Waldemar GRZECHCA*

METHODOLOGY FOR COST ORIENTED ASSEMBLY LINE BALANCING PROBLEMS

Abstract: Assembly lines are special flow-lines production systems which are typical in the industrial production of high quantity standardized commodities. Assembly line balancing is based on evenly distribution of operations between workstations so that each idle time of the machines was minimal. It is assumed that operation times on the machines and the precedence relationship existing between the operations which results from the technology of the production line and the cycle time or the number of machines are known. In the paper short description about general assembly line problem and cost-oriented is described. Next the author presents description of used algorithms. A numerical example is calculated and the results are discussed. Presented paper shows that cost oriented assembly line balancing problem differs from time oriented problem and the differences are underlined.

Keywords: assembly line, optimization, production, scheduling, manufacturing systems.

1. Introduction of assembly line balancing problem

Designing a system of production has always been an important issue in industrial engineering. This problem becomes even more important as a result of international competition and an extremely rapid progress in manufacturing technologies. At the present time productively systems are characterized by extremely short production times for all models, the high level of automation, the needs of new technologies and high expenditures for the construction of the assembly lines. Those features made designers aware of new issues, thereby forcing them to create of new lines and frequent improvements of lines as well as the creation of appropriate and accurate software tools which will make designing the lines easier. These problems pay more attention to scientific community, but many important questions remain unanswered. There are no adequate tools such as CAD, whose main task would be to optimize the efficiency of production systems, therefore the development of tools is becoming a problem not only innovative, but it would have been applied in many industries. Assembly

* Institute of Automation Control, The Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland, e-mail: waldemar.grzechca@polsl.pl

line balancing problem (ALBP) consists of a finite set of tasks, where each of them has a duration time and precedence relations, which specify the acceptable ordering of the tasks. One of the problems inherent in organizing the mass production is how to group the tasks to be performed on a workstation so as to achieve the desired level of efficiency. Line balancing is an attempt to locate tasks to each workstation on the assembly line. The basis ALB problem is to assign a set of tasks to an ordered set of workstations, so that the precedence relationships were satisfied, and performance factors were optimized. Assembly line balancing is based on evenly distribution of operations between workstations so that each idle time of the machines was minimal. It is assumed that operation times on the machines, the precedence relationship existing between the operations which results from the technology of the production line and the cycle time or the number of machines, are known. To complete the balance process, each operation must be assigned only once and to only one workstation (Fig. 1).

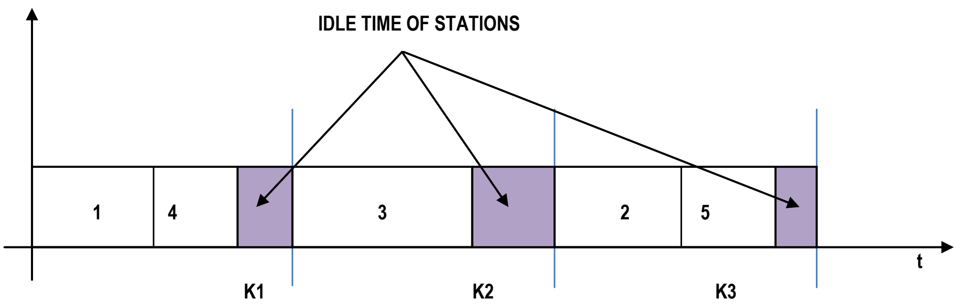

Fig. 1. *Assembly Line Balancing Problem*

2. Basic assumption of assembly line balancing problem

Design of the production system is based on information about the product and technological capabilities of available equipment and accessories. Thus, the problem of assembly line balancing (ALB) is closely related to the allocation of operations to the respective workstations, which form the assembly lines satisfying the beforehand optimization criteria. This problem can be divided into different subcategories such as:

- appropriate choice of equipment depending on the machine operations,
- balancing and dimensioning of workstations (assigning tasks to the station),
- dimensioning of storage space,
- an appropriate choice of frequency and quantity of raw materials and internal transport.

Firstly, define what we mean by the assembly line. The assembly line is a system whose units are assembly stations. It consists of a finite set of operations. Each of them has a set of duration time and the precedence relationships which specifies acceptable arrangement of operations. Balancing the line tends to a uniform allocation of operations to each workstation from which the line is built [1]. So, the assembly line consists of a number of (work) stations arranged along a conveyer belt or similar mechanical material handling equipment. The workpieces (tasks) are consecutively launched down the conveyer belt and are steadily moved from station to station. At each station, a certain part of the total work, necessary to manufacture the product, is performed [2]. Line balancing is a task of particular importance (the designer must answer the question of how to create and then use the line). Appropriate balance not only affects the quality of the manufactured product, but also on the time when it will be delivered to the market. Appropriate allocation of operations is the most interesting task from a point of view of efficiency of the line. As the first problem, to increase the efficiency of line was taken by B.Bryton, however, the first publication on this subject was developed and released in 1955 by M.E.Salveson. ALB problem involves grouping assembly operations in admissible subsets that create jobs. It is understood that the operation is set with the order and time relationship. There are two variants of ALB [1]:

1) for a given cycle must specify a minimum number of subsets of operations, which forms workstations,
2) for a given number of workstations a minimum cycle must be determined.

According to the classical method of balancing the line for a set cycle-time, the tasks' times should be so allocated to the workstations that the time loss (the idle time of machines) as short as possible (in the ideal case would not be at all). In the theoretical considerations the alternative devices are not taken into account (otherwise the problem is called "supply problems"). To solve the simple assembly line balancing problem (SALBP), there are known heuristic and exact methods. To this day many algorithms have been published to solve the assembly line balancing problem (ALBP). Almost all the algorithms are related to the first variant ALB. Some algorithms take into account the random times of assembly operations. It is worth recalling that the issue of ALB (for the first variant), determines not only the minimum number of subsets of operations, but also the operations belonging to these subsets. In order to obtain a final solution for ALB, both methods - the exact one, which gives optimal solution (due to the minimum number of stations Type 1 or due to the minimum cycle time TYPE 2) and approximate methods, which result in an acceptable solution, are being used for the calculation.

In the assembly process, there are many constraints, which lead to the creation of models. For this reason, comparison of algorithms for various models is difficult. To solve problems with multiple constraints ALB heuristic methods have been used so far. Exact algorithms are for the model without additional restrictions. Balancing the assembly line very often refers to a single line balancing. The simplest version of this issue is well known as "backpack problem", which should accommodate as many different objects in a number of identical "containers" of a certain capacity. Assuming constancy of the time the tasks assigned to the machines backpack problem can be treated as a SALB, in which we strive to open the smallest number of stations, while avoiding overflow (exceeding the cycle time).

In 1986 Baybars proposed the following assumptions for the problem of SALB [3]:

- all input is known and certain,
- tasks are indivisible (cannot be separated into several machines),
- tasks must be performed in accordance with the requirements of the manufacturing process,
- all tasks must be performed,
- each station can perform any task,
- tasks' times are independent of the filling stations which are carried out,
- tasks can be performed on any of the stations,
- line is serial,
- line is designed for the unique model of the product,
- cycle time is known and unchanging, or
- number of stations is known and unchanging.

Duration of the tasks is crucial in the ALBP. It depends on many factors such as complexity of operations, qualification of operators and reliability of used machinery. In the case of simple tasks, the differences may be small, however, started operations can cause much longer delays. In the case of employees-oriented line, the work efficiency is influenced even more factors, including motivation, teamwork within a group or individual skills.

3. Measures of quality of alb problem

The assembly line balancing is the best, if for each workstation, the sum of elementary operations' times is equal to the cycle time of workstation. Unfortunately, this is not always possible. That's way, the measures have been created, that allow for comparing the methods used to solve these types of tasks [4], [5], [6], [7].

Measurement used is:

- effectiveness of the line – Line Efficiency (LE);
- coefficient of smoothness – Smoothness Index (SI);
- line time – Time (T).

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:

$$LE = \frac{\sum_{i=1}^{M} ST_i}{c \cdot M} \cdot 100\% \qquad (1)$$

where: M – total number of workstations, c – cycle time ST_i – station time of station i.

Smoothness Index (SI) – describes relative smoothness for a given assembly line balance. Perfect balance is indicated by smoothness index 0. Smaller value of SI, more the line is balanced.

This index is calculated in the following manner:

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

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

Line time (T) – is a factor depending on the number of stations. If this time smaller, the better line will be balanced.

$$T = (K - 1) \cdot c + ST_K \tag{3}$$

where: c – cycle time, K – total number of workstations, ST_K – last time station.

Discussing those measures of quality, great attention should be paid to their usefulness in the assessment of feasible solutions to the assembly line balancing problem. The line time directly informs us about when the final product leaves the factory line. It is obvious that a solution with less line time is a better solution. For the purposes of discussing problem smoothness index has been created, which informs us about existence of idle time on the line. A value greater than zero, means that such a time appeared on the line. Due to the performance of operation times dependency on this index, we can use this factor for comparison the solution for the same task for different methods and different cycle time. The effectiveness of the line makes sense for solutions with different numbers of workstations.

4. Cost oriented assembly line balancing problem

Original approach to the problem of balancing the line was to focus exclusively on minimizing the number of stations by placing them as many tasks as possible, having idle time in mind, that it has the greatest impact on costs incurred during the assembly process. Another approach to the problem offered Matthias Amen [4] published by himself research work in 2000. Amen proved that the cost-oriented balancing line problem is the generalized problem searching for a solution with the minimum number of stations. Nevertheless, achieving one of these goals makes it the second requirement won't be satisfied. The reason for this is the approach to calculating costs. Amen assumed that the cost of the station is not the sum of all operations assigned and performed on it, but only the cost of the most advanced operations. This is the result of analyzing that the most complex tasks should be allocated to the most skilled workers, or the best machine, while simpler tasks can be performed by workers with less experience, or by simpler machines.

4.1. List of symbols used in the formal problem statement [4]

I number of tasks [–]
h, i index for the tasks, $h, i \in 1, ..., I$
M number of stations [–]
m index for the stations, $m \in 1, ..., M$
m_i station, to which task i is assigned to, $i = 1(1)I$
I_m^s set of tasks assigned to station $m, m = 1(1)M$
c cycle time [TU/PU]
k total relevant costs per product unit [MU/PU]
k_m^s cost rate of station $m, m = 1(1)M$, [MU/TU]
k_i^t cost rate of task $i, i = 1(1)I$, [MU/TU]
d_i^t duration of task $i, i = 1(1)I$, [TU/PU]
d_m^s duration of the operation in station $m, m = 1(1)M$, [TU/PU]
V_i set of tasks which are immediate (= direct) predecessors of task $i, i = 1(1)I$

(Note: Dimensions: TU – time units, PU – product units, MU – monetary units. Labels: s – station, t – task, w – wage rate.)

4.2. Amen's algorithm. Expanded time-oriented ALBP into cost-oriented ALBP

The algorithm proposed by Amen, says that to the tasks are being assigned to the stations with the lowest cost [4]. Then, the next task can be assigned to the same station (if tasks preceded it, have already been assigned) with the duration time, which (summed with times already assigned tasks) does not exceed the cycle time and while the cost of which is equal to or less than the costs of the tasks already assigned to the station. Cost factor has been introduced to estimate an appropriate cost index. Sometimes the most important criterion is cost, not the duration of the process, so really the number of idle times, whether the duration of performing the tasks can be meaningless.

4.3. Assumptions of Amen's algorithm

Algorithm to assign all tasks to the stations and conflict resolution:

1) Sort the tasks by the costs from the smallest to largest.
2) If the station is empty, select the task with the lowest cost. If it is not empty, select the task with the highest cost, but not more than the cost of the station.
3) If the costs are the same, choose the task that has more time.
4) If times are the same, choose the task that has a smaller ID.
5) Jump to 2, until all the tasks have been assigned.

4.4. Scholl's and Becker's improved algorithm

Despite the effectiveness of Matthias Amen's algorithm in 2003, Scholl and Becker demonstrated in their work [8], that the maximally load station (as Amen called his algorithm) has its drawbacks and it does not always lead to the optimal solution. In some cases, Amen's method [4] gives much worse results than the standard method of time. Scholl and Becker proposed improvements to their method that were intended to prevent such situations. In the algorithm [8] the principle of not increasing the cost of the station is not applied (the next task may have either higher or lower cost), but primarily the task with the highest cost is chosen (in Amen's method opposite), then the next task with the highest possible cost is being looked to assign it as the next task to the station. If there are several tasks with the same cost, this task with a maximum time is chosen (which does not exceed a total cycle time) and then (if the times are the same), a task with the lowest ID.

4.5. Assumptions of Scholl's and Becker's algorithm

Algorithm to assign all tasks to the stations and conflict resolution:

1) Sort tasks by the costs from largest to smallest.
2) Choose a job with the largest cost.
3) If the costs are the same, choose the job that has more time.
4) When times are the same, choose the job that has a smaller ID.

4.6. Cost simply algorithm

Algorithm to assign all tasks to the stations and conflict resolution:

1) Sort tasks by the costs from the largest to smallest.
2) Choose a job with the largest cost.
3) If the costs are the same, choose the job that has less time.
4) When times are the same, choose the job that has a smaller ID.

5. Numerical example

In this chapter a numerical example will be presented. Precedence graph is shown in Figure 2. Blue field indicates number of task, green field means cost of the operation and yellow gives us duration time of operation.

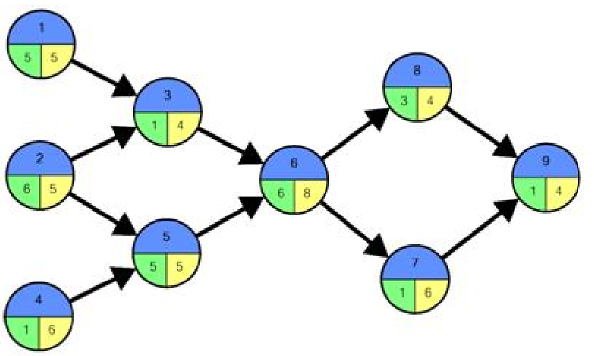

Fig. 2. *Precedence graph of numerical example*

In the experiment the value of cycle time is changed from 11 to 10. Below the result of assigning tasks using different methods are shown.

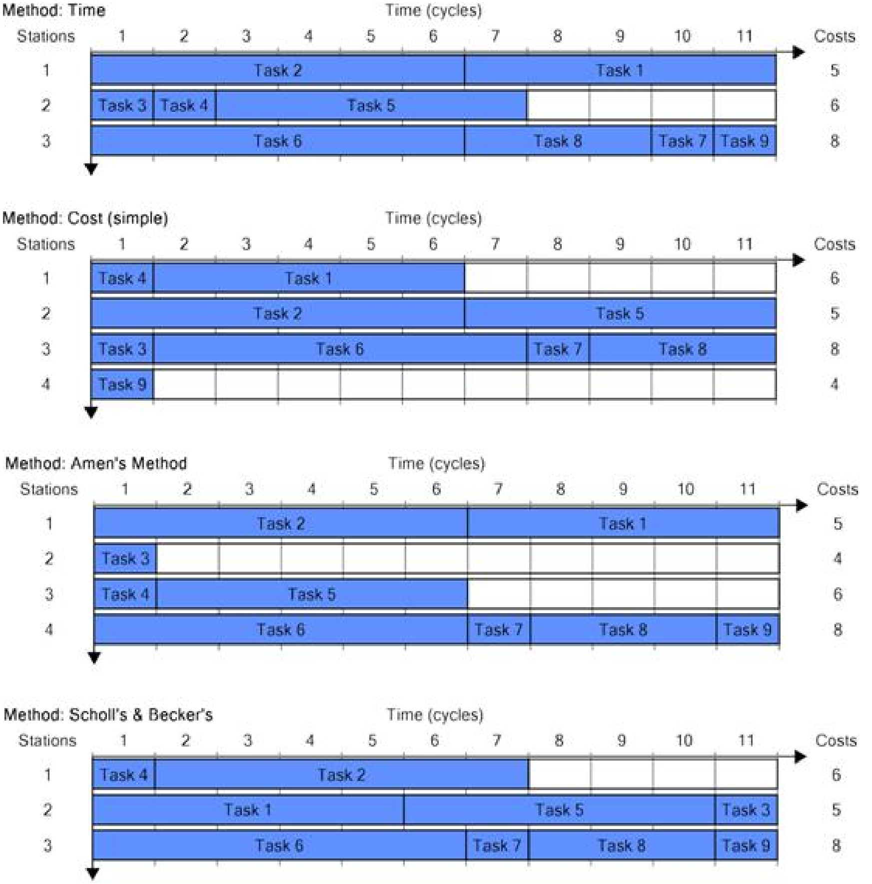

Fig. 3. *Gant chart's for c=11*

Table 1. *Task assigning for different algorithms for c=11*

$c = 11$	Algorithm							
Station	Time		Cost (simple)		Amen's		Scholl's & Becker's	
m	I^s_m	k^s_m	I^s_m	k^s_m	I^s_m	k^s_m	I^s_m	k^s_m
1	{2, 1}	5	{4, 1}	6	{2, 1}	5	{4, 2}	6
2	{3, 4, 5}	6	{2, 5}	5	{3}	4	{1, 5, 3}	5
3	{6, 8, 7, 9}	8	{3, 6, 7, 8}	8	{4, 5}	6	{6, 7, 8, 9}	8
4	–	–	{9}	4	{6, 7, 8, 9}	8	–	–

Table 1. *(continued)*

Sum of costs	19	23	23	19
M	3	4	4	3
LC	209	253	253	209
LE	88%	66%	66%	88%
LT	33	34	44	33
SI	4	11.18	11.18	4

Results for c=10

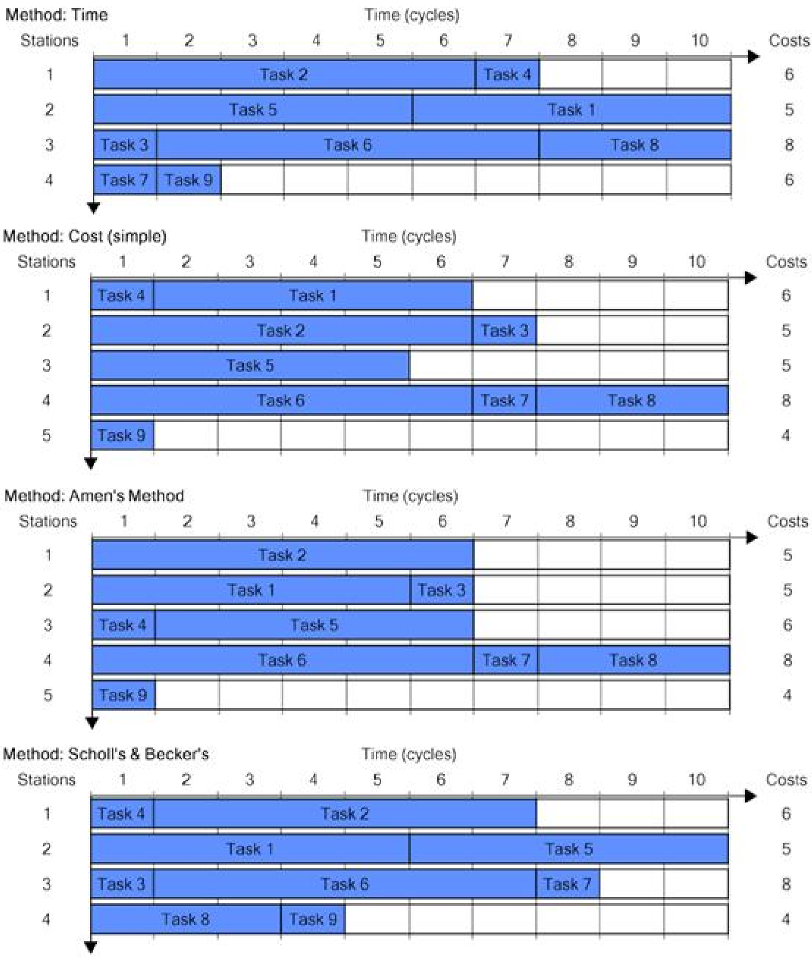

Fig. 4. *Gant chart's for c=10*

Table 2. *Task assigning for different algorithms for c=10*

c = 10	Algorithm							
Station	Time		Cost (simple)		Amen's		Scholl's & Becker's	
m	I^s_m	k^s_m	I^s_m	k^s_m	I^s_m	k^s_m	I^s_m	k^s_m
1	$\{2, 4\}$	6	$\{4, 1\}$	6	$\{2\}$	5	$\{4, 2\}$	6
2	$\{5, 1\}$	5	$\{2, 3\}$	5	$\{1, 3\}$	5	$\{1, 5\}$	5
3	$\{3, 6, 8\}$	8	$\{5\}$	5	$\{4, 5\}$	6	$\{3, 6, 7\}$	5
4	$\{7, 9\}$	6	$\{6, 7, 8\}$	8	$\{6, 7, 8\}$	8	$\{8, 9\}$	8
5	–	–	$\{9\}$	4	$\{9\}$	4	–	–
Sum of costs	25		28		28		24	
M	4		5		5		4	
LC	250		280		280		230	
LE	73%		58%		58%		73%	
LT	32		41		41		34	
SI	8.54		11.44		11.35		7	

6. Conclusions

Time oriented assembly line balancing problem was first described. But in 90's last century some researches started to analyze cost oriented problem. Assumptions of general assembly line balancing problem were very common and very often real companies need more detailed solutions. To achieve better results in final production factories focus not only on best time oriented balance but they focus on costs of equipment and skills of workers. Therefore new algorithms are implemented to search for cost oriented balance. In the paper 3 algorithms of cost oriented and 1 algorithm of time oriented balancing are compared. As we can see for c=11 and c=10 of discussed numerical example we get different solutions but not always special built algorithms are the best. It means that in cost oriented assembly line balancing problem we can also use old time oriented algorithms and then choose the appropriate solution.

Acknowledgement

The research was supported by Grant BK214\Rau1\2011t.5

References

[1] Marecki, F., *Modele matematyczne i algorytmy alokacji i zbiorów na linii montażowej*, ZN. Pol. Śl. Seria Automatyka 82, pp. 31–33, Gliwice (1986)

[2] Rosenberg, O., Ziegler, H., *A comparison of heuristic algorithms for cost-oriented assembly line balancing*, Zeitschrift fur Operations Research 36, pp. 477–495, (1992)

[3] Baybars, I., *A survey of exact algorithms for the simple assembly line balancing problem*, Management Science 32, pp. 909–932, (1986)

[4] Amen, M., *An exact method for cost-oriented assembly line balancing*, International Journal of Production Economics 64, pp. 187–195, (2000)

[5] Amen, M., *Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time*, International Journal of Production Economics 69, pp. 255–264, (2001)

[6] Amen, M., *Heuristic methods for cost-oriented assembly line balancing: A survey*, International Journal of Production Economics 68, pp. 1–14, (2000)

[7] Grzechca W., *Dualny problem balansowania linii montażowe*, in Komputerowo zintegrowane zarządzanie Tom I, pp. 532–539. Opole (2010)

[8] Scholl, A., Becker, Ch., *A note on an exact method for cost-oriented assembly line balancing*, Universitat Bern, Institut fur Unternehmensrechnung und Controlling, Bern, Switzerland, (2000)