Just in Time Strategy in Balancing of Two-Sided Assembly Line Structure

W. Grzechca

Abstract —**In this paper a problem of two-sided assembly line balancing problem and Just in Time strategy is considered. Problem of assigning tasks to assembly line is known more than 60 years. Different balances occur in different quality of final results. Smoothness index (SI), time of line (LT) and line efficiency (LE) help to estimate the ends solutions. Additionally detailed idle times are described. In production systems products are assembled in required quantity directly to the market. One of the critical elements in Just in Time strategy is smooth flow production process. Author tries to connect assembly line balancing problem with this strategy. A numerical example is calculated and conclusions are presented.**

Index Terms—**Assembly line, estimation of final results, just in time strategy, two-sided assembly line structure.**

I. Introduction

More than 100 years ago the idea of assembly line was introduced in Ford factory in Detroit. It was designed to be an efficient, highly productive way of manufacturing a particular product. Now in XXI century this way of assembly of final products is still very common and we can find it in many companies over the world. The basic assembly line consists of a set of workstations arranged in a linear fashion, with each station connected by a material handling device (transfer lines, roller conveyors, cranes etc.). The components are processed depending on set of tasks and they are performed at each station during a fixed period called as cycle time. The time it takes to complete a task at each workstation is known as the process time [1]. 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 [2]. In order for the assembly line to maintain a certain production rate, the sum of the processing times at each station must not exceed the stations' cycle time]. If the sum of the processing times within a station is less than the cycle time, idle (delay) time is said to be present at that station [3]. One of the main issues concerning the development of an assembly line is how to arrange the tasks to be performed. The tasks are allocated to workstations according to known precedence relationships (very often in form of precedence graph) and specific restrictions which aim to optimize one or more objectives. A feasible assignment of tasks to workstations should guarantee that the following constraints: (1) each task must be assigned to exactly one workstation, (2) all precedence relationships among tasks must be satisfied and (3) the total process time of all the tasks assigned to a workstation cannot exceed the cycle time. The problem of assigning tasks to workstations in such a way that some objectives are optimized is called assembly line balancing problem – ALBP. We can recognize generally two types of ALBP - minimizing number of workstations for a given cycle time (TYPE 1 of ALBP) or minimizing the cycle time for a given number of workstations (TYPE 2 of ALBP). The assembly line balancing problem (ALBP) originated with the invention of the assembly line. Helgeson et al [4] were the first to propose the ALBP, and Salveson [5] was the first to publish the problem in its mathematical form. However, during the first forty years of the assembly line's existence, only trial-and-error methods were used to balance the lines. Since then, there have been numerous methods developed to solve the different forms of the ALBP. Salveson [5] provided the first mathematical attempt by solving the problem as a linear program. Gutjahr and Nemhauser [6] showed that the ALBP problem falls into the class of NP-hard combinatorial optimization problems. This means that an optimal solution is not guaranteed for problems of significant size. Therefore, heuristic methods have become the most popular techniques for solving the problem. But we should underline that many studies on assembly line including exact solution methods and heuristics have been reported in the literature. The detailed reviews of such studies are given by Baybars [2], Erel and Sarin [3], and Scholl and Becker [7]. In the literature assembly line is classified as: straight assembly line, assembly line with parallel stations, U-shaped assembly line or two-sided assembly line. Other classification takes into account number of products which are produced on the line (single model line, multi-model line and mixed-model line).

II. Just in Time Strategy in Two – Sided Assembly Line

Just in time (JIT) is considered as one of the most effective management systems in manufacturing since it has been introduced. Based on Toyota Production System there are three elements should be considered in order to implement JIT. But as many companies all over the world are trying to implement JIT, the implementation elements have been varies. JIT can be considered as manufacturing techniques that produce and deliver part of final product in just amount needed. The concept of JIT works in reverse direction where final assembly line is taken as a starting point. In [8] we can find a list of several critical elements which can be identified during JIT strategy implementation. One of them is smooth production flow which is strongly connected with assembly line balancing problem. Flow or physical layout of the

Manuscript received December 20, 2012; revised February 15, 2013.
Waldemar Grzechca is with the Silesian University of Technology, 44-100 Gliwice, ul. Akademicka 2A, Poland (e-mail: waldemar.grzechca@polsl.pl).

DOI: 10.7763/IJMMM.2013.V1.14

production facilities is arranged to make the process flow is streamlined as possible. Smoothen the flow, workplace organization is needed Adding idea to smooth flow production is when lines are run continuously and parts are move piece by piece down a line. Another way to obtain smooth flow is a good balance of assembly lines. In next section of this article an idea how to find a good balance which doesn't collapse the strategy of JIT will be discussed. A two-sided structure of assembly line will be taken into account (Fig. 1)

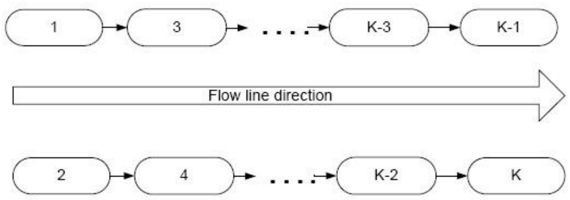

Fig. 1. Two-sided assembly line structure.

Two-sided assembly lines are typically found in producing large-sized products, such as trucks and buses. Assembling these products is in some respects different from assembling small products. Some assembly operations prefer to be performed at one of the two sides [9]. The consideration of the preferred operation directions is important since it can greatly influence the productivity of the line, in particular when assigning tasks, laying out facilities, and placing tools and fixtures in a two-sided assembly line [10]. A two-sided assembly line in practice can provide several substantial advantages over a one-sided assembly line [9]. These include the following: 1) it can shorten the line length, which means that fewer workers are required, 2) it thus can reduce the amount of throughput time, 3) it can also benefit from lowered cost of tools and fixtures since they can be shared by both sides of a mated-station, and 4) it can reduce material handling, workers movement and set-up time, which otherwise may not be easily eliminated. These advantages give a good reason for utilizing two-sided lines for assembling large-sized products.

A line balancing problem is usually represented by a precedence diagram as illustrated in Fig. 4. A circle indicates a task, and an arc linking two tasks represents the precedence relation between the tasks. Each task is associated with two values: t_i is the task processing time and *d* (= L, R or E) is the preferred operation direction. L and R, respectively, indicate that the task should be assigned to a left- and a right-side station. A task associated with E can be performed at either side of the line. While balancing assembly lines, it is generally needed to take account of the features specific to the lines. In a one-sided assembly line, if precedence relations are considered appropriately, all the tasks assigned to a station can be carried out continuously without any interruption. However, in a two-sided assembly line, some tasks assigned to a station can be delayed by the tasks assigned to its companion [9]. In other words, idle time is sometimes unavoidable even between tasks assigned to the same station. Consider, for example, task *j* and its immediate predecessor *i*. Suppose that *j* is assigned to a station and *i* to its companion station. Task *j* cannot be started until task *i* is completed. Therefore, balancing such a two-sided assembly line, unlike a one-sided assembly line, needs to consider the sequence-dependent finish time of tasks.

This notion of sequence dependency further influences the treatment of cycle time constraint. Every task assigned to a station must be able to be completed within a predetermined cycle time. In a one-sided assembly line, this can readily be achieved by checking the total operation time of tasks assigned to a station. Therefore, a task not violating any precedence constraints can be simply added to the station if the resulting total amount of operation time does not exceed the cycle time. However, in a two-sided assembly line, due to the above sequence-dependent delay of tasks, the cycle time constraint should be more carefully examined. The amount of time required to perform tasks allocated to a station is determined by the task sequences in both sides of the mated-station as well as their operation time. It should be mentioned that two-sided assembly line is a special case of single assembly line. Therefore it is possible to use some procedures and measurements, which were for simple assembly line developed.

III. Heuristic Method for Two – Sided Assembly Line

As it was written, Bartholdi [9] was the first researcher who considered and described two-sided assembly line balancing problem – TALBP. Kim *et al*. [10] developed a genetic algorithm for TALBP. Lee et al. [11] generated an assignment procedure for TALBP in order to maximize work relatedness and slackness. Baykasoglu and Dereli [12] proposed an ant colony method for TALBP with zoning constraints. Hu *et al*. presented a station oriented enumerative algorithm that is integrated with Hoffmann heuristic to develop a system for solving TALBP [13]. The all mentioned methods belong to heuristic methods. In 2007 it was proposed an exact method – branch and bound procedure to balance the two-sided assembly line optimally. Below is presented heuristic method introduced by Bartholdi. It helps to assign tasks using simple assignment rules.

A task group consists of a considered task *i* and all of its predecessors. Such groups are generated for every un–assigned task. As mentioned earlier, balancing a two–sided assembly line needs to additionally consider operation directions and sequence dependency of tasks, while creating new groups [10] While forming initial groups IG(i), the operation direction is being checked all the time. It's disallowed for a group to contain tasks with preferred operation direction from opposite sides. But, if each task in initial group is E – task, the group can be allocated to any side. In order to determine the operation directions for such groups, the rules (direction rules DR) are applied:

DR 1. - set the operation direction to the side where tasks can be started earlier and DR 2. -the start time at both sides is the same, set the operation direction to the side where it's expected to carry out a less amount of tasks (total operation time of unassigned L or R tasks).

Generally, tasks resulting from "repeatability test" are treated as starting ones. But there is exception in form of first iteration, where procedure starts from searching tasks (initial tasks IT), which are the first ones in precedence relation.

To those who are considered to be the first, the next tasks will be added, (these ones which fulfil precedence constraints). Whenever new tasks are inserted to the group *i*, the direction, cycle time and number of immediate predecessors are checked. If there are more predecessors than one, the creation of initial group j comes to the end. When set of initial groups is created, the last elements from those groups are tested for repeatability. If last element in set of initial groups IG will occur more than once, the groups are intended to be joined – if total processing time (summary time of considered groups) is less or equal to cycle time. Otherwise, these elements are deleted. In case of occurring only once, the last member is being checked if its predecessors are not contained in Final set FS. If not, it's removed as well. So far, FS is empty. Whenever two or more initial groups are joined together, or when initial group is connected with those one coming from FS – the "double task" is added to initial tasks needed for the next iteration. At the end of each iteration, created initial groups are copied to FS. In the second iteration, second step, we may notice that predecessor of last task coming from IG(1) is included in Final Set, FS(2). The situation results in connecting both groups under holding additional conditions:

Side{IG(1)} = Side{FS(2)},
Time + time < cycle.

After all, there are no more IT tasks, hence, preliminary process of creating final set is terminated. Though, all of candidates may be assigned equally, the only one group may be chosen. Which group it will be – for this purpose the rules helpful in making decision, will be defined and explained below:

AR 1. Choose the task group FS(*i*) that may start at the earliest time.
AR 2. Choose the task group FS(*i*) that involves the minimum delay.
AR 3. Choose the task group FS(*i*) that has the maximum processing time.

In theory, for better understanding, we will consider a left and right side of mated – station, with some tasks already allocated to both sides. In order to achieve well balanced station, the AR 1 is applied, because the unbalanced station is stated as the one which would probably involve more delay in future assignment. This is the reason, why minimization number of stations is not the only goal, there are also indirect ones, such as reduction of unavoidable delay. This rule gives higher priority to the station, where less tasks are allocated. If ties occurs, the AR 2 is executed, which chooses the group with the least amount of delay among the considered ones. This rule may also result in tie. The last one, points at relating work within individual station group by choosing group of task with highest processing time. For the third rule the tie situation is impossible to obtain, because of random selection of tasks. The implementation of above rules is strict and easy except the second one. Shortly speaking, second rule is based on the test, which checks each task consecutively, coming from candidates group FS(*i*) – in order to see if one of its predecessors have already been allocated to station. If it has, the difference between starting time of considered task and finished time of its predecessor allocated to companion station is calculated. The result should be positive, otherwise time delay occurs. Having rules for initial grouping and assigning tasks described in previous sections, we may proceed to formulate formal procedure of solving two – sided assembly line balancing problem [12].

Let us denote companion stations as *j* and *j'*,
D(*i*) – the amount of delay,
Time(*i*) – total processing time (Time{FS(*i*)}),
S(*j*) – start time at station j,

Step 1: Set up *j* = 1, *j'* = *j* + 1, S(*j*) = S(*j'*) = 0, U – the set of tasks to be assigned.

Step 2: Start procedure of group creating, which identifies FS = {FS(1), FS(2), …, FS(n)}. If FS = ∅, go to step 6.

Step 3: For every FS(*i*), *i* = 1,2, … , n – compute D(*i*) and Time(*i*).

Step 4: Identify one task group FS(*i*), using AR rules

Step 5: Assign FS(*i*) to a station *j* (*j'*) according to its operation direction, and update S(*j*) = S(*j*) + Time(*i*) + D(*i*). U = U – {FS(*i*)}, and go to STEP 2.

Step 6: If U ≠ ∅, set *j* = *j'* + 1, *j'* = *j* + 1, S(*j*) = S(*j'*) = 0, and go to STEP 2, Otherwise, stop the procedure.

IV. Measures of End Balance

Some measures of solution quality have appeared in line balancing problem. Below are presented three of them [2] and [7].

Line efficiency (LE) shows the percentage utilization of the line. It is expressed as ratio of total station 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: *K* - total number of workstations, *c* - cycle time.

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

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

where: ST_{max} - maximum station time (in most cases cycle time), ST_i - station time of station *i*.

Time of the line (LT) describes the period of time which is need for the product to be completed on an assembly line:

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

where: *c* - cycle time, *K* - total number of workstations. T_k – load time of the last station.

In two – sided assembly line balancing method within mated-stations, tasks are intended to perform its operations at the same time to the both sides. In consequence, modification has to be introduced to line time parameter which is the consequence of parallelism. We must treat those stations as two double ones (mated-stations), rather than individual ones S_k. Accepting this line of reasoning, new formula is presented below:

where:

Km – number of mated-stations,

K – number of assigned single stations,

$t(S_K)$ – processing time of the last single station.

As far as smoothness index and line efficiency are concerned, its estimation, on contrary to LT, is performed without any change to original version. These criterions simply refer to each individual station, despite of parallel character of the method.

But for more detailed information about the balance of right or left side of the assembly line additional measures will be proposed:

Smoothness index of the left side

$$SI_L = \sqrt{\sum_{i=1}^{K}\left(ST_{maxL} - ST_{iL}\right)^2} \quad (5)$$

where:

SI_L- smoothness index of the left side of two-sided line,

ST_{maxL}- maximum of duration time of left allocated stations,

ST_{iL}- duration time of *i*-th left allocated station.

Smoothness index of the right side

$$SI_R = \sqrt{\sum_{i=1}^{K}\left(ST_{maxR} - ST_{iR}\right)^2} \quad (6)$$

where:

SI_R- smoothness index of the right side of two-sided line,

ST_{maxR}- maximum of duration time of right allocated stations,

ST_{iR}- duration time of *i*-th right allocated station.

In this point, it’s worth to mention about a special case, when mated-station includes instead of two stations, just one. Such a situation takes place, where one station is loaded to a certain point that not allows for assigning any more tasks for this part of the line. As the result, one station stays empty. In this case we got an assembly line which is a structure of incomplete two-sided assembly line. It is possible to estimate the balanced line in two ways: as a single line with parallel stations or incomplete two-sided line. As we can see there are some differences in final measurements of balanced line. The reason is that using heuristic methods we design two-sided assembly line. These kinds of heuristics are very sensitive to cycle time value. Some final balances for different value of cycle time don’t represent a complete two-sided structure. It is very difficult estimate the quality of such final result (empty station represents 100% idle time or empty station is ignored and we don’t take it into account). Very often very useful is knowledge about idle times in the assembly line system. In TALBP we can notice idle time directly after opening new workstation, unavoidable idle times between tasks on the same workstation and idle time which occurs before we close workstation and our product moves to the next machine. Below a new idea about estimation of idle times in TALBP is shown (Fig. 2)

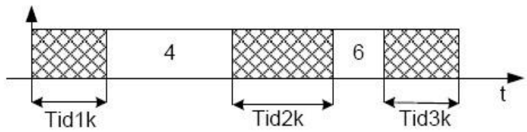

Fig. 2. Idle times in k[th] workstation.

$$LT = c \cdot \left(Km - 1\right) + Max\left\{t(S_K),t(S_{K-1})\right\} \quad (4)$$

where:

T_{id1k} – idle time after opening the k^{th} workstation

T_{id2k} – unavoidable idle time between tasks on the same k^{th} workstation (sum if there are more the one delay),

T_{id3k} – idle time before closing the k^{th} workstation.

Equation (7) gives the maximum value of idle time T_{idMAX} in two-sided assembly line balancing problem:

$$T_{idmax} = \max(T_{id1k}, T_{id2k}, T_{id3k}) \quad (7)$$

This value gives us the knowledge about the longest delay in system and helps to find workstations with idle times. Three elements set allows very easy to identify all types of idle times in the discussed assembly structure.

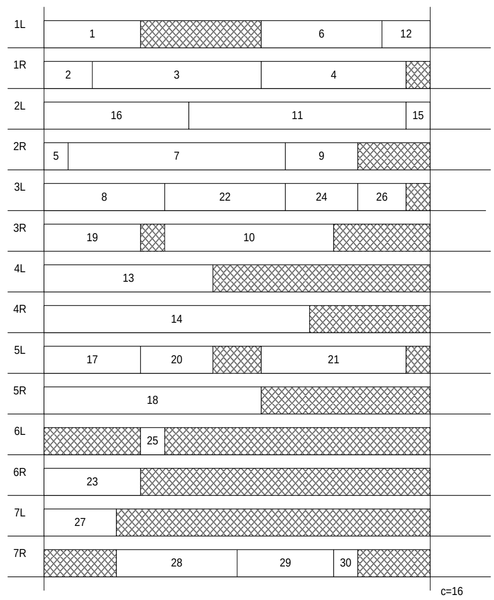

Fig. 3. Detailed balance of two- sided assembly line for cycle time c = 16.

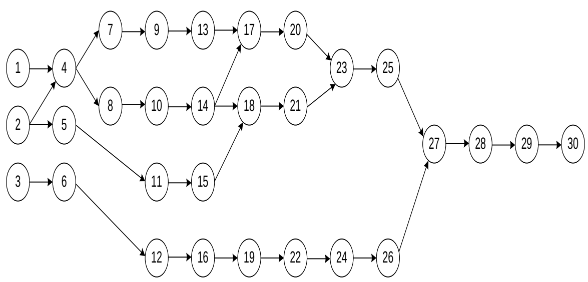

Fig. 4. Precedence graph of numerical example.

V. NUMERICAL EXAMPLE

A 30 tasks numerical example is considered (Fig. 4). A precedence graph is presented and duration times and directions of operations are given (see Fig. 3. and Table I and Table II).

TABLE I: DURATION TIMES AND DIRECTIONS OF OPERATIONS OF NUMERICAL EXAMPLE

Task	Time	Direction	Task	Time	Direction
1	4	L	16	6	L
2	2	E	17	4	L
3	7	R	18	9	R
4	6	R	19	4	E
5	1	E	20	3	E
6	5	L	21	6	L
7	9	E	22	5	L
8	5	L	23	4	R
9	3	R	24	3	E
10	7	E	25	1	E
11	9	E	26	2	E
12	2	E	27	3	L
13	7	L	28	5	R
14	11	R	29	4	E
15	1	E	30	1	E

TABLE II: FINAL RESULTS OF TWO – SIDED ASSEMBLY LINE BALANCING PROBLEM FOR DIFFERENT CYCLE TIMES

Station	c=14			c=15			c=16			c=17			c=18		
	T_{id1}	T_{id2}	T_{id3}	T_{id1}	T_{id2}	T_{id3}	T_{id1}	T_{id2}	T_{id3}	T_{id1}	T_{id2}	T_{id3}	T_{id1}	T_{id2}	T_{id3}
1L	0	0	0	0	0	3	0	5	0	0	5	3	0	5	4
1R	0	0	1	0	0	0	0	0	1	0	5	1	0	0	3
2L	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
2R	0	0	1	0	0	2	0	0	3	0	0	2	0	0	2
3L	0	0	1	0	0	0	0	0	1	0	0	5	0	0	0
3R	0	0	7	0	1	3	0	1	4	0	0	0	0	0	1
4L	0	0	0	0	0	8	0	0	9	0	0	10	11	0	0
4R	0	0	3	0	0	4	0	0	5	0	0	6	0	0	7
5L	0	0	10	0	2	0	0	2	1	0	2	2	18	18	18
5R	0	0	5	0	0	6	0	0	7	0	0	8	0	0	9
6L	0	4	0	4	0	10	4	0	11	4	0	12	0	0	12
6R	0	3	4	0	0	11	0	0	12	0	0	13	18	18	18
7L	14	14	14	0	0	**12**	0	0	**13**	0	0	**14**	4	0	10
7R	0	0	4	3	0	2	3	0	3	3	0	4	0	4	0

VI. CONCLUSION

In this paper a two – sided assembly line balancing problem and just in time strategy are discussed. Different and very fast changes of market demand causes that assembly line has to be balanced very often. A good balance allows obtaining small delays and a good flow of materials and semi products. A choice of appropriate value of cycle time allows controlling the final products quantity and additionally the character of delays in manufacturing system. The detailed knowledge of delays together with mentioned in this article measures of final results give managers the possibility to plan the production rate and to estimate the quality of the end balance.

ACKNOWLEDGMENT

This work was supported by Grant BK219/Rau1/2012.

REFERENCES

[1] R. J. Sury, "Aspects of assembly line balancing," *International Journal of Production Research*, vol. 9, pp. 8-14, 1971.

[2] I. Baybars, "A survey of exact algorithms for simple assembly line balancing problem," *Management Science,* vol. 32, pp. 11-17, 1986.

[3] E. Erel and S. C. Sarin, "A survey of the assembly line balancing procedures," *Production Planning and Control,* vol. 9, pp. 414-434, 1998.

[4] W. B. Halgeson and D. P. Birnie, "Assembly line balancing using the ranked positional weighting technique," *Journal of Industrial Engineering,* vol. 12, pp. 18-27, 1961.

[5] M. E. Salveson, "The assembly line balancing problem," *Journal of Industrial Engineering,* vol. 6, pp. 62-69, 1955.

[6] A. L. Gutjahr and G. L. Neumhauser, "An algorithm for the line balancing problem," *Management Science*, vol. 11, pp. 308-315, 1964.

[7] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *European Journal of Operation Research*, vol. 158, pp. 666-693, 2006.

[8] A. A. Nurul Fateha, M. Y. Nafrizuan, and Y. A. Razlan, "Review on elements of JIT implementation," in *Proc. International Conference on Automotive, Mechanical and Materials Engineering (ICAMME'2012),* Penang, Malaysia, May 19-20, 2012, pp.118-124.

[9] J. J. Bartholdi, "Balancing two-sided assembly lines: a case study," *International Journal of Production Research,* vol. 23, pp. 403-421, 1993.

[10] K. Y. Kim, W. S. Song, and J. H. Kim, “A mathematical model and a genetic algorithm for two-sided assembly line balancing,” *Computers and Operations Research*, vol. 36, pp. 853-865, 2009.

[11] T. O. Lee, Y. Kim, and Y. K. Kim, “Two-sided assembly line balancing to maximize work relatedness and slackness,” *Computers and Industrial Engineering*, vol. 40, pp. 273-292, 2001.

[12] A. Baykasoglu and T. Dereli, “Two-sided assembly line balancing using an ant-colony based heuristic,” *International Journal of Advanced Manufacturing Technology*, vol. 36, pp. 582-588, 2008.

[13] X. Hu, E. Wu, and Y. Jin, “A station oriented enumerative algorithm for two-sided assembly line balancing problem,” *European Journal of Operational Research*, vol. 186, pp. 435-440, 2007.

Waldemar Grzechca was born in Gliwice in Poland in 1964. He was graduated at the Silesian University of Technology in 1989 and he started to work at the same time at the Department of Automation Control, Electronics and Computer Science. He focused his research on accuracy of robots trajectories and robots applications in manufacturing companies. Next he has moved his research and interest to manufacturing systems. He studied different structures of machines (sequencing and scheduling in single machine and parallel machines configurations, job shop system, flow shop system, etc.). He is especially interested in assembly lines balancing problem. He investigates different types of lines and focuses on estimation of final results of balance of single and two-sided lines. He is author of more than 60 conference and journal papers which deals with assembly lines problems and an editor of a book titled Assembly Line – Theory and Practice.