Journal of Machine Enginnering. 12, No. 3, 2012

assembly line balancing problem,
heuristic methods,
final results estimation

Waldemar GRZECHCA[1]

DIFFERENT STRUCTURES OF MANUFACTURING SYSTEMS IN LOW PRODUCTION DEMAND

Since more than 100 years ago a simple assembly line was introduced in the American Ford automotive factory. Nowadays we can find papers and books with description of different structures and different kinds of production. This article deals with the problem of low production demand which is coming from the market and considers three different manufacturing structures: single (straight) assembly line, U-line and assembly round table. The description of all above mentioned structures is given. The fundamental assumptions according to balancing problem are shown. Selected heuristics methods for solving assembly line balancing problem are described. Advantages and disadvantages of these structures are considered. Also numerical examples are calculated and final results are estimated (smoothness index, line efficiency, time of line, number of turns for rotating round table). At the end the conclusions and remarks are presented.

1. INTRODUCTION TO ASSEMBLY LINE BALANCING PROBLEM

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 at the assembly line in which a task is performed on the product usually joining one or more new parts. These tasks can be performed by machinery, robots, and/or human operators. Once the product enters a station, the task is then performed, and the product is moved to the next station. The time it takes to complete a task at each operation 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 to be set so that the desired amount of the 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 (including the transfer time from station to station) must not exceed the stations' cycle time [3]. 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 [4].

---

[1] Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science

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 [5]. For the manufacturing of any product, there are some sequences of tasks that must be followed. Since the process time of the different tasks is usually not the same, an imbalance occurs which generates losses. Therefore one tries to balance the processing times. The assembly line balancing problem (ALBP) originated with the invention of the assembly line. Helgeson and Birnie [6] were the first to propose the ALBP, and Salveson [7] was the first to publish the problem in its mathematical form. An ALBP generally consists of finding a feasible line balance, i.e., an assignment of each task to a station such that the cycle time constraints, the precedence constraints and possible further restrictions are fulfilled. The most popular ALBP is called Simple Assembly Line Balancing Problem (SALBP*)*. It simplifies the more general ALBP by introducing the following assumptions [8-10]:

- mass-production of one homogeneous product,
- all tasks are processed in a predetermined mode (no processing alternatives exist),
- paced line with a fixed common cycle time according to a desired output quantity,
- the line is considered to be serial with no feeder lines or parallel elements,
- the processing sequence of tasks is subject to precedence restrictions,
- deterministic (and w.l.o.g. integral) task times,
- no assignment restrictions of tasks besides precedence constraints,
- a task cannot be split among two or more stations,
- all stations are equally equipped with respect to machines and workers.

Two goals can be considered in addition to the precedence relations between the tasks: the minimization of the number of workstations for a given cycle time (SALBP-I) and the minimization of the cycle time for a given number of workstations (SALBP-II).However, during the first forty years of the assembly line’s existence, only trial-and-error methods were used to balance the lines [4]. Since then, there have been numerous methods developed to solve the different forms of the ALBP. Salveson [7] provided the first mathematical attempt by solving the problem as a linear program. Gutjahr and Nemhauser [11] have shown 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.

2. LOW DEMAND OF PRODUCTION

The volume of production is not a widely discussed topic in literature. There are numerous articles about mixed-model assembly systems however they do not investigate the problem of low product demand. A formulation of a problem given in [12] should give an idea about it. J. Bukchin indicates that it’s long gone, when everybody was buying a black painted Ford T as long as it was cheap. Back then, high productivity was achieved by introducing a perfectly single model with no additional features. Nowadays, the life cycle of a products is relatively short and the demand for varied product is high. Consequently, a set of similar products needs to be assembled in relatively low volume. The goal to such an approach is a flexible response to shorter product life cycles, low to medium production volumes, changing demand patterns and a higher variety of product models and options.

The conditions for such an installation are:

- assembly-to-order production,
- low product demand (low volume production),
- number of tasks greater than number of stations,
- lack of mechanical conveyance,
- highly skilled workers.

It might be extended with conditions given by Heike [13]:

- flexible fixtures,
- flexible tooling,
- delivery of material.

Such conditions give a good base for an assembly system robust to demand changes. Having a good balancing algorithm is a goal in this case. When the demand for a set of similar products is insufficiently high in order to install a complete assembly line a solution given in [14] might be used. Most of the authors use combined precedence diagrams in order to reduce multiple models into a single model. As the plant layout, the majority uses a straight line in some cases allowing parallel workstations for omitting the bottleneck effects. What more, some allow duplicating stations in series. Authors' investigation of U-shaped lines indicate their benefits over traditional serial lines. Some of them are:

- improvement in labour productivity,
- job enlargement for human operators,
- great interaction between operators,
- reduction in number of required workstations,
- lead time contraction,
- increase of flexibility.

They suggest [15] this kind of lines in case of number of tasks less than 30 and 10 stations. Fixed position layout should be taken into account dealing with heavy workpieces as it is more convenient to switch the operators places rather than i.e. rotating the part [13]. Generally, when set-up times required between different versions are significantly high a job shop layout suits the best.

3. ASSEMBLY ROTATING ROUND TABLE

The model and the procedure discussed in this paragraph bases on [14] Battini introduces a mixed-model assembly system consisting of a rotating assembly table with a fixed number of stations. It is a semi-automated system therefore some stations are occupied by human operators; some by machines and other are free. Human operators are indicated by "O" while automated ones as "A". The resource assignment is assumed to have no limitations, every operator or machine can be placed at any station of the table. The product assembled with such a system is assumed to be homogenous with some additional features that enable the creation of a joint precedence diagram with known tasks' durations.

The rotating table is a multi-turn one; as a matter of fact a batch of one single product is completed in *n* number of turns, with $n \geq 2$. The table is an example of an unpaced synchronous line controlled assembly system. It means that all the tasks performed by operators need to be completed before the shift of the table. It is assumed that it has a pneumatic motion and all operators need to press a button as information that they finished their task. If all the tasks are finished the table switches their position with switch time $t_s \geq 2s$. Every switch of the table moves the workpiece to the following station – one station at each table switch (Fig. 1).

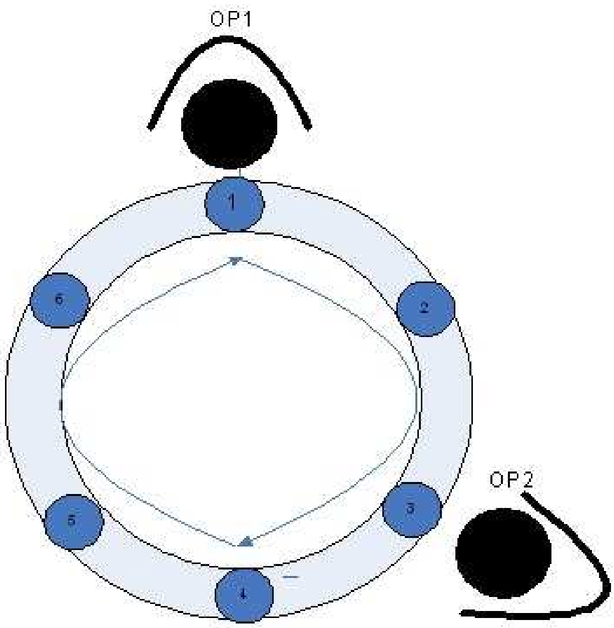

Fig. 1. Example of assembly round table (two human operators and six stations)

The assumptions of the round table are:

1. the table is of multi-turn type,
2. precedence diagrams of all model types can be accumulated into a single combined precedence diagram,
3. the line production policy is "assemble-to-order",
4. workpieces are fixed on the table and there is only one workpiece at the station of the table at a time,
5. each station has only either one operator or one actuator,
6. idle operators cannot be used to help the operators of other stations,
7. the table switches only when all the opened stations have finished their job.
8. the first task of the cycle is the load of all the workpieces of the same batch on a table and is always assigned to the first operator,

9. the last task of the cycle is the download of the assembled units and can be assigned to any operator.

The objectives for this assembly system are:

1. optimize the load balancing of each station activated in the rotating table,
2. optimize the resource positioning in order to minimize the entire make span of the assembly batch, and consequently, the average cycle time.

4. STRAIGHT AND U-SHAPED ASSEMBLY LINE STRUCTURE

Assembly lines can be classified in two general groups as straight (serial, traditional) assembly lines with single and multi/- mixed products and U-type (U-shaped) assembly lines with single and multi/mixed products. The main difference between them is the design of assembly lines. U- type ALBP is one of the generalizations of Simple Assembly Line Balancing Problem SALBP. The U- type ALBP is introduced and modeled by Miltenburg and Wijngaard [16]. Figure 2 shows the main difference between the straight-line and U- type layouts. When using a straight-line layout, operators must work on a contiguous length of the line. When using a U- type layout, operators are allowed to work across both "sides" of the line. This is shown with the U- type layout in Fig. 2 where the operator from station 1 performs tasks on the front side of the line, travels to the back side to complete tasks which are assigned to the other station (i.e. station N-1 , and then returns to the front side of the line to begin the next cycle. As it can be seen from Fig. 2, the U- type line allows more possibilities on how to assign tasks to workstations; the number of stations needed for a U- type line layout is *never* more than the number of stations needed for the traditional straight line [17]. The reason for this is that in the traditional ALBP for a given workstation, the set of possible assignable tasks is confirmed by those tasks whose *predecessors* have already been assigned to workstations, whereas in the U- type line problem the set of assignable tasks is determined by all those tasks whose *predecessors* or *successors* have already been assigned [17].

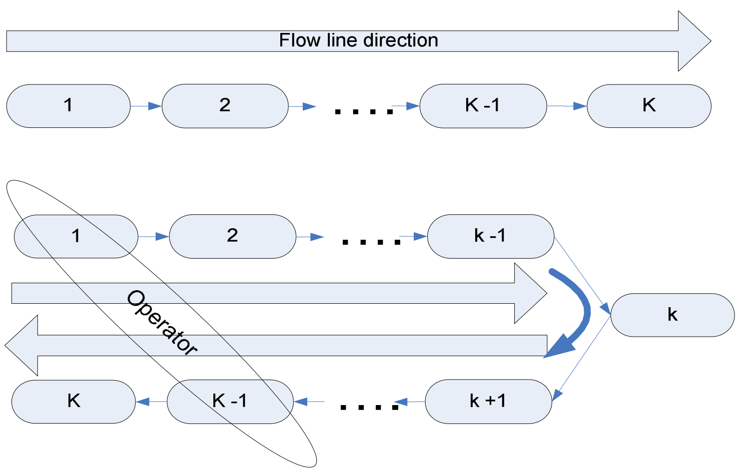

Fig. 2. Straight and U shaped assembly line structures

5. ESTIMATION OF FINAL RESULTS OF BALANCING PROBLEM

Some measures of the performance of line balancing have appeared in literature. Below are presented three of them [7],[18].

Line efficiency (LE) shows the percentage utilization of the line. It is expressed as the 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:

ST_i - processing time of station i,

K - total number of workstations,

c - cycle time.

The smoothness index (SI) describes the 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 - processing time of station i.

K - total number of workstations.

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

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

where:

c - cycle time,

K - total number of workstations,

T_K - station time of the last station K

The average cycle time (C) for a rotating round table is calculated due to the formula:

$$C = \frac{\sum_{l=1}^{Z} \sum_{k:k \in AS_Z} \{\max[t(S_K)]_Z + t_s\} \cdot X_k}{K} \tag{4}$$

where:

$t(S_k)$ - station load (the sum of operation times of all operation assigned to station k),

AS_Z - set of stations activated in turn z,
Z - 1,..,Z are table runs,
K - total number of stations,
t_s - switch time of the table,
X_k - distance in switches between the major load station and each activated station in turn z.

6. NUMERICAL EXAMPLE

In this chapter an illustrative example of a serial assembly line, an U- shaped assembly line and an assembly rotating round table is shown. First an example with 8 tasks for the final product is considered (Fig. 3., Table 1). Next a 20 tasks example is calculated. In both cases for finding the end solution of balance a heuristic procedure (Update Immediately First Fit – Number of Followers) was implemented.

6.1. 8 TASKS EXAMPLE

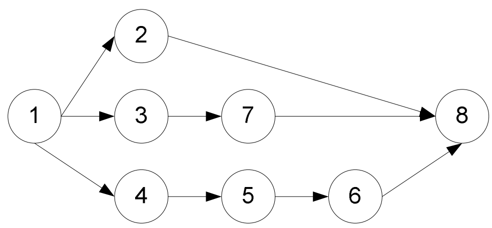

Fig. 3. Precedence graph of numerical example – 8 tasks

Table 1. Operation time of numerical example – 8 tasks

Task i	Time t_i	Task i	Time t_i	Task i	Time t_i	Task i	Time t_i
1	18	3	6	5	7	7	11
2	13	4	9	6	14	8	2

We consider a serial assembly line with two workers it means with workstation, see Fig. 4. It is a problem knows as Simple Assembly Line Balancing Problem Type 2 when the number of stations is given and the value of cycle time is to be calculated, see Fig. 4.

$$c = \left\lceil \frac{\sum_{i=1}^{N} t_i}{K} \right\rceil \quad (5)$$

where:

c - cycle time of serial assembly line,
t_i - operation time of task i,
N - number of tasks,
K - number of stations

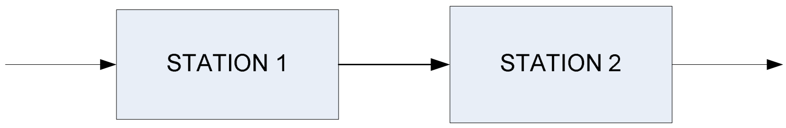

Fig. 4. Straight line of 2 stations

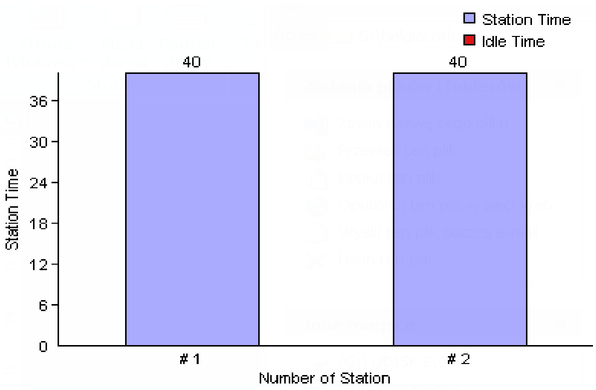

Fig. 5. Balance of serial line for calculated example

The calculated cycle time is 40 (the total operation time is 80) so we got the final solution of balanced line: Station 1 {1, 4, 3, 7) and Station 2 {5, 2, 6, 8), Fig. 5. The solution is optimal (mostly we obtain using a heuristic method only feasible solution) and calculated measures are: SI = 0, LE = 100% and LT = 80). Next we consider an assembly rotating table with 2 human operators and six workstations. We obtain the final results for 6 cases which mean that we calculate the average cycle time for six different locations of human workers. Starting from position 1 and 2 (Fig. 6.) we relocate the second operator to location 3, 4, 5 and 6. Operator 1 is always assigned to station 1. Relocation of Operator 2 causes that the distance between both workers changes (Fig. 7.).

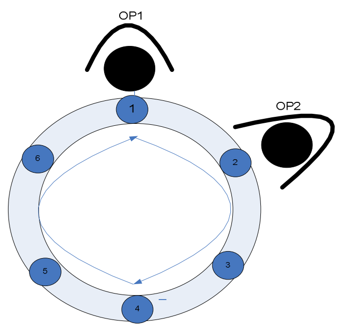

Fig. 6. Location of human workers at an assembly rotating round table (1st case)

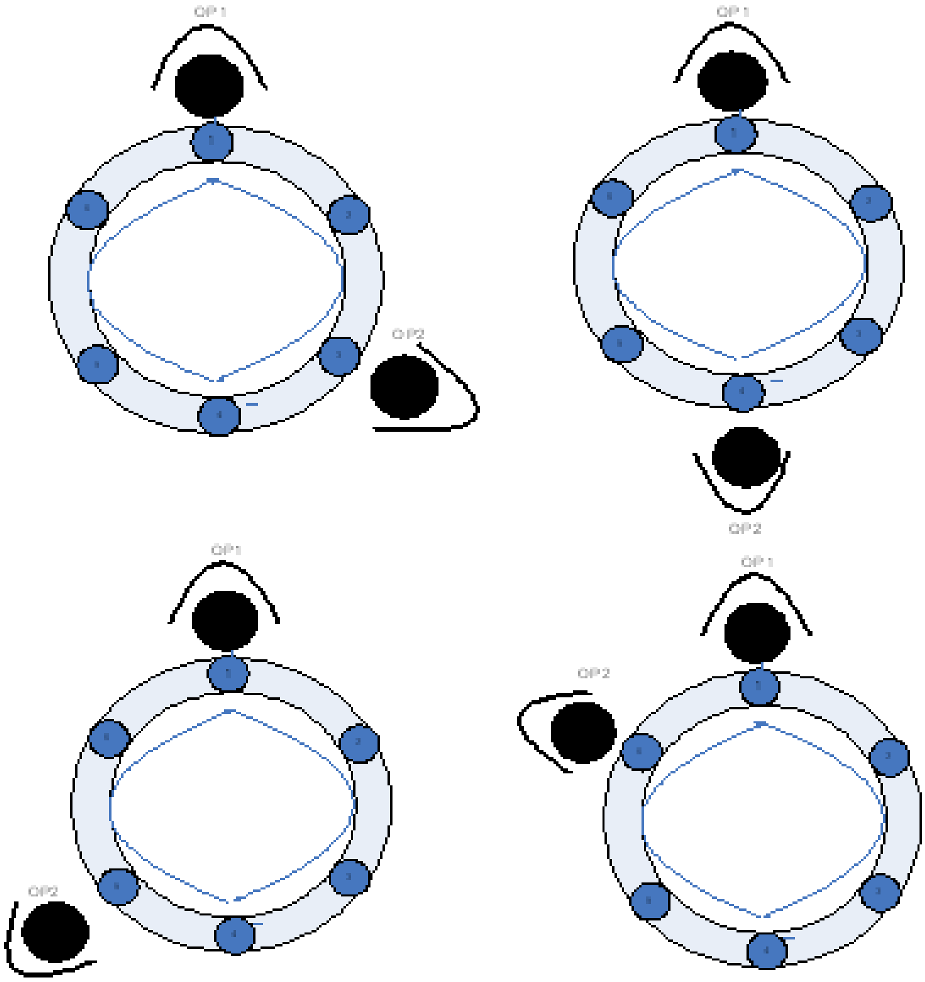

Fig. 7. Different locations of workers at an assembly rotating round table (2 workers and 6 stations)

The main goal of the described algorithm [14] is to balance the assembly system in order to minimize the time between loading the first workpiece and unloading the final product. Initially, when the table is empty, the first human resource Op_j becomes the current operator Op_c. The table is loaded with the critical task CT as selected from the available tasks list. The (Critical Task ÷ is defined among the available tasks j as critical when it has the major number of successor tasks or, in case of equity, when his t_j is the maximum of all t's. It follows that the current operator Op_c is assigned to the first station occupied by a human operator and this station's load, denoted as S_{Opc}, is initially equal to the length of CT task assigned. As the workpiece has been loaded, balancing procedure starts. The upper bound of station's load needs to be calculated in order not to overload the current operator. This upper bound is defined as the maximal task time t_{max}. Only for the first activated resource t_{max} is considered to be the assigned task time. For the successively activated resources Op_{max} is chosen among all activated stations.

They express the COST and SAVING parameters as a portion of the cycle time. If task λ belongs to AT, the SAVING factor is calculated due to formula 6:

$$SAVING(\lambda) = \frac{(t_{\lambda} + t_{s}) \cdot Y}{K} \tag{6}$$

where:

t_{λ} - operation time of task λ,

t_{s} - switch time of the table,

Y - distance in switches between Op_c and Op_f,

K - number of stations.

For the COST parameter three cases need to be considered. In case 1, when the cycle time is conditioned by the load of the slower resource OP_{max} until the end of a turn of the table and the additional task slows down the table of X switches, which separate Op_{max} from Op_c following formula is used:

$$\text{If } \ t(S'_{Opc}) \leq t(S_{Opmax}) \ \text{ then } \ COST(\lambda) = \frac{t_{\lambda} \cdot X}{K} \tag{7}$$

where:

$t(S'_{Opc})$ – station load of current operator after assignment of task λ,

$t(S_{Opmax})$ – station load of the operator with major load,

t_{λ} - operation time of task λ,

X - distance in switches between Op_c and the last activated operator,

K - number of stations.

In case 2, when the current operator becomes the slowest of the table after assignment of task λ and the delay is equal to the weight average of the differences between S'_{Opc} and S_k greater then Op_c station time, before the assignment of task λ the following formula is used:

$$\text{If } \ t(S'_{Opc}) > t(S_{Opmax}) \text{ and } t_{\lambda} > [t(S'_{Opc}) - t(S_{Opmax})] \text{ then}$$

$$COST(\lambda) = \frac{\sum_{k: t(S_k) > (S_{Opc})} \{[t(S'_{Opc}) - t(S_k)] \cdot X_k\}}{K} + \frac{t_{\lambda} \cdot X}{K} \tag{8}$$

where:

$t(S'_{Opc})$ – station load of current operator after assignment of task λ,

$t(S_{Opmax})$ – station load of the operator with major load,

$t(S_k)$ – station load of station k,

t_{λ} - operation time of task λ,

X - distance in switches between Op_c and the last activated operator,

K - number of stations.

The last case 3 takes place, when the current operator already represents the slowest resource of the table and the further assignment of task λ causes a delay of the table equal to its operation task.

$$\text{If } \ t(S'_{Opc}) > t(S_{Opmax}) \ \text{ and } \ t_{\lambda} < [t(S'_{Opc}) - t(S_{Opmax})] \ \text{ then}$$

$$COST(\lambda) = t_{\lambda} \tag{9}$$

where:

$t(S'_{Opc})$ – station load of current operator after assignment of task λ,

$t(S_{Opmax})$ – station load of the operator with major load,

t_{λ} - operation time of task λ.

Having COST and SAVING values for all available tasks a preferable task list is created. The task with greater SAVING or in case of equality the one with greater number of successors F_j is the chosen task. At any time ‘t’ the table is occupied by an operator with maximum load, denoted as Op_{max}. The stations which finished their tasks become idle until deactivation of Op_{max}..This idle time is the gap between the maximum workload $t(Op_{max})$ and the resource workload $t(Op_j)$.

In Table 2 are also presented results of different locations of workers at rotating round table assembly when Operator 1 and Operator 2 are assigned to different stations. The best average cycle time for the assembly on a rotating round table is 53 time units and it occurs always when Operator 1 and Operator 2 are located next to each other. In this case we need to execute only two turns. The final solution is: Operator 1 executes tasks 1 and 6 and Operator 2 executes tasks 2, 3, 4, 5, 7 and 8. Additionally we can calculate the time when the final product is ready to unload from the assembly system. In our case the ready product leaves the system after 216 units of time.

Table 2. Operation time results of numerical example.

	OP 1	OP2	Average Cycle Time	Turns
1	Station 1	Station 2	53	2
2	Station 1	Station 3	61	3
3	Station 2	Station 3	53	2
4	Station 1	Station 4	56	3
5	Station 2	Station 4	61	3
6	Station 3	Station 4	53	2
7	Station 1	Station 5	58	3
8	Station 2	Station 5	56	3
9	Station 3	Station 5	61	3
10	Station 4	Station 5	53	2
11	Station 1	Station 6	70	2
12	Station 2	Station 6	58	3
13	Station 3	Station 6	56	3
14	Station 4	Station 6	61	3
15	Station 5	Station 6	53	2

6.2. 20 TASKS EXAMPLE

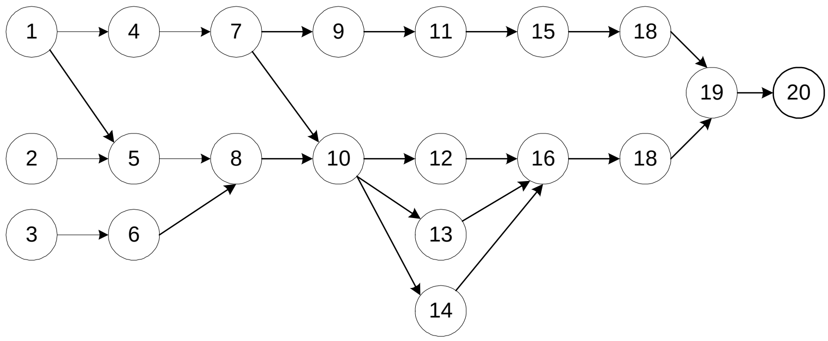

Fig. 8. Precedence graph of 20 tasks example

Low demand products with less than 30 tasks are often assembled in serial or U-shaped assembly lines. The precedence graph and the operation times of tasks for an example of 20 tasks are presented in Fig. 8. and Table 3. Next the final results of balance for cycle time c=7 of the mentioned two structures are shown (Table 4).

Table 3. Operation time of numerical example

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

Table 4. Final results of numerical example

SERIAL LINE		ULINE	
Station	Tasks	Station	Tasks
1	1, 2	1	1, 2
2	4, 5	2	4, 5
3	3, 7	3	3, 7, 20
4	6	4	6
5	9, 11	5	9, 11
6	8, 15	6	8, 15
7	10, 12	7	10, 12
8	13, 14	8	13, 14

9	16, 17	9	16, 17
10	18, 19	10	18, 19
11	20		
SI = 2,09		SI = 0,84	
LT = 71		LT = 70	
LE = 84,42%		LE = 92,86%	

7. CONCLUSIONS

In the article three assembly systems were considered. First assembly lines (straight and U-shaped) were presented. Next the assembly on a rotating round table was shown. The problem seems interesting for low product demand (low number of tasks and less than 30 tasks problem). Known procedures of solving balance of line structures allow to get very easy optimal or near optimal solution for the two station line. Products with less than 30 tasks can be assembled on U- shaped manufacturing lines. It results very often in less number of workstations. We need to remember that the low number of tasks appear from limited space between machines. The investigated assembly rotating round table allows quick changes of assembling different product. The described heuristic procedure improves the result of average cycle time from 70 to 53 time units. This kind of assembly table takes benefits from a layout described in paragraph 3 of this article dealing with their disadvantages such as monotony, boredom, operators overload and bad communication. Different measures of final result (smoothness index, line efficiency, line time or average cycle time) simplify the choice of the most appropriate solution.

REFERENCES

[1] SURY, R.J., 1971, *Aspects of assembly line balancing*, Int. Journal of Production Research, 9, 8-14.
[2] BAYBARS, I., 1986, *A survey of exact algorithms for simple assembly line balancing problem,* Management Science, 32, 11-17.
[3] FONSECA D.J., GUEST C.L., ELAM M., KARR C.L., 2005, *A fuzzy logic approach to assembly line balancing,* Mathare & Soft Computing, 12, 57-74.
[4] EREL E, SARIN S.C., 1998, *A survey of the assembly line balancing procedures*, Production Planning and Control, 9, 34-42.
[5] KAO, E.P.C., 1976, *A preference order dynamic program for stochastic assembly line balancing,* Management Science, 22, 19-24.
[6] HALGESON W. B., BIRNIE D. P., 1961, *Assembly line balancing using the ranked positional weighting technique*, Journal of Industrial Engineering, 12, 18-27.
[7] SALVESON, M.E., 1955, *The assembly line balancing problem*, Journal of Industrial Engineering, 62-69.
[8] BAYBARS I., 1986, *A Survey of Exact Algorithms for the Simply Assembly Line Balancing Problem,* Management Science, 32, 909-932.
[9] SCHOLL A., 1999, *Balancing and sequencing of assembly lines,* Physica-Verlag.
[10] BOYSEN N., FLIEDNER M, SCHOLL A., 2007, *A classification of assembly line balancing problem,* European Journal of Operational Research, 184, 39-55.

[11] GUTJAHR, A.L., NEUMHAUSER G.L., 1964, *An algorithm for the balancing problem*, Management Science, 11, 23-35.

[12] BUKCHIN J.,.M. DAR-EL M. E.,RUBINOVITZ J., 2002, *Mixed model assembly line design in a make-to-order environment*, Computers & Industrial Engineering, 41, 405-421.

[13] HEIKE G., RAMULU M., SORENSON E., SHANAHAN P., MOINZADEH K., 2001, *Mixed model assembly alternatives for low-volume manufacturing*: *The case of the aerospace industry*, Int. J. Production Economics, 72, 103-120.

[14] BATTINI D, FACCIO M, FERRARI E., PERSONA A., SGARBOSSA F., 2006, *Design configuration for a mixed-model assembly system in case of low product demand*, Springer-Verlag, London Limited.

[15] AASE G.R., OLSON J.R., SCHNIEDERJANS M.J., 2004, *U-shaped assembly line layouts and their impact on labor productivity: An experimental study*, European Journal of Operational Research, 156, 698-711.

[16] MILTENBURG J., WIJNGAARD J., 1994, *The U-line line balancing problem*, Management Science, 40, 1378–1388.

[17] AJENBLIT, D. A., WAINWRIGHT, R. L., 1998, *Applying genetic algorithms to the U-shaped assembly line balancing problem*, In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, 96–101.

[18] GRZECHCA W., 2010, *Assembly Systems for Low Product Demand - Estimation of Final Result*,. In Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics, Funchal, 259-264.