A Review Paper on Algorithms Used for Simple Assembly Line Balancing Problems in the Automotive Industry

Salah Eddine Ayoub El Ahmadi,
Laila El Abbadi,
Moulay Taib Belghiti
National School of Applied Sciences
Ibn Tofail University
Kenitra, MOROCCO
salaheddineayoub.elahmadi@uit.ac.ma
laila.elabbadi@uit.ac.ma
Moulaytaib.belghiti@uit.ac.ma

Abstract

An assembly line is a technique used in mass production industries especially in the automotive industry, it consists of many work stations organized along a belt transport system or another handling equipment. The Assembly line balancing problem (ALBP) is a crucial question because it affects directly the productivity of the whole manufacturing system. In this paper we give an up to date review about this topic and we discuss the development of the classification of the assembly line balancing problems (ALBP) and the procedures and algorithms that propose solutions to the ALBP in our case.

K**eywords**
Assembly line, Assembly line balancing problem, Productivity**.**

1. Introduction

The objective of the assembly line balancing problem (ALBP) is to assign multiple tasks to an ordered sequence of workstations, such as the precedence relations are satisfied, and some measurements of effectiveness are optimized. (E.g. to minimize the balance delay or minimize the number of work stations or the cycle time; etc.) in order to increase the system productivity which is the ratio of output over input and it depends on several factors such as workers skills, job methods and machines used. (Nuchsara and Nalin 2007).

Salveson was the first who published a paper about the assembly line balancing problems (ALBPs) in 1955, he suggested a linear programming solution to this type of problems. Since then, the topic of line balancing has been a great field of research in the industry field. However, since the ALB problem falls into the non-deterministic polynomial-time hard (NP-hard) class of combinatorial optimization problems, it has consistently developed the efficient algorithms for obtaining optimal solutions. Many research efforts have been directed towards the development of algorithms or heuristics that propose solutions to the ALBPs (e.g. Kilbridge and Wester, 1961; Helgeson and Birnie, 1961; Hoffman, 1963; Arcus, 1966; Baybars, 1986) and exact methods to solve the ALB problems. (E.g. Jackson, 1956; Bowman, 1960; Van Assche and Herroelen, 1978; Mamoud, 1989; Hackman et al., 1989; Sarin et al., 1999). Recently two articles by Scholl and Becker (2006); Becker and Scholl (2006) provide the review of the exact and heuristic solution procedures for simple assembly line balancing (SALB). In this paper, we give a review about the development of the classification of the ALBPs and a comparison of the procedures used in the balancing of assembly lines in the automotive industry.

2. The Assembly Line

2.1 Definition

An assembly line consists of a set of work stations where some specific tasks are carried to produce a product. Work stations are linked together by a transportation system that moves the work in process from one work station to the other.

The first work station is fed by a manufacturing system and after the cycle time (the time between the entry of a piece into the system and the entry of the following piece. It means also the time that is available for each station to perform the assigned tasks before the product passes to the next workstation), the product at each station is transferred to the next station, each task has a process time and each work station process tasks assigned to it within this cycle time. At the end of each cycle, a product comes out of the last station. Tasks cannot be assigned to stations arbitrarily. There are some technological constraints, such as tasks that should be processed in a specific order, i.e. tasks that have some precedence relations (Bulut 2012).

In general, balancing an assembly line means to allocate the elementary operations to be carried out to different stations, respecting several constraints and according to specific objectives.
For this purpose, the total amount of work necessary to assemble a work piece (job) is split up into a set V = {1,…., n} of elementary operations named tasks. Performing a task j takes a task time t_j and requires certain equipment and/or skills of workers. The total workload necessary for assembling a work piece is measured by the sum of task times ∑t. These elements can be summarized by a precedence diagram as shown in Figure 1. It contains a node for each task and node weights for the task times (Nuchsara and Nalin 2007).

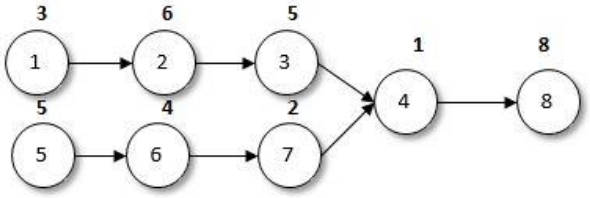

Figure 1. Precedence diagram

2.2 Types of Assembly Lines

2.2.1 Single-Model Assembly Line

Single-Model Assembly Line is a production system that consists of several workstations aligned along a conveyor belt. Those workstations are interconnected between them through the conveyor belt whose speed depends on the total time of the tasks of the bottleneck station (known also as cycle time) to produce only one model product (Sandeep and Sunil 2014).

2.2.2 Multi Model Assembly Line

In this type of lines, several products are assembled in batches. The batch production line is used in the case of multiple different products, or family of products, which present significant differences in the production processes. Using batch production leads to scheduling and lot-sizing problems.

2.2.3 Mixed Model Assembly Line

This type of lines includes different models of the same base product, which have identical production process and assembled simultaneously in the same line. A typical example is a family of cars with different options: some of them will have a sunroof, others will have ABS, etc. In this type of line, the same resources are needed to assemble all the products (Brahim and Alain 2006).
We present in figure 2 the 3 types of assembly lines:

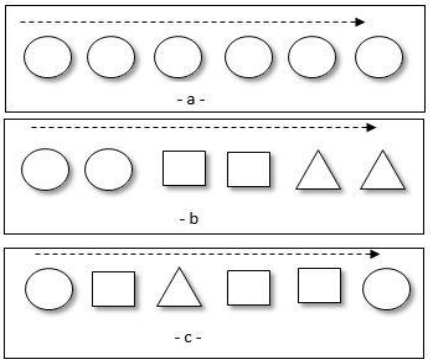

Figure 2. Types of assembly lines

2.3 Assembly line balancing in the automotive industry

In the automobile industry an assembly line starts with a bare chassis. Then Components are attached successively as the growing assemblage moves along a conveyor. Parts are matched into subassemblies on feeder lines that intersect the main line to deliver exterior and interior parts, engines, and other assemblies. As the units move by, each worker along the line performs a specific task, and every part and tool is delivered to its point of use in synchronization with the line.

The assembly plant in the automotive industry is generally composed of 2 workshops, the mechanical workshop where the workers assemble the basic mechanical parts of the vehicle (the motor, the brakes, the battery, the radiator, the fuel tank, the gear box…) and the saddlery workshop where they assemble the interior pieces (the seats, the dashboard, the seatbelt, the steering wheel, the lights, …). Each workshop is composed of elementary units, and those units are the assembly lines that we must balance in order to get higher productivity and less workers.

3. Classification of Assembly line balancing problems

There are different kinds of problems in the assembly line balancing. One of the possible classifications is the one proposed by (Baybars 1986), in which he divides the balancing problems to two major ones: the simple problems (SALBP) and the general problems (GALP).
SALBP: it refers to Simple Assembly Line Balancing Problem, it's the simple version of balancing problems, where the objective is to minimize the cycle time for a fixed number of workstations and vice versa.
GALBP: it refers to General Assembly Line Balancing, it includes the problems that are not included in SALBP. Those are: mixed model line balancing problem, U-shaped assembly line problems, robotic assembly line balancing problem and multi-objective assembly line problems.

Our focus in this paper will be more on the Simple ALBPs because they are common in the industry world and especially in the automotive companies, and the assumptions given by (Baybars 1986) for this type of problems can be listed as follows:

(S-1) Mass-production of one homogeneous product.
(S-2) All tasks are processed in a predetermined mode (no processing alternatives exist).
(S-3) Paced line with a fixed common cycle time according to a desired output quantity.
(S-4) The line is considered to be serial with no feeder lines or parallel elements.
(S-5) The processing sequence of tasks is subject to precedence restrictions.
(S-6) Deterministic (and integral) task times t_j.
(S-7) No assignment restrictions of tasks besides precedence constraints.
(S-8) A task cannot be split among two or more stations.
(S-9) All stations are equally equipped with respect to machines and workers.

The Simple Assembly Line Balancing Problems are basically classified into SALBP 1 and SALBP 2.
The SALBP 1 is the assembly line balancing problem in which the objective is to group the tasks into a minimum number of workstations for a given cycle time, which in turn maximizes the balancing efficiency of the assembly line.
The ALBP 2 is the type 2 of the assembly line balancing problem, in which the tasks are grouped into a given number of workstations such that the cycle time is minimized.
In the case that we are studying, the cycle time is given (130 seconds) and the objective is to group the tasks into a minimum number of workstations, so the type of assembly line balancing problems studied in this article are the Simple Assembly Line Balancing Problems SALBP1.

The simple assembly line balancing problem type 1 is a basic problem of assembly line balancing in the industry in general. The tasks i= 1….m are defined by the task times t_j and their positions in the precedence graph (Tom 2014).
The goal is to minimize m as number of the charged stations given a fixed cycle time c.
For SALBP-1 a solution is feasible if:

i. The tasks of each station do not have a task time sum larger than c.
ii. No predecessor of any task j is assigned to a later station than ^{j} is assigned to.

SALBP-1 is a NP-hard problem, it means that this type of problems is belongs to the non-deterministic polynomial-time hard class of combinatorial optimization problems so that heuristics are essential to obtain upper bounds for the problems. In addition to that we need lower bounds methods to assess the quality of found solutions.

The notations considered in the SALBP1 mathematical model are clarified as follows:

s workstation, s = 1, 2, …., n
m machine, m = 1, 2, …., r
w worker, w = 1, 2, …., h
k task, k = 1,2, …, j
C cycle time
As number of workstations
Bm number of machines
X_{ks} = 1 if task k is assigned to workstation s
0 otherwise
t_k time required by task k
E_a earliest task in precedence relation
L_a latest task in precedence relation
x_{as} = 1 if earliest task a is assigned to workstation s
0 otherwise

The mathematical equation of SALBP-1 with resource constraints is presented in below equations.

$$f_1 = \sum_{k=1}^{j} tk \qquad (1)$$

The first objective function (1) is to minimize the number of workstations for a given cycle time

$$\prod_{S=Ek}^{Lk} X_{ks} = 1 \qquad (2)$$

Constraint (2) is an assignment constraint, which ensures that each task is assigned only once.

$$\sum_{k \in F_s} t_k \, x_{ks} \leq C \qquad (3)$$

Constraint (3) is a cycle time constraint, which ensure that the total times in each workstation does not exceed the given cycle time.

$$\sum_{s=E_a}^{L_a} s x_{as} \leq \sum_{s=E_b}^{L_b} s x_{bs} \quad \text{for } \forall\, (a,b) \in P \qquad (4)$$

Constraint (4) is a precedence relation constraint, which guarantees that the precedence relation among tasks is not violated.

$$\sum_{k \in F_s} x_{ks} \leq \|F_s\| A_s \qquad (5)$$

Constraint (5) is a workstation constraint, which guarantees that a workstation is utilized if the task(s) is/are assigned to it (Kamarudin 2017).
There are many exact and heuristic procedures, branch and bound procedures and dynamic programming procedures that propose solutions for SALBP1, we will study in this paper the branch and bound procedures because those are applicable in the assembly lines in the automobile industry.

4. Comparison between branch and bound procedures

We will compare the 4 basic branch and bound procedures:

- Johnson's (1988) FABLE
- Nourie and Venta's (1991) OptPack

- Hoffmann's (1992) Eureka
- Scholl and Klein's (1997) SALOME

To make an overall comparison we will start by comparing the first step which is the branching step, and then bounding and finally the logical tests done by those four procedures.

4.1 Branching (Enumeration)

In case of FABLE and OptPack a single task is assigned to the earliest station not being filled maximally in each branching step (it means the task that have available time) (task-oriented branching).
Each subproblem created is immediately branched again (laser search). Since the tasks are assigned to the earliest station available, the stations are filled in forward direction from the first to the last station, so the enumeration algorithm does not repeat or enumerate partial solutions more than once. While FABLE does not apply any heuristic prior to constructing the enumeration tree, OptPack starts with an initial solution found by the Hoffmann heuristic. (Armin and Robert 1999)

Since FABLE and OptPack perform a laser search, they find a feasible solution in the first complete branch of the tree without waiting the other branches to be analyzed, the quality of which depends on the task renumbering used.

Eureka applies the lower bound method. At first, a laser search procedure, which branches in a station-oriented manner by assigning a complete load to the earliest unloaded station at a time, is applied to find a solution with at most the minimal number of stations.

4.2 Bounding

Eureka and OptPack apply only the simple bound which is implicitly contained in their enumeration processes, but FABLE and SALOME use more bound arguments which are based on correspondences of SALBP-1 with other combinatorial optimization problems such as bin packing, one- and parallel-machine sequencing, vehicle routing. For example, the number of tasks whose operation times exceed half the cycle time (plus half the number of tasks with operation time c/2) is a lower bound on the number of stations, because each of these tasks requires a station of its own. This simple bin packing bound also applies to many other capacity constrained problems. (Armin and Robert 1999)

4.3 Logical tests

The most important component of FABLE is its strong use of dominance rules and reduction procedures. The most intuitive dominance rule is the maximum load rule which allows fathoming any sub problem whose partial solution contains a non-maximal load.
These rules are also applied by SALOME which additionally utilizes information gained from computing bounds for reducing sub problems by assigning (prefixing) tasks to stations.
Eureka does not apply any logical test. Even non-maximal station loads are allowed. (Armin and Robert 1999)

The measures lead to different rankings of the procedures. The performance of the four procedures FABLE, OptPack, Eureka and SALOME are examined within a computational experiment using many data sets, the comparison concern the average relative deviations, the maximal deviations, the number of proven optimal solutions and the computation time. Table 1 presents the rankings of the studied procedures.

A computational experiment of the four algorithms (procedures) is done in an automotive company, to compare between those algorithms and determine the advantages and disadvantages of each one, the tests are performed in a personal computer with a 8 GB of RAM and the procedures are applied to data set collected from a simple assembly line with 126 manual workstations, and the given cycle time is 130 seconds.
We compare the average relative deviations, the computation time, the idle time, and the number of optimal solutions for each procedure.

Table 1 presents the rankings of the studied procedures.

Table 1. Rankings of the procedures studied

	SALOME	Eureka	FABLE	OptPack
The average relative deviations	1	3	4	2
The number of proven optimal solutions	1	3	2	4
Idle time	1	4	2	3
The computation times	1	4	2	3

As a result of this comparison, SALOME outperforms the other algorithms with respect to all of the measures, Eureka shows less precision, more idle time in the workstations and takes more time in computation than the other algorithms, OptPack is performing in the average relative deviations but it fails in proving optimal solutions in the majority of the instances.

Though SALOME contains no initial heuristic, its flexible enumeration scheme aiming at finding very good solutions as early as possible leads to good and often optimal solutions very quickly. The dynamic rule controlling the bidirectional selection of stations is responsible for constructing rather small enumeration trees allowing for proving optimal solutions.

Conclusion

This paper tries to compare between the branch and bound procedures (SALOME, Eureka, FABLE, OptPack) and concerning all the measures, SALOME clearly outperforms the other procedures. But it needs improvements in reducing the size of the enumeration tree, which is reduced in the OptPack Algorithm by using the tree dominance rule.
The limitation of this research is related to the constraints of workstations, so this research cannot be generalized to the automobile industry in all the cases. So, in the future, this research can be done with the optimization and minimization of those constraints.

References

Nuchsara Kriengkorakot and Nalin Pianthong, The Assembly Line Balancing Problem: Review articles, *KKU Engineering Journal Vol.* 34 No .2 (133 - 140) March – April 2007

Aykut Bulut, Simple Assembly Line Balancing Problem (SALBP) Type 2, *in press,* Mai 2012

Sandeep Choudhary, Sunil Agrawa, Single Model Assembly Line Balancing for Newly Recruited Employees, *5 th International & 26th All India Manufacturing Technology, Design and Research Conference (AIMTDR 2014)* December 12th–14th, 2014

Brahim Rekiek, Alain Delchambre, Assembly Line Design: The Balancing of Mixed-Model Hybrid Assembly Lines with genetic algorithms, *Springer Edition,* 2006

I. Baybars. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science, 32(8):909–932,* August 1986

Tom Pape, Heuristics and lower bounds for the simple assembly line balancing problem type 1: Overview, computational tests and improvements, *European Journal of Operational Research,* 2014

N.H. Kamarudin, M.F.F. Ab. Rashid, Modelling of Simple Assembly Line Balancing Problem Type 1 (SALBP-1) with Machine and Worker Constraints, *International PostGraduate Conference on Applied Science & Physics,* 2017

Armin Scholl, Robert Klein, Balancing assembly lines effectively - A computational comparison, *European Journal of Operational Research 114,* (1999)

Biographies

Salah Eddine Ayoub El Ahmadi, is a 1st year PhD student at the Electric, engineering and computing and mathematical science Laboratory attached to the National School of Applied Sciences of Kenitra (ENSA-Kenitra), University Ibn Tofail, Morocco and he earned his bachelor and master's degree of industrial engineering from the same university. He has worked in the industry (automotive sector specifically).

His research interests majorly include performance and operations management and focuses on Assembly line Balancing Problems and Algorithms for solving those problems, Lean Manufacturing and Logistics.

Laila EL ABBADI, PhD. is a professor of industrial engineering at the National School of Applied Sciences of Kenitra (ENSA-Kenitra), University Ibn Tofail, Morocco and a member of the Systems Engineering Laboratory attached to the same University. Her research focuses on Lean Manufacturing, production systems, quality management, Industry 4.0 and quality assurance in higher education.

Moulay Taib Belghiti, PhD. is a professor of mathematics and computing at the National School of Applied Sciences of Kenitra (ENSA-Kenitra), University Ibn Tofail, Morocco and head of the engineering and computing and mathematical science Laboratory. His research focuses on operational research, optimization and linear programming.