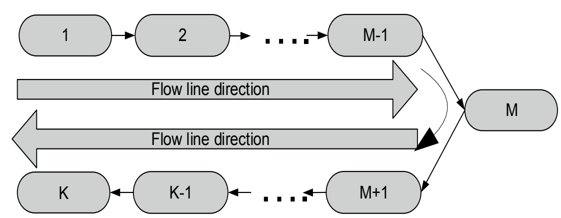

Fig. 2. U-shaped assembly line

Products enter the U-shaped assembly line at the front-side and exit from the back-side of the line. Studies on U-shaped assembly lines provide evidence for the potential to improve visibility and communications skills between operators, reduce operator requirements, increase quality, reduce work-in-process inventory, etc. (Nakade et al. 2008).

In the same year as the U-line layout was introduced, the problem of balancing parallel assembly lines was presented (Fig. 3). Gökcen et al. (2006) and Ismail et al. (2011) studied alternative assembly line designs for single and mixed model products. The objective was to determine the number of assembly lines with minimum total manpower. The parallel assembly line balancing problem (PALBP) consists of two connected sub - problems: assigning of tasks to parallel lines and balancing parallel lines.

Many manufacturing companies have one or more assembly lines. When the market demand is high enough, it is not uncommon to duplicate the entire assembly line. This provides the advantage of the realisation of production rates in given period of time. Another advantage of parallel assembly lines is seen during workstation breakdowns. If equipment problems occur at a workstation, other lines can continue to run.

As an extension of serial lines, bottlenecks are replaced with parallel stations (Fig.4). Tasks performed on parallel stations are the same and throughput is in this way increased (Bukchin et al. 2003).

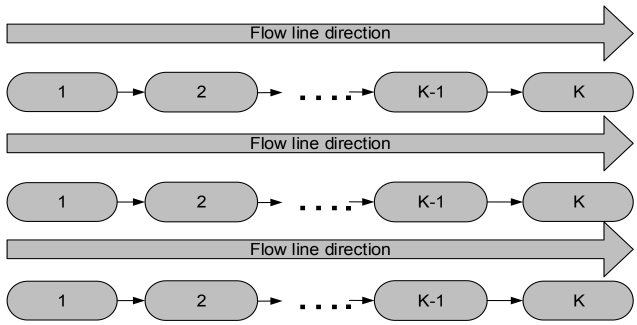

Fig. 3. Parallel assembly lines

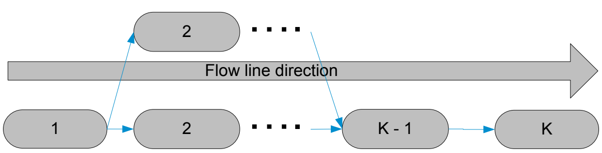

Fig. 4. Parallel stations

Two-sided assembly lines are mainly used for the case of heavy work pieces when it is more convenient to operate on both sides of a work piece rather than rotating it. Instead of a single work-place, there are pairs of directly facing stations such as 1 and 2 in Fig. 5. Such a solution makes the line much more flexible as the work piece can be accessed either from left or right. In comparison to serial lines, two-sided assembly lines may (Lee at el. 2001):

- Shorten the line length.
- Eliminate unnecessary work caused when reaching to the other side of the work piece.

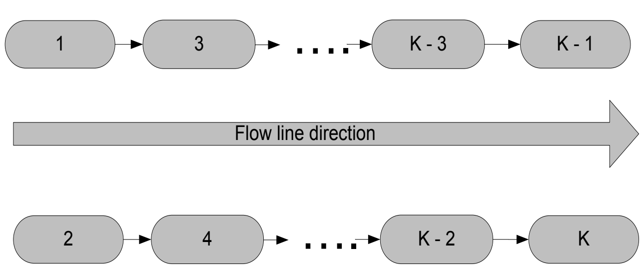

Fig. 5. A two – sided assembly line

In summary, the structure of the assembly line largely defines what type of ALBP must be tackled. The simplest problem involves just the assignment of tasks to the workstations of a serial, single-model paced line. However, in addition, multi/mixed model lines require model sequencing and lot sizing; U lines require the assignment of multiple tasks to workers and parallel lines require a decision about the number of lines to be created.

3. SELECTED HEURISTIC PROCEDURES

In this section heuristics that are useful in the balancing of the serial line and that were used in the illustrative example in section 5 are presented. It is possible to find different heuristic approaches (Scholl , 1998; Talbot et al., 1986; Süer and Dagli, 1994). Most of the heuristics are based on metaheuristic learning techniques, such as: genetic algorithms, tabu search and simulated annealing. In these heuristics different strategies are used to represent assembly line balancing solutions and neighbour generation mechanisms. The common assumptions of the heuristics are listed below:

- only one product is produced on each assembly line,
- precedence graphs for each product are known,
- task processing times of each product are given and operators working at each workstation of the line are multi-skilled.

In the well-known Ranked Positional Weight (RPW) procedure due Helgeson and Bernie (1961) tasks are ranked in descending order of their positional weight (the summation of the task processing time and the processing times of all its successors). The steps involved in this method are as follows:

1. Determine the positional weight (PW) for each task (time of the longest path from the beginning of the task to the end of the precedence graph).
2. Rank the tasks based on their PW's. The task with the highest PW is ranked first.
3. Assign unassigned tasks to the workstations in order of the rank established in step 2.
4. If at any workstation additional time remains after assignment of a task, assign the next succeeding ranked task to the workstation, as long as this does not violate the precedence relationships, and the cycle time is not exceeded.
5. Repeat steps 3 and 4 until all tasks are assigned to workstations.

The immediate update first-fit class of heuristics (IUFF) were proposed by Hackman et al. (1989). Each heuristic in the class depends on a particular numerical score function, to be chosen by the user. Some typical numerical score functions are given in Table 1. The steps involved in the class are as follows:

1. Assign a numerical score to each task.
2. Update the set of available tasks whose immediate predecessors have been assigned.
3. Assign the available task with the highest numerical score to the first station for which the capacity and precedence constraints are not violated.

4. Repeat steps 2 and 3 until all tasks are assigned to workstations.

Table 1. The priority rules used in the case study.

Name	Priority value
MaxRPW	Positional weight
IUFF - NOF (MaxF)	Number of followers
IUFF - NOP (MaxP)	Number of predecessors
IUFF - WET (MaxTime)	Task time

Kilbridge and Wester (1961) proposed a heuristic (KWM) that selects tasks for assignment to workstations according to their position in the precedence diagram.. The steps in the method are:

Preprocessing: Rearrange the precedence graph so those nodes representing tasks of identical precedence are arranged vertically in columns: I, II,... List the tasks in order of their columns, column I at the top of the list. If an element can be located in more than one column, list all columns by the element to show the transferability of the element. The cycle time *c* is then determined by finding all combinations of the primes of $T=\sum t_i$ (the total task processing times). Select a feasible cycle time c. The permissible number of stations is K = roundup ($\sum t_i/c$).

1. To assign tasks to workstations, start with the Column I tasks. Continue the assignment procedure in order of column number until the cycle time is reached.
2. Assign tasks to the station represented by the current column such that the sum of the task time for the station does not exceed c.
3. Delete the assigned tasks from the total number of tasks and repeat step 2.
4. If the station would exceed c owing to the inclusion of a certain task, this task is assigned to the next station.
5. Repeat steps 1 to 4 until all tasks are assigned to workstations.

4. MEASURES OF FINAL RESULTS

Various measures of solution quality have appeared in order to evaluate and compare different ALBP solutions (Scholl 1998). Three of these are presented below

Line efficiency (LE) expresses the percentage utilization of the line. It is expressed as the ratio of total workstation time to the cycle time multiplied by the number of workstations:

$$LE = \frac{\sum_{i=1}^{K} ST_i}{c \cdot K} \cdot 100\% \tag{1}$$

where:
ST_i = the time of workstation I (the sum of the duration times of all the tasks assigned to workstation i),
K = the number of workstations and
c = the cycle time.

The Smoothness index (SI) describes the relative smoothness of assembly line balance. Perfect balance is indicated by smoothness index of 0. This index is calculated in the following manner:

$$SI = \sqrt{\sum_{i=1}^{K} (ST_{max} - ST_i)^2} \tag{2}$$

where:
ST_{max} = max (ST_i), the maximum station time (in most cases, the cycle time).

Line Time (LT) describes the period of time that is needed for the product to be completed on the assembly line:

$$LT = c \cdot (K - 1) + T_K \,. \tag{3}$$

5. THE CASE STUDY

In this section, a numerical instance arising from an actual case study in Poland is solved and discussed. The precedence graph of the single product is presented in Fig. 6. Table 2 shows the processing times (in minutes) of the 23 tasks required to make a single commodity. The Polish company specified a given cycle time of c = 10 minutes and wished to minimize the number of workstations. Thus, we wish to solve an instance of SALBP-1, as defined in Section 1. Among the solutions found, the one that has most favorable combination of quality measures (given as (1), (2), and (3) in Section 4) was finally chosen.

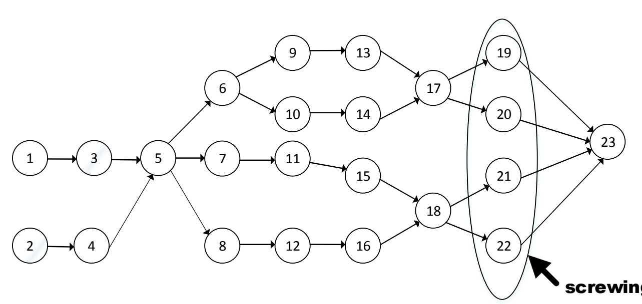

Fig. 6. The precedence graph of the numerical example

Table 2. Processing times (in minutes)

Task	t_i	Task	t_i	Task	t_i
1	5	9	9	17	1
2	7	10	6	18	6
3	8	11	4	19	3
4	2	12	2	20	3
5	5	13	6	21	3
6	3	14	9	22	3
7	1	15	2	23	5
8	8	16	4		

Our analysis started with the ALBP procedure of assigning the 23 tasks involved to workstations. Operations 19 ÷ 22 are screwing activities, all with operation times of 3 minutes. To find feasible solutions to the corresponding ALBP, the RPW, IUFF-NOF, IUFF-NOP IUFF-WET, and KWM heuristics were implemented. The results are given in Table 3. Note that, the higher the value of LE is and the lower the values of K, SI and LT are, the better a solution is. It can be seen in Table 3 that the best solution identified was produced by IUFF-WET, as its values are never inferior to the values of all the other solutions.

Table 3. Solutions to the original numerical instance

Heuristic	K	LE	SI	LT
RPW	13	80.77	9.54	125
IUFF-NOF	12	87.5	6.4	118
IUFF-NOP	12	87.5	6.4	118
IUFF-WET	12	87.5	6.24	118
KWM	13	80.77	9.54	125

The next step in the experiment involved potentially splitting the screwing tasks by determining how far the nut could feasibly be screwed along the bolt. It was discovered that the length of the screw can divided into three positions (1/3(1) or 2/3(2) or total (3) length), as indicated in Fig. 7.

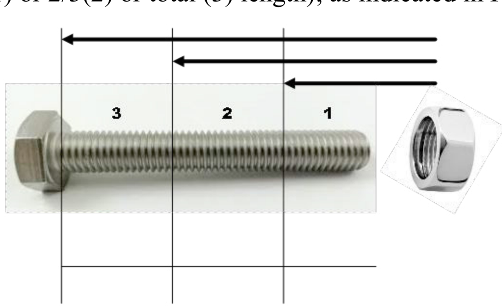

Fig. 7. Screw and nut (three positions of screwing)

Thus, the only feasible possibilities for performing the screwing tasks are as follows:

A – Any screwing task can be left as it is and each nut is fully screwed up in one task. The duration of the single screwing task is 3 minutes (tasks 19 – 22 in Fig. 6).

B – Any screwing task can be split into two subtasks with the nut first screwed two thirds along the bolt and then, as a subsequent subtask, screwed the remaining third (tasks 19a – 19b, 20a – 20b, 21a – 21b, 22a – 22b, in Fig. 8 with duration times 2 and 1 minute, respectively). The duration of the two screwing subtasks is 2 + 1 = 3 minutes.

C – Any screwing task can be split into two subtasks with the nut first screwed one third along the bolt and then, as a subsequent subtask, screwed the remaining two thirds. tasks 19a – 19b, 20a – 20b, 21a – 21b, 22a – 22b, in Fig. 8 with duration times 1 and 2 minutes, respectively). The duration of the two screwing subtasks is 1 + 3 = 3 minutes.

The modified precedence graph, reflecting the possibility of any screwing operation being split by screwing only one third or two thirds along the bolt, is presented in Fig. 8.

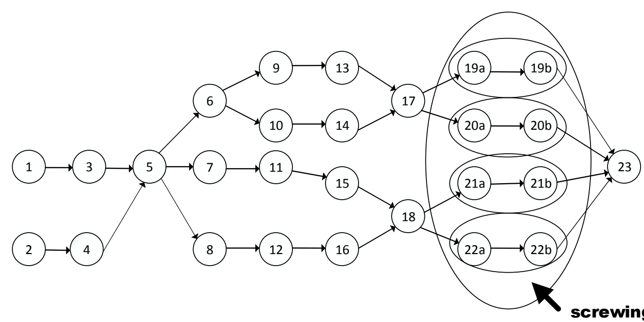

Fig. 8. The precedence graph of the numerical with modified screwing activities

The modified precedence graph contains 27 tasks that can be considered in assembly line balancing procedures. For example, the notation ABCA represents the combination for which the screwing activities are executed in the following order – A (task 19 with duration time 3), B (tasks 20a and 20b with duration times 2 and 1, respectively), C (tasks 21a and 21b) with duration times 1 and 2, respectively) and A (task 22 with duration time 3). All feasible combinations are presented in Table 5. After more than 300 experiments, the best solutions for the combinations in Table 5, obtained by implementing the previously mentioned heuristics, in terms measures (1), (2) and (3), are shown in Table 6. Many of the solutions were very similar to each other and did not improve on the solutions to the original instance shown in Table 3. The assignments of tasks for the solutions to both the original and the modified numerical instances produced by IUFF-WET are given in Table 4. In Fig. 9 the workload of workstations of the assembly line balancing process for combination AAAA is shown. In Fig. 10 the workload of workstations of the assembly line balancing process for combination BBBB is presented. As can be observed, the workload of the stations 1÷9 is the same (as produced by the IUFF – WET heuristic). The workload of only stations 10, 11 and 12 differs because the tasks are split in the final stage of the assembly process.

Table 4. Workstations task assignment for the IUFF – WET heuristic for the AAAA and BBBB combinations

IUFF – WET AAAA combination		IUFF – WET BBBB combination	
Station	Task assignment	Station	Task assignment
1	2,4	1	2,4
2	1,	2	1,
3	3	3	3
4	5,6,7	4	5,6,7
5	9	5	9
6	8,12	6	8,12
7	10,11	7	10,11
8	14,	8	14,
9	13, 16	9	13, 16
10	15,18,17	10	15,18,21a
11	19,20,21	11	22a,17,19a.20a, 19b,20b,21b
12	22,23	12	22b,27

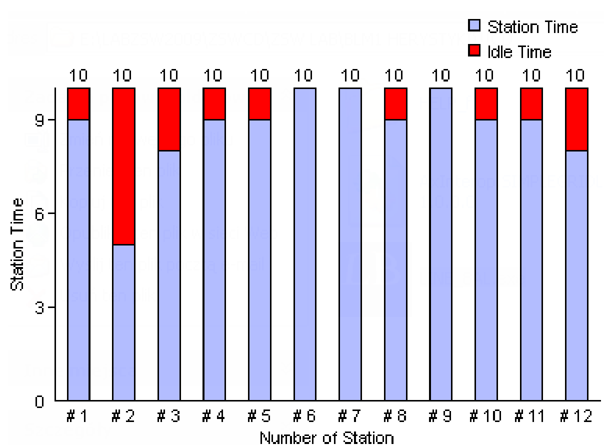

Fig. 9. Station times in the solution to the original numerical instance (AAAA) produced by IUFF-WET

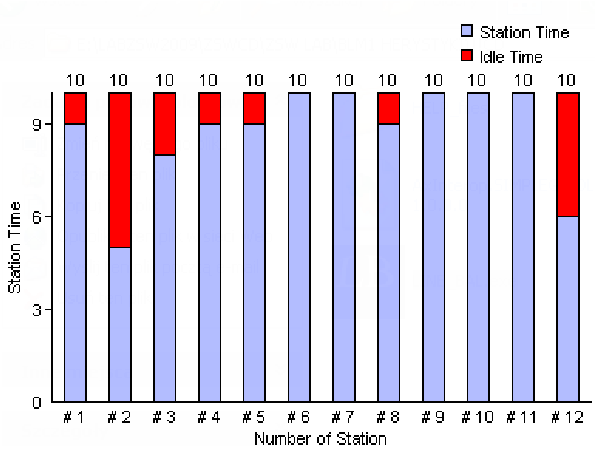

Fig. 10 Station times in the solution to the modified numerical instance (BBBB) produced by IUFF-WET

From Fig. 8 we know task splitting causes an increase in the number of tasks. In comparing the task assignments in Table 4, it is clear that the assignment is different for workstations 10, 11 and 12, with significantly more tasks now assigned to workstation 11. As can be seen in Fig. 10, the new task assignment leads to an improvement of efficiency at workstations 10 and 11, as there is now no idle time at these workstations.

Table 5. All feasible combinations for the numerical instance

AAAA	ACAA	ABBA	AACC	CBCA
BAAA	AACA	ACBA	AABC	CBBB
ABAA	AAAC	ACCA	BBBA	BBBB
AABA	BBAA	ABCA	CBBA	CBBC
AAAB	CBAA	AABB	CCBA	CCCB
CAAA	CCAA	AACB	CCCA	CCCC

Table 6. Best solutions when screwing tasks can be split

Combination	Heuristic	K	LE	SI	LT
BBBB	RPW	12	87.5	7	120
CBBC	KWM	13	80.77	9.43	125
CCCC	RPW	13	80.77	9.54	125
BAAA	RPW	13	80.77	9.54	125
CAAA	NOF	12	87.5	6.71	117
ACCC	RPW	12	87.5	7	120
CCCA	KWM	13	80.77	9.43	125
CCCA	WET	12	87.5	6.56	117
ABAA	NOF	12	87.5	6.4	118
CBBB	NOP	12	87.5	8.66	118
BBBB	WET	12	87.5	9.43	116
CBBC	WET	12	87.5	8.54	118
AACA	NOF	12	87.5	9.64	125

Once again, the higher the value of LE is and the lower the values of K, SI and LT the better a solution is. It can be seen in Table 6 that a better solution is identified as BBBB, produced by IUFF – WET. It is interesting to compare it with the best solution identified for AAAA (no splitting) in Table 3. The values of K and LE are identical in both solutions and the value of SI is superior in Table 3. But there is a small gain of (118 – 116 = 2 minutes) in the line time (LT) for BBBB.

We end this section with a summary of the main results produced for the numerical instance of the case study. The best solution we could find without allowing task splitting was identified by the IUFF – WET heuristic. This solution consists of 12 workstations (9 with idle time), with line efficiency of 87.5%, a smoothness index of 6.24 and a line time of 118. When task splitting is allowed, the best solution we could find was also identified by the IUFF – WET heuristic. It has 12 workstations (7 with idle time), and line efficiency of 87.5% as well, but with a smoothness index of 9.43 and a line time of 116. Thus, the introduction of task splitting has brought about a deterioration of smoothness but an improvement in the line time.

6. CONCLUSIONS AND FURTHER RESEARCH

In the article, the authors present solutions to an assembly line balancing problem with split tasks. In the assumptions given in Section 1, that have been widely adopted for more than 40 years, tasks should be not split between two or more workstations. The description of the production process and the technology are assumed to be very detailed and all tasks have already been divided into basic processes. But it can still happen that such tasks as screwing and drilling can be split. The numerical instance of the assembly line balancing problem comes from an actual Polish factory and the screwing activities occur towards the end of the assembly process. As is expected, splitting the screwing tasks results in a better task assignment to the workstations (resulting in smaller station duration times). As we observed, there is some improvement in line time. as it is possible to obtain the final product two minutes earlier than without splitting. Even such small steps of improvement provide hope that in some cases the result can be even more efficient. We observe that task splitting is likely to be more effective nearer the beginning of the assembly or disassembly line.

ACKNOWLEDGEMENTS

This research was supported by Grant BK265/Rau1/2014 (Poland). While it was being conducted, the first author was supported by a BE MUNDUS (European Commission) grant and the second author by a PVNS (Brazil) grant.

REFERENCES

Baybars, I., (1986). A survey of exact algorithms for simple assembly line balancing problem, *Management Science,* 32, 11-17

Bryton, B., (1954). Balancing of a continuous production line. M.S. Thesis, Northwestern University, Evanston, IL, USA.

Bukchin, J., Rubinovitz, J., (2003). A weighted approach for assembly line design with station paralleling and equipment selection, *IIE Transactions*, 35, 73-85

Erel,E., Sarin, S.C., (1998). A survey of the assembly line balancing procedures, *Production Planning and Control,* 9, 414-434

Fonseca, D.J., Guest, C.L., Elam, M., Karr, C.L., (2005). A fuzzy logic approach to assembly line balancing, *Mathware & Soft Computing,*12,57-74

Göcken, H., Ağpak, K., Benzer, R., (2006). Balancing of parallel assembly lines, *International Journal of Production Economics* 103, 600-609

Gutjahr, A.L., Neumhauser, G.L., (1964). An algorithm for the balancing problem, *Management Science*, 11, 23-35

Hackman, S.T., Magazine, M.J., Wee, T.S., (1989). Fast, Effective Algorithms for Simple Assembly Line Balancing Problems, *Operations Research,*Vol.37, No. 6, 916-924

Helgeson, W. B., Birnie, D. P., (1961). Assembly line balancing using the ranked positional weighting technique, *Journal of Industrial Engineering,* 12 (6), 394-398

Ismail, N., Esmaelian, G.R., Hamedi, M., Sulaiman,.S., (2011). Balancing of parallel assembly lines with mixed model product, *International Conference on Management and Artificial Intelligence IPEDR* 6, 120-124

Kao, E.P.C., (1976). A preference order dynamic program for stochastic assembly line balancing, *Management Science*, 22, 19-24

Kilbridge, M.D, Wester, L. (1961). A heuristic method of assembly line balancing, *Journal of Industrial Engineering,* 12 (4), 292-298

Lee, T.O., Kim, Y., Kim, Y.K., (2001). Two-sided assembly line balancing to maximize work relatedness and slackness, *Computers & Industrial Engineering,* 40, 273-292

Miltenburg, J., (1998). Balancing U-lines in multiple U-line facility, *European Journal of Operational research* 109, 1-23

Miltenburg, J., (2001). U-shaped production lines: a review of theory and practice, *International Journal of Production Economics* 70, 201-214

Miltenburg, J., Wijngaard, J., (1994). The U-line balancing problem, *Management Science* 40 (10), 1378-1388

Nakade, K., Nishiwaki, R, (2008). Optimal allocation of heterogeneous workers in a U-shaped production line, *Computer & Industrial Engineering,* 54, 432-440

Salveson, M.E., (1955). The assembly line balancing problem, *Journal of Industrial Engineering,* 62-69

Scholl, A., (1998). *Balancing and sequencing of assembly line*, Physica- Verlag, Heidelberg

Scholl, A., Voβ, S., (1996). Simple Assembly Line Balancing: Heuristic Approaches, *Journal of Heuristics*, 2, 217-244

Scholl, A., Becker C., (2006). State of art exact and heuristic solutions procedures for simple assembly line balancing, European Journal of Operational Research, 168, 666-693

Shwetank, A., Rajeev, J., R, Mishra, P.K., Yadav , H.C., (2013). A heuristic approach for U-shaped assembly line balancing to improve labor productivity, *Computers & Industrial Engineering,* 64, 895-901

Süer, G.A., Dagli, C., (1994). A knowledge based system for selection of resource allocation rules and algorithms*, Handbook of Expert System Applications in Manufacturing: Structures and Rules*, 108-147

Talbot, F. B., Patterson, J.H., Gehrlein, W.V., (1986). A comparative Evaluation of Heuristics Line Balancing Techniques, *Management Science,* 32, 430-454