DOKUZ EYLUL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

# TWO-SIDED ASSEMBLY LINE BALANCING USING TEACHING-LEARNING BASED OPTIMIZATION ALGORITHM AND GROUP ASSIGNMENT PROCEDURE

by
**Dilek AYDIN**

January, 2013
İZMİR

# TWO-SIDED ASSEMBLY LINE BALANCING USING TEACHING-LEARNING BASED OPTIMIZATION ALGORITHM AND GROUP ASSIGNMENT PROCEDURE

**A Thesis Submitted to the**
**Graduate School of Natural and Applied Sciences of Dokuz Eylül University**
**In Partial Fulfillment of the Requirements for the Degree of Master of Science in**
**Industrial Engineering, Industrial Engineering Program**

**by**
**Dilek AYDIN**

**January, 2013**
**İZMİR**

## M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"TWO-SIDED ASSEMBLY LINE BALANCING USING TEACHING-LEARNING BASED OPTIMIZATION ALGORITHM AND GROUP ASSIGNMENT PROCEDURE"** completed by **DİLEK AYDIN** under supervision of **ASST.PROF.DR. GONCA TUNÇEL MEMİŞ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

..............................................

Asst. Prof. Dr. Gonca TUNÇEL MEMİŞ

Supervisor

| | |
|---|---|
| .............................................. | .............................................. |
| Assoc.Prof.Dr. Hasan SELİM | Assoc.Prof.Dr. Hasan ÖZTÜRK |
| (Jury Member) | (Jury Member) |

Prof.Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

# ACKNOWLEDGMENTS

First, I would like to thank my supervisor Asst. Prof.Dr. Gonca TUNCEL MEMIS for her guidance, support and advice on every stage of my thesis. In addition, I would like to thank to Alper HAMZADAYI for his help and interest he have shown to my study.

I would also wish to express my thanks to Emre HALICI, Tuğçe ŞENGÜN and Erhan ÖZCAN for their supports and helps throughout application of this study.

Last, I would like to emphasize my thankfullnes to my parents for their love, confidence, encouragement and support in my whole life.

Dilek AYDIN

# TWO-SIDED ASSEMBLY LINE BALANCING USING TEACHING-LEARNING BASED OPTIMIZATION ALGORITHM AND GROUP ASSIGNMENT PROCEDURE

## ABSTRACT

Assembly line balancing plays a crucial role in modern manufacturing companies in terms of the growth in productivity and reduction in costs. The problem of assigning tasks to consecutive stations in such a way that one or more objectives are optimized subject to the required tasks, processing times and some specific constraints is called the Assembly Line Balancing Problem (ALBP). Depending on production tactics and distinguishing working conditions in practice, assembly line systems show a large diversity. Although, a growing number of researchers addressed ALBP over the past fifty years, real-world assembly systems which require practical extensions to be considered simultaneously have not been adequately handled. This thesis deals with an industrial assembly system belonging to the class of two-sided line with several additional assignment restrictions which are often encountered in practice. First, we solved the two-sided ALBP by using a heuristic approach named Group Assignment Procedure. Then, we used Teaching-Learning Based Optimization (TLBO) Algorithm which is recently developed for the optimization of mechanical design problems, and then applied to various engineering problems. Computational results are compared in terms of the line efficiency, and the solution structure with workload assigned to the stations was presented.

**Keywords:** Assembly line balancing, two-sided assembly lines, teaching-learning based optimization, group assignment procedure

# İKİ TARAFLI MONTAJ HATTI DENGELEME PROBLEMİNDE ÖĞRETME-ÖĞRENME TABANLI OPTİMİZASYON ALGORİTMASI VE GRUP ATAMA YÖNTEMİ KULLANIMI

## ÖZ

Montaj hattı dengeleme modern üretim sistemlerinde verimlilik artışı ve maliyetlerin azaltılması açısından son derece önemli bir rol oynamaktadır. Gerekli işler, operasyon zamanları ve belirli atama kısıtları dikkate alınarak, bir ya da daha fazla amacı optimize edecek şekilde işlerin ardışık istasyonlara atanması problemi Montaj Hattı Dengeleme Problemi (MHDP) olarak adlandırılır. Gerçek hayattaki çalışma koşulları ve uygulanan üretim yöntemlerine bağlı olarak montaj hattı sistemleri geniş ölçüde çeşitlilik göstermektedir. Son elli yılı aşkın bir süredir artan sayıda araştırmacı MHDP'leri üzerinde çalışmasına rağmen, uygulamaya yönelik ek kısıtlar içeren gerçek hayat montaj sistemleri literatürde yeteri kadar ele alınmamıştır. Bu tezde, endüstriyel sistemlerde sıklıkla karşılaşılan bir takım ek atama kısıtları içeren iki-taraflı bir montaj hattı dengeleme problemi üzerinde çalışılmıştır. İlk olarak, sezgisel bir yaklaşım olan Grup Atama Prosedürü kullanılarak problem çözülmüştür. Daha sonra, son zamanlarda mekanik tasarım problemleri için geliştirilen ve çeşitli mühendislik problemlerine uygulanmış olan Öğretme-Öğrenme Tabanlı Optimizasyon (TLBO) Algoritması kullanılmıştır. Uygulama sonuçları hat etkinliği açısından karşılaştırılmış ve elde edilen çözüm yapısı istasyonlara atanan iş yükleri ile birlikte sunulmuştur.

**Anahtar sözcükler:** Montaj hattı dengeleme, iki taraflı montaj hatları, öğretme-öğrenme tabanlı optimizasyon algoritması, grup atama prosedürü.

## CONTENTS

# CHAPTER ONE
# INTRODUCTION

Nowadays, the majority of production processes in our country and all over the world are carried through assembly operations. Therefore, assembly lines form the basis of the manufacturing systems where production is performed in a flow-line production system; it is called as a “mass production”. In these lines, raw materials or semi-finished goods enter from one point and they pass a number of operations, then they leave from manufacturing process as finished products. First, in 1913, Henry Ford started out with the idea of mass production and he designed an assembly line to manufacture the automobiles. Since then, Assembly Line (AL) concept has been pervaded, as it has widely proven its effectiveness to produce well-qualified, low-cost standardized similar products.

A classic assembly line is composed of serial stages, in which workpieces (jobs) are flowed down the line and transferred from one workstation to the other through workforce or material handling equipment. At each stage, definite assembly operations are completed repeatedly in order to obtain finished products. The tasks are allocated to workstations considering some restrictions including precedence constraints, number of workstations, cycle time and incompatibility relations between tasks. The problem of assigning jobs to consecutive workstations that one or more goals are optimized based on the required tasks, processing times and some particular constraints are named the Assembly Line Balancing Problem (ALBP).

The process of balancing is a crucial task in designing highly efficient and cost effective assembly lines. The establishment or re-arrangement of a line is quite an expensive investment so effective regulations of lines are essential at the beginning of process. Lines need to be balanced in the design stage; otherwise unbalanced lines cause inefficiency in production, increased cost, and a lot of casualties such as waste of labor or equipment.

Since the classical ALB problem was first described in 1955 by Salveson, many studies have been done with regard to assembly line design problems. Researchers have focused on improving qualified and fast solution approaches for solving the line balancing problem in assembly systems. In the first researches, the authors studied on mostly minimizing number of workstations and used mathematical modeling methods, e.g. integer programming and goal programming. Then, they head towards heuristic approaches to handle large size problems.

Based on the restrictions on operation directions, assembly lines can be classified as one-sided assembly lines and two-sided assembly lines. Two-sided assembly lines are usually designed to produce high-volume large-sized standardized products, such as automobiles, trucks, buses and home appliances, in which some tasks must be performed at a specific side (left-side or right-side) of the product. Although a large number of methods for solving one-sided assembly line balancing problem have been studied in literature, little attention has been paid to balancing of two-sided assembly lines (Simario & Vilarinho, 2009). The literature review shows that over the past ten years the researchers started to study on two-sided assembly lines that are recognized to be of crucial importance in real life. However, problems considered in these studies were generally test problems from the literature (e.g., P9, P12, P24, P65, and P148 (Bartholdi, 1993; Kim et al., 2000; Lee et al, 2001)). Real-world assembly systems which require practical extensions to be considered simultaneously have not been adequately handled by the authors. This thesis deals with an industrial assembly system belonging to the class of two-sided line with several additional assignment restrictions which are often encountered in practice. First, we solved the problem by using a heuristic approach named Group Assignment Procedure where assignments were carried out based on task groups rather than individual tasks in order to maximize work relatedness and work slackness. Then, we used Teaching-Learning Based Optimization (TLBO) Algorithm which is recently developed for the optimization of mechanical design problems, and then it has been applied to various engineering problems.

The remainder of this thesis consists of four chapters.

Chapter 2 includes an overview of assembly line balancing. We defined main concepts related to assembly lines, classification of problem types and solution approaches to solve line balancing problems.

In Chapter 3, we presented a literature review in detail which includes the analysis of the studies on assembly line balancing problems which spans 17 years from 1995 to 2012.

In Chapter 4, we introduced an industrial assembly system, which can be characterized as a two-sided assembly line. In order to improve the line balance implemented by the company for a given cycle time, the assembly line balancing problem is solved by using two solution approaches; Group Assignment Procedure and Teaching-Learning Based Optimization Algorithm. Computational results are compared in terms of the line efficiency, and the solution structure with workload assigned to the stations was presented.

In Chapter 5, we summarized the research work made by this thesis and discussed concluding remarks for possible future research.

# CHAPTER TWO
# ASSEMBLY LINE BALANCING

## 2.1 Introduction

In this section, the main features and additional characteristics of assembly line systems are provided, the basic concepts relevant to assembly line balancing problem with various problem classification schemes are presented, and then the solution methods to line balancing problems are discussed.

## 2.2 Assembly Lines

An assembly line (AL) is a production process which is composed of different operations. Workpieces are successively combined on a product at each station to manufacture a final product. ALs are the mostly used technique in mass production, as they enable the assembly of complicated products by workers with restricted training and devoted robots and/or machines.

Assembly lines consist of workstations arranged by a conveyor belt or a similar material handling system. The parts are flowed towards end of the line and transferred among the workstations (Scholl, Fliedner & Boysen, 2010). At every station, specific operations are performed continually in connection with cycle time. When tasks are completed at each station, finished product is obtained (see Figure 2.1).

### *2.2.1 Basic Concepts of Assembly Lines*

*Assembly* is a process of combining different parts with the purpose of obtaining finished product.

*Assembly line* is a manufacturing line which consists of a sequence of stations arranged along a conveyor belt. The parts are consequentially flowed to end of the line and are moved throughout the line. All stations have a set of precedence relations and an operational process time.

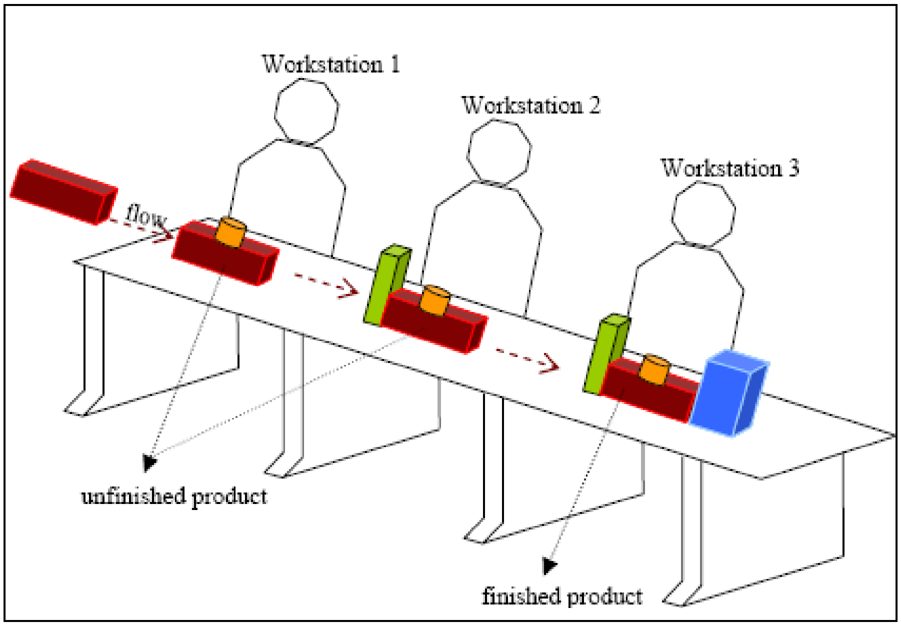


Figure 2.1 Concept of assembly line (Ozmehmet T., 2007)

*Operation/task* is a job, which is the smallest indivisible part of an assembly process on a product.

*Station/Workstation* is a location in which one or more tasks are performed by one or more workers along the assembly line.

*Cycle time* is a time, which represents maximum amount of time the job allowed to spend at each station to reach targeted production rate.

*Workstation/station time* is equivalent to total time of the completion of operations allocated to a workstation.

*Task processing time/task time* is the time needed to start and finish a task on an assembly line.

*Station slack/delay time* is a time, which represents difference among the workstation time and cycle time at any one station.

*Precedence diagram* shows the sequence of tasks as a graphical representation. Some operations have to follow to each other due to the technical specifications of assembly. This diagram is the most important information while sequencing distributing tasks among workstations. As we seen in the figure below, nodes of the graph represent tasks, node weights represent task times and arcs show precedence relations.

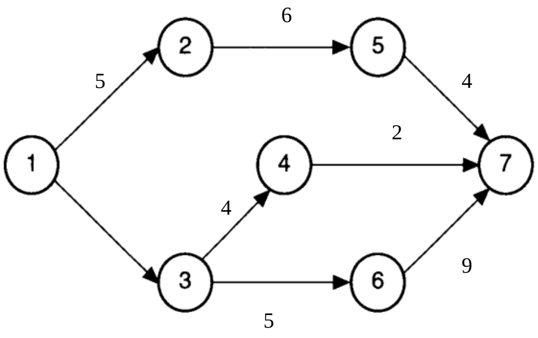


Figure 2.2 Precedence diagram

For example, in Figure 2.2, task 4 can start if and only if task 3 is completed. In other words, task 3 precedes task 4.

### *2.2.2 Classification of Assembly Lines*

Depending on production tactics and different conditions in practice, assembly line systems show a large diversity; therefore they can be classified in various ways. Figure 2.3 illustrates five main classifications of ALs in terms of number of models, line control, frequency, level of automation, and line layout.

**Number of models**

| | | |
|---|---|---|
| Single model | Mixed model | Multi model |

**Line control**

| | | |
|---|---|---|
| Paced | Unpaced asynchronous | Unpaced synchronous |

**Frequency**

| | |
|---|---|
| First-time installation | Reconfiguration |

**Level of automation**

| | |
|---|---|
| Manuel lines | Automated lines |

**Line layout**

| | | |
|---|---|---|
| Serial line | U-shaped line | Feeder line |

**Operation direction**

| | |
|---|---|
| One-sided assembly line | Two-sided assembly line |

Figure 2.3 Investigated classifications of assembly lines

#### *2.2.2.1 Number of Models*

Assembly lines are distinguished in terms of the number and variety of finished products in the line (see Figure 2.4) (Scholl, 1999).

a. *Single model*

When producing high volume of a product, single-model assembly lines are mostly used to carry out a single homogenous product. In addition, if more than one product is produced on the same line, but neither setups nor distinct differences in processing times occur, the assembly system is also called as a single model line, such as case in the production of CDs or drinking cans.

*b. Multi model*

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 presents significant differences in the production processes. Using batch production leads to scheduling and lot-sizing problems.

*c. Mixed model*

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 (Rekiek & Delchambre, 2006).

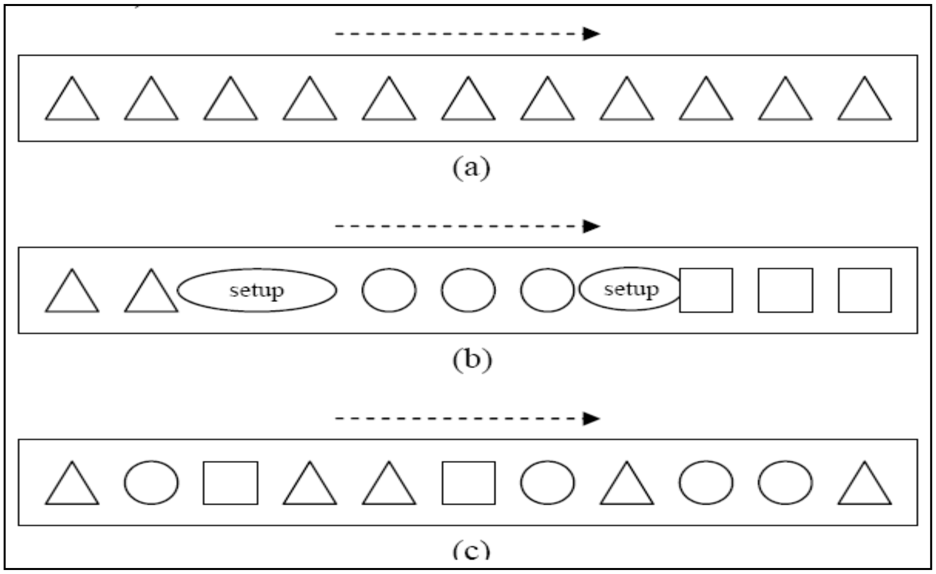


Figure 2.4 Assembly lines for single model (a), multi model (b), mixed model products (c)

#### *2.2.2.2 Line Controls*

Assembly lines can also be distinguished with regard to the line control. Figure 2.5 illustrates the classification of ALs based on the type of line control (Groover, 2001). In this classification scheme, we come up a “velocity” concept. Each line has a velocity that some of them are fixed, others are variable. In variability cases, buffers occur between stations on the line.

##### *a. Paced line*

In paced assembly systems, a common cycle time is given which limits operation times at all stations. The same cycle time is applied to all workstations, so they can begin their tasks at the same time and work-pieces are moved at the same rate.

##### *b. Unpaced line-synchronous case*

The assembly lines, in which workpieces are moved when the required tasks are finished rather than a predetermined time is passed, are called as *unpaced lines*. There are buffer storages along the line. If buffer is full strictly, station is blocked due to the buffer capacity restriction. In the *synchronous systems*, the parts are transferred among the stations as soon as the required operations are completed (Ozmehmet T., 2007).

##### *c. Unpaced line-asynchronous case*

Under *asynchronous case*, a workstation proceeds on its work-piece as soon as it has completed all tasks, and as long as the successor is not prevented by other work-piece. Thus, it can proceed to perform the following work-piece, while the predecessor station keeps delivering the new work-pieces on time (Ozmehmet T., 2007).

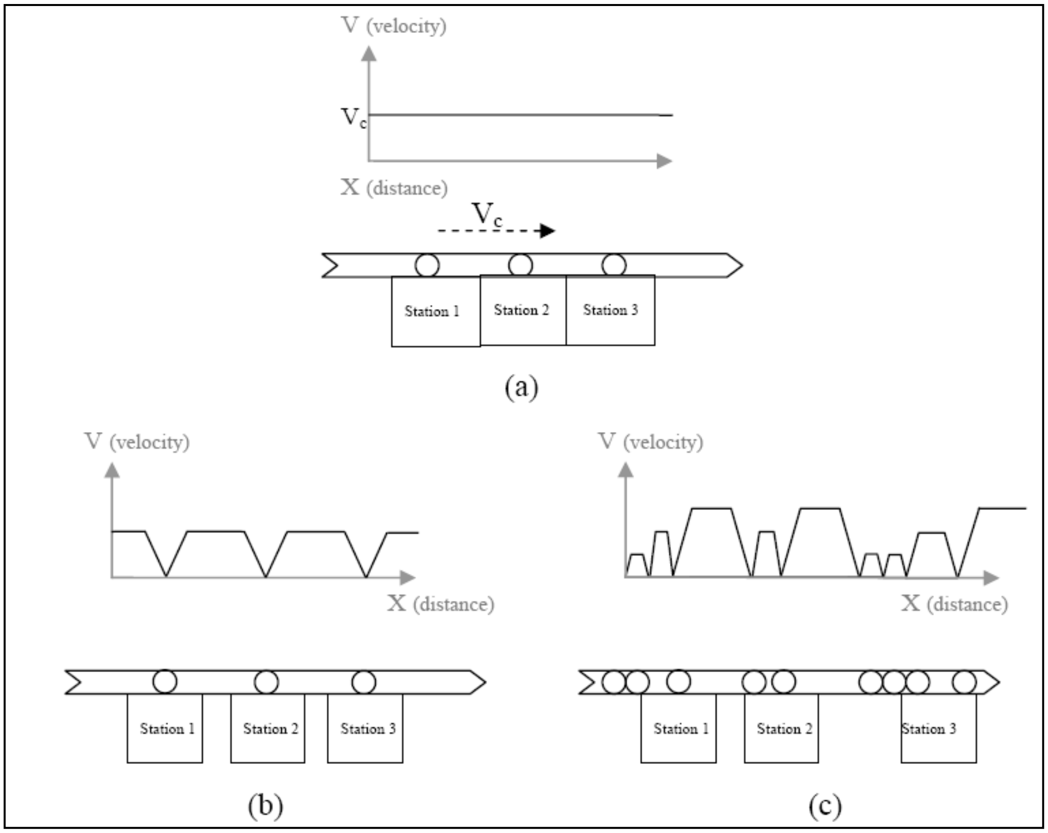


Figure 2.5 Velocity-distance diagrams and physical layout for three types of line control: paced line (a), unpaced line-synchronous case (b), unpaced line-asynchronous case (c) (Groover, 2001).

### *2.2.2.3 Frequency*

#### *a. First-time installation*

When an assembly line system is invested in the first time and parts required to supply have not been bought yet, workstations may be acted as if abstract entities, to which a determined number of operations can be assigned. Alternatives in process can affect on the determining of the precedence graph in various forms. Various machinery or variously skillful workers can perform the same operation at changing effort and costs.

#### b. *Reconfiguration*

A reconfiguration is required when an important revision in the structure of the production program occurs, such as constant shift for the demand on product models. In case those workstations are already existent, as an aim, minimizing the number of workstations is usually less meaningful. Besides, the cycle time is mostly defined with respect to sales forecasts. As an additional aim, it is usually suggested to share the work load as balanced as possible between the stations (Rekiek & Delchambre, 2006).

### *2.2.2.4 Level of Automation*

#### a. *Manuel lines*

Manual lines are mostly used in which the parts produced on the line are fragile or if they need to be hold tightly, as machines/robots often lack the necessary accuracy. Moreover, some countries have low labour costs so manual labor can be a low-cost alternative to expensive automatised machinery. Process times under manual labour depend on stochastic deviations, as the performances of workers are subject to a diversity of factors, e.g., working environment, lack of motivation, the mental/physical stress or pressures.

#### b. *Automated lines*

In the case of automated lines, tasks at the stations are performed automatically. On the other hand, transfers can be performed with two different types of lines: mechanical and nonmechanical. In nonmechanical lines, parts pass from one station to another manually. In mechanical lines, conveyors and related material handling systems are used to transfer the parts between the workstations.

#### *2.2.2.5 Line Layout*

Assembly lines can also be categorized according to the line layout as given in Figure 2.6.

##### *a. Serial line*

The first implementation of assembly line production systems have begun as serial lines. Workstations are arranged consecutively. Assembly operation is started at first station and completed once a product leaves from the end of the line. In a serial line, work flow is easier and faster, but it has a disadvantage of a large area covering (Ozgormus, 2007).

##### *b. U-shaped line*

In *U-shaped lines*, workstations are aligned throughout a quite narrow U, input and output of the line is at same position. Because of its shape, input and output sides are so close together. Workstations in between those sides may operate at two parts of the line facing each other simultaneously. It signifies that a workpiece may revisit the same workstation during the production period without changing the flow way of the line. Thus, balances of workstation loads are usually better than *serial lines* because of the larger number of task-workstation integrations in *U-shaped lines*.

##### *c. Feeder line*

*A Feeder line* consists of a main line with subassemblies. For instance, electronic devices frequently include a number of electronic subassemblies, which has to be combined to obtain a main part.

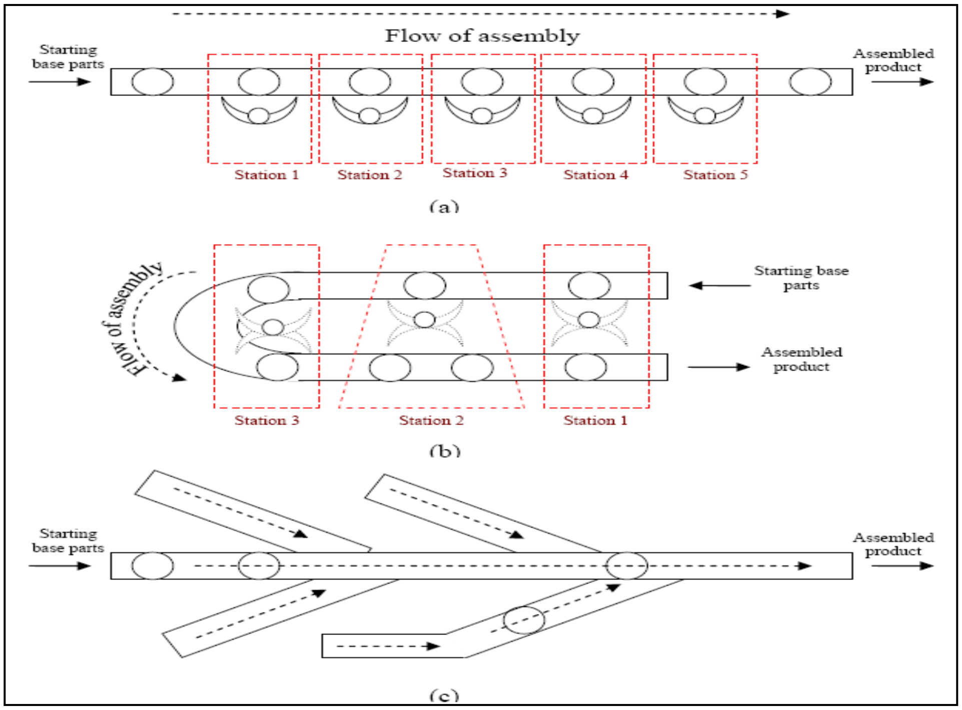


Figure 2.6 Serial lines (a), U-shaped lines (b), feeder lines (c) (Ozmehmet T., 2007)

#### *2.2.2.6 Operation Direction*

*a) One-sided assembly line*

Finally, assembly lines can be categorized based on the restrictions on operation directions. If only one side (left or right side) is used in an assembly line, then it is called as one-sided assembly line. Most of the studies in the literature dealt with balancing of one-sided assembly lines.

*b) Two-sided assembly line*

A two-sided assembly line is a type of production line in which different assembly tasks are performed in parallel at both sides of the line as shown in Figure 2.7. In this situation, some of the assembly operations should be performed at strictly one side of the line (right or left side) and the others can be assigned to either side of the line. Thereby, tasks are classified into three types according to the restrictions on the operation directions: L (left), R (right) and E (either)-type tasks.

Two-sided assembly lines are usually designed to produce large-sized high-volume products such as automobiles, buses, and trucks. These lines have some advantages over one-sided assembly lines: (i) shorter line length (ii) reduced throughput time, worker movements, and setup time (iii) lower cost of tools and fixtures (iv) less material handling (Bartholdi, 1993).

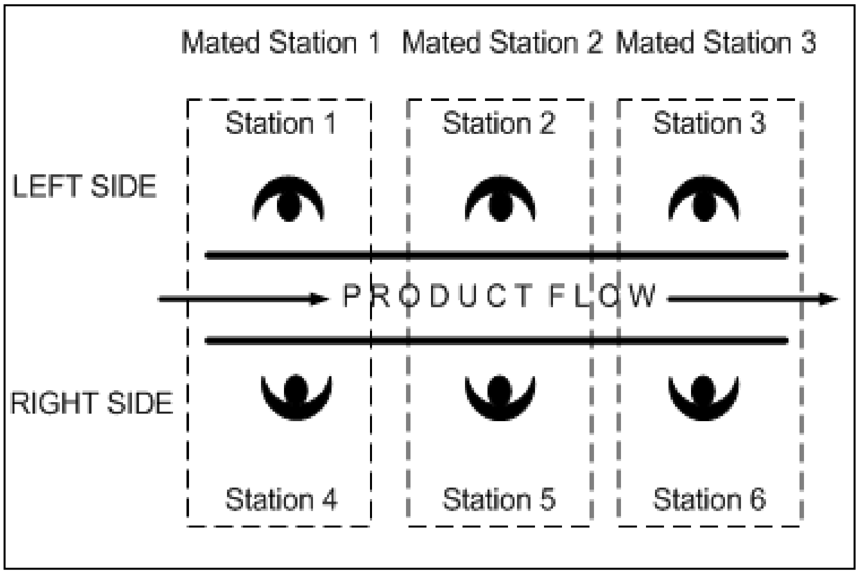


Figure 2.7 Configuration of a two-sided assembly line

## 2.3 Assembly Line Balancing

The establishment of any assembly line is a long dated decision and needs remarkable capital investment. For this reason, an assembly line is tried to design and/or balance as efficiently as possible. In recent years, a lot of researches were dedicated to line balancing in assembly systems.

An ALBP deals with the assignment of the operations between stations so that a given objective function is optimized considering the precedence relations.

### *2.3.1 Classification of Assembly Line Balancing Problems*

ALBPs can be classified based on the problem structure and objective function. Furthermore, each of them is subclassified in themselves.

First group, based on the objective function, includes seven types of line balancing problems. In Type-F problem, there is no any objective function that searches optimum result. The aim is to obtain a feasible line balance for a given cycle time and number of stations. Type-1 and Type-2 have a double relation; for a given cycle time, the first aim tries to minimize the number of workstations, and the second aim tries to minimize the cycle time for a given number of workstations. Type E is the commonly used problem type in which it is aimed to maximize the line efficiency by simultaneously minimizing both the cycle time and the number of stations. Finally, Type-3, 4 and 5 correspond to the objectives of smoothing workload between the workstations, maximization of work relatedness and multiple objectives with Type-3 and Type-4, respectively (Gen, Cheng & Lin, 2008).

The second group is also categorized into two classes according to the problem structure. First class contains Single Model Assembly Line Balancing Problem (SMALBP), Mixed Model Assembly Line Balancing Problem (MMALBP), and Multi Model Assembly Line Balancing Problem (MuMALBP). Second class contains Simple Assembly Line Balancing Problem (SALBP) and General Assembly Line Balancing Problem (GALBP).

The SMALBP involves assembly of just one type of product. The MuMALBP includes more than one product produced in batches. The MMALBP contains assembly line which produce a variety of similar product models simultaneously and continuously, but not in batches. Additionally, SALBP, the simplest version of the ALBP and the special version of SMALBP, contains producing of only one product in the line that is paced line with fixed cycle time, deterministic independent processing times, serial layout, no assignment restrictions, equally equipped (and skilled) workstations, one sided stations and fixed rate launching. In contrast, GALBPs consider further restrictions and problem attributes like incompatibilities between tasks, different line shapes (e.g., two-sided and U-shaped lines), stochastic dependent operation times, space constraints or parallel stations, along with many others. In other words, GALBP is a generalization of SALBP and includes all of the problems that are not categorized as SALBP. Hence, more realistic ALBPs can be formulated and be solved (Ozmehmet T., 2007; Tuncel & Topaloglu, 2013).

Various classifications of ALBPs are shown in Figure 2.8.

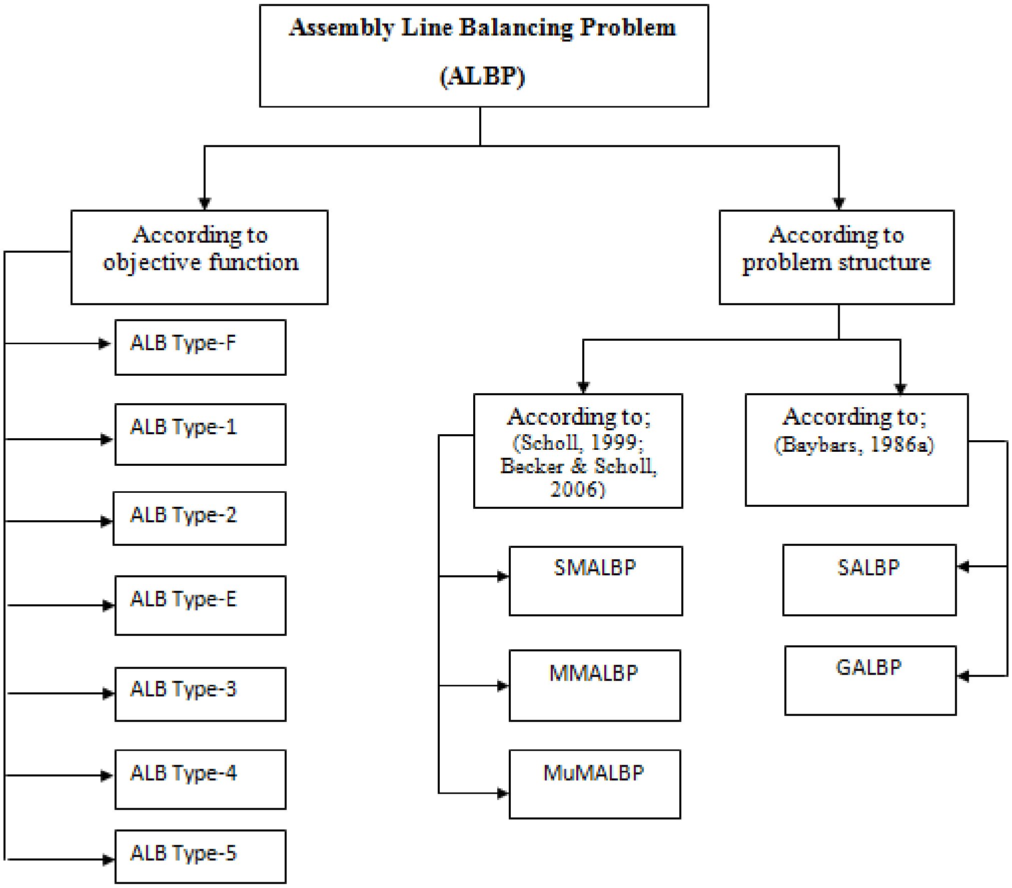


Figure 2.8 Classification of ALBP

Main constraints in ALBP are the cycle time and task precedence constraints. Apart from these constraints, some other constraints may restrict feasible assignments of tasks to stations. These additional constraints are summarized below (Baybars, 1986; Scholl, 1999; Boysen, Fliedner & Scholl, 2008):

*Task zoning constraints:* Some zoning restrictions constrain the assignment of various operations to a specific station which is named *positive zoning constraints* and others forbid the assignment of operations to the same station which is named *negative zoning constraints*. Positive zoning constraints are mostly related with the usage of common equipment or tooling. Hence, some of the operations are needed to assign to the same workstation. Negative zoning constraints are usually related with the technological issues. It may not be possible to perform some tasks in the same workstation because of safety reasons or any other causes.

*Workstation related constraints*: Some operations need particular equipment or material that is only available at a certain workstation so these tasks should be assigned to that workstation.

*Position related constraints*: In producing of the large and heavy workpieces, they have a fixed position and cannot be turned. In this case, we come up position related constraints which are commonly faced in balancing two-sided assembly lines. In that case, tasks are grouped according to the position in which they are performed.

*Operator related constraints:* Some tasks need different levels of skill depending on the operation complexity. Assigning a qualified operator to a determined task is better to combine more monotonous tasks and more variable tasks in the same workstation in order to induce higher levels of job satisfaction and motivation. In addition, stress or pressure in work environment and happiness effect on worker performance significantly.

*Synchronization constraints:* In two-sided lines, sometimes a task can be required to be performed simultaneously with another task by two operators working at opposite side of the line. If a task has synchronization constraint, it has to be assigned to a workstation at the opposite side of the line where its mated-task was started in parallel.

### *2.3.2 Solution Approaches for Assembly Line Balancing Problem*

The solution approaches of ALBPs can be classified into two main groups: exact methods and approximate methods. The exact methods are optimum seeking methods and they constitute the group of enumeration procedures. Approximate methods include heuristic approaches and meta-heuristics. In Figure 2.9, a classification scheme of solution approaches for ALBP is depicted.

*Optimum seeking methods:* In the literature, several approaches for determining lower bounds on the objectives of ALBPs (Type-1 and Type-2) are proposed for solving the problems. The lower bounds are obtained by solving problems which are derived from the considered problem by omitting or relaxing constraints. Most of these techniques fall into two categories; tree search based procedures like branch and bound (B&B) or graph based ones like dynamic programming. A survey on exact methods for ALBP can be found in Scholl (1999).

*Heuristic methods:* Due to the problem size, near optimal or optimal solutions determined by approximation methods are more preferable and acceptable in practice, as they can be applied more efficiently than the other methods. These approaches are divided into two categories; simple heuristics and meta-heuristics.

Heuristics approaches are based on logic and common sense rather than on a mathematical proof. They are composed by constructive or ‘‘greedy” procedures which make use of a static or dynamic priority rule to assign tasks to different workstations (Tuncel & Topaloglu, 2013). None of these methods guarantees an optimal solution, but offer relatively good solutions in much reduced computing times. Most widely used simple heuristics are Ranked Positional Weight Technique (RPWT) (Helgeson and Birnie, 1961), Kilbridge and Wester’s (1961), Moodie and Young's (1965), Hoffman (1963), Immediate Update First-Fit (Hackman, Magazine and Wee, 1989) heuristics. The heuristic procedures for SALBP and GALBP are critically examined and summarized by Ghosh and Gagnon (1989) and Erel and Sarin (1998), respectively.

Meta-heuristics, on the other hand, are improvement procedures which start with an initial solution or population (predefined number of solutions) obtained with a heuristic or randomly generated, then improves it. These methods provide effective approximate solutions for difficult combinatorial optimization problems. In recent years, the usage of meta-heuristics (e.g. Genetic Algorithm, Simulated Annealing, Tabu Search, and Ant Colony Optimization) for solving ALBPs has been received widespread attention among researchers and practitioners.

ALBP takes part in the NP-hard class of combinatorial optimization problems. Therefore, heuristic approaches or simulation based methods which provide reasonable solutions in a shorter time are used more than optimization methods such as linear programming, integer programming, and dynamic programming, which find the best solution to the problem.

Figure 2.9 Classification of solution approaches for ALBP

# CHAPTER THREE
# LITERATURE REVIEW on ASSEMBLY LINE BALANCING

## 3.1 Review of Related Literature

Over the fifty years, many algorithms and heuristic approaches have been proposed to solve wide variety of assembly line balancing problems. The studies which span 17 years from 1995 through 2012 are briefly summarized below and chronologically listed in Table 3.1.

Rubinovitz & Levitin (1995) developed a genetic algorithm (GA) to solve SALBP Type-2. The results were compared with multiple solutions technique (MUST) for balancing single model assembly lines which was suggested by Dar-El and Rubinovitch (1979). The proposed GA performs much faster than MUST for large size problems (i.e. assembly lines with more than 20 workstations) and high flexibility ratio.

Sawik (1995) used an integer programming model for designing and balancing of flexible assembly systems in which different product types were assembled simultaneously. The objective of this study was to assign tasks to stations with limited capacities in order to balance station workloads and station-to-station product movements subject to precedence relations among the tasks.

A multiattribute-based approach was introduced by Kabir & Tabucanon (1995) to determine the number of workstations. A set of appropriate number of stations that were balanced for every model were created. A multiattribute evaluation model was proposed to select the number of stations considering diversity, production rate, minimum distance moved, division of labor, and quality by using the analytic hierarchy process and simulation.

Kim et al. (1996) used a GA for solving several line balancing (ALB) problems with different objectives of Type 1 through Type 5. The authors proposed a new method by improving the classic GA so that it was able to be flexibly fitted to different sorts of aims in the ALB problems.

Klein & Scholl (1996) used a branch and bound algorithm for solving the SALBP Type 2. The problem includes assigning jobs to a determined number of stations in a paced assembly line. In addition, possible precedence restrictions among the operations have to be bear in mind. The authors used a new enumeration technique which was complemented by several bounding and dominance rules. This method was called Local Lower Bound Method.

Ugurdag, Rachamadugu & Papachristou (1997) addressed the problem of assigning jobs between stations so that the cycle time was minimized (Type-2). They provided two step heuristic method, which was based on an integer programming formulation.

Gökçen & Erel (1997) improved a binary goal programming model to solve a MMALBP. This model was based on the concepts proposed by Patterson and Albracht (1975) and the 0-1 goal programming model developed by Deckro and Rangachari (1990) for the SMALBP.

Kim et al. (1998) proposed a new heuristic method based on GA to maximize workload smoothness. The algorithm emphasized utilization of problem-specific information and heuristics to develop the qualification of searching good solutions in the design of representation scheme and genetic operators. The computational results indicated that the developed method outperforms the current heuristics and the compared GA.

Sarker & Pan (1998) addressed a MMALBP by using integer programming. Minimizing the total cost of the availability time and idle time because of various parameters of the line (e.g., launch interval, starting point of task, length of station, upstream walk, locus of the worker’s movement, and task sequences of the mixed models) was the aim of the study. The models were tested on a three-station mixed-model line where the station type is assumed either closed or open.

Gökçen & Erel (1998) introduced a binary integer programming model for the MMALBP. The constraints of the model spitted into four groups: assignment restrictions, precedence relations, cycle time constraint, station constraints with the objective of minimizing the number of stations. The experimental results showed that the model was capable of solving problems with up to 40 tasks in the combined precedence diagram.

Ajenblit & Wainwright (1998) developed a GA solution for the Type-1 U-shaped assembly line balancing problem. One of the main properties of this study was to provide a general frame which can be used to solve the two possible alterations of the problem, minimizing total idle time, balancing of the workload between stations or a combination of both of them. The authors compared the proposed algorithm with the 61 test problems in the literature.  Implementation results showed that the GA acquired the same conclusions as in previous authors in 49 problems, superior conclusions in 11 problems, and just one problem did worse.

Chan et al. (1998) presented how a GA can be applied to solve the line balancing problem in the clothing industry. The numerical results revealed that the efficiency of the GA in handling the considered ALB problem is much better than a greedy algorithm.

Hyun, Kim & Kim (1998) considered three practically important objectives: minimizing total utility work, keeping a constant rate of part usage, and minimizing total setup cost. The sequencing problem with multiple objectives was described and its mathematical formulation was provided. A GA was designed to find near-Pareto or Pareto optimal solutions. A new genetic evaluation and selection mechanism was proposed which was called Pareto stratum-niche cubicle. The results depicted that the suggested GA outperformed the current genetic algorithms, especially for the large size problems which involve great variation in setup cost.

Sarin, Erel & Dar-El (1999) developed a method for solving the single model, stochastic assembly line balancing problem to minimize the total labor cost and the expected incompletion cost occurring from operations which are not finished in the detected cycle time. The procedure was based on determination an initial dynamic programming based solution and its development using a branch and bound algorithm.

Tamura, Long & Ohno (1999) presented a sequencing problem with a bypass subline. The sequencing problem with objectives of leveling the part usage rates and workloads was formulated, and three different algorithms based on goal chasing method, Tabu Search (TS) and dynamic programming were used.

Scholl & Klein (1999) compared the performance of the most effective branch and bound procedures for solving type 1 of the simple assembly line balancing problem (SALBP-1), namely Johnson's (1988) FABLE, Nourie and Venta's (1991) OptPack, Hoffmann's (1992) Eureka and Scholl and Klein's (1997) SALOME for new data sets. Implementation results showed that SALOME is the most powerful procedure.

Gökçen & Erel (1999) presented a shortest route formulation of the MMALB problem. The formulation was based on the shortest-route model developed by Gutjahr & Nemhauser (1964) for the SMALBP. On this basis, the mixed-model system was made into a single-model problem with a combined precedence diagram. Network model was developed in which the nodes (i.e. sets of tasks) of the network were constructed with similar to the Gutjahr and Nemhauser's procedure. Computational results indicated that the proposed model is more efficient than the shortest-route formulation presented in Roberts & Villa (1970).

Bautista et al. (2000) considered noncompatibilities between some set of operations, so if two operations were noncompatible they can't assign to the same station with the objective of minimizing of cycle time for a predetermined number of workstations. The authors proposed a Greedy Randomized Adaptive Search Procedure (GRASP) derived from the applying of several classic heuristics based on the priority rules and a GA.

Frey (2000) considered a paced assembly line with overlapping work zones and a fixed launching rate. A two-step method based on the discrete-event model of the assembly process is presented. Firstly, the given information about the production line and the possible tasks was processed within a branch-and-bound procedure to form a Petri-net model, and then secondly, valid sequences for a new job set were calculated by solving the obtained equations. The algorithm was applied to a sample problem from the automotive industry.

Kim et al. (2000) dealt with two-sided ALB problem including positional constraints. They developed a GA to solve this problem with the objective of minimizing the number of workstations. The authors showed that the proposed GA is flexible to solve various types of optimization criteria and constraints in two-sided ALB problems.

Sabuncuoglu, Eren & Tanyer (2000) suggested a heuristic method which has a structure based on a GA with a special chromosome for solving the deterministic SMALBP. This structure was partitioned dynamically via the evolution process. In addition, elitism was developed in the model by using some concepts of Simulated Annealing. Numerical results with the proposed algorithm denoted that the suggested method outperformed the current heuristics on several test problems.

Ponnambalam, Aravindan, Naidu & Mogileeswar (2000) presented a multi-objective GA method for solving ALBs. The authors used the line efficiency, number of workstations, the smoothness index before trade and transfer and the smoothness index after trade and transfer as the performance criteria. The proposed algorithm was compared with six well-known heuristic algorithms, e.g., Moodie and Young, ranked positional weight, immediate update first fit, Hoffmann precedence matrix, Kilbridge and Wester, and rank and assign heuristic methods. The experimental results showed that the proposed GA outperformed the other heuristics with respect to the performance measures considered. However, the completion time for the GA is longer due to searching for global optimum solutions with more iteration.

Sawik (2000) developed integer programming (IP) formulations and a heuristic solution procedure for a bicriterion loading and assembly plan selection problem in a flexible assembly line. The goal was balancing workloads between workstations and minimizing total transportation time in a unidirectional flow system. For the first goal, the workloads were balanced using a linear relaxation-based heuristic. For the second goal, assembly sequences and routes for all products were chosen using a network flow-based model.

Carnahan, Norman & Redfern (2001) used GA approach for the SALBP Type 2. Three heuristics were improved to obtain good result in the balancing problem which considered both the time and physical demands of the assembly tasks e.g., a combinatorial GA, a ranking heuristic and a problem space GA. In light of the implementation results, the authors concluded that the problem space GA was the most fitted at obtained balances in the others.

Lee et al. (2001) introduced a group assignment procedure for two-sided (left-and right-side) assembly line balancing problem. Minimizing the number of workstations was the aim in this study. For a cycle time, the authors considered positional constraints due to the facility layout, i.e. a task has to be assigned to a prespecified workstation. Group assignment method was used to construct candidate groups and assign tasks according to their rules. The computational results revealed that this method outperformed several heuristics.

Chen, Lu & Yu (2002) used a hybrid GA method for the problems of line balancing with various objectives (e.g., maximizing workload smoothness, minimizing cycle time, minimizing the number of tools and used machines, minimizing the complication of assembly sequences and minimizing the frequency of tool change). They concluded that, the proposed method can efficiently yield a lot of alternative assembly plans to support the design and operation of an assembly system.

Pastor, Andris, Duran & Pirez (2002) addressed a real-life MuMALBP which includes approximately 400 operations, 4 models of the same product, space and tool constraints. The aim of the study was to maximize production rate, obtain an equal cycle time for all models and an equal workload for all stations. To solve this problem, the authors used four different heuristics and two Tabu Search approaches.

Nicosia, Pacciarelli & Pacifici (2002) studied on an ALBP with non-identical workstations, under precedence and cycle time constraints. The objective of this study was minimizing the total cost of the workstations. A hybrid dynamic programming and branch-and-bound algorithm were implemented for solving optimally large instances of assembly line design problem.

Goncalves & De Almedia (2002) proposed a hybrid GA for the SALBP Type-1. The proposed approach combined a heuristic priority rule, a local search procedure and a GA. The results of the computational experiments showed that the proposed hybrid GA performed remarkably well on a set of SALB Type-1 problems from the literature.

Simario & Vilarinho (2002) presented a mathematical model and an iterative genetic algorithm-based procedure for a MMALB Type-2 problem including parallel workstations. In addition to the aim of minimizing the cycle time, the model aims to balance the workloads among the workstations for the various product models.

McMullen & Tarasewich (2002) applied ant colony optimization technique to solve the ALB problem with the complicating factors of parallel workstations, stochastic task durations, and mixed-models. Performance analysis results confirmed that the ant colony algorithm is competitive with the other heuristic methods such as simulated annealing in terms of several performance measures (e.g., cycle time ratio, design cost, probability of jobs being completed on time).

Agpak & Gökçen (2002) used fuzzy integer programming approach to address U-shaped ALBP. Cycle time, number of stations and work load values were considered as fuzzy variables. This paper is the first study which used fuzzy integer programming for the U-shaped assembly line balancing. The authors applied the proposed model on the Jackson Problem (1956) and solved using GAMS.

Martinez & Duff (2004) dealt with the U-shaped SMALB Type-1 problem. They first solved this problem using 10 heuristic rules adapted from the simple assembly line balancing problem. Then, they modified the GA proposed by Ponnambalam, Aravindan & Mogilesswar (2000). The results of the study showed that optimal or near optimal solutions were produced to improve the current solution.

Stockton et al. (2004a) examined the application of GAs to the SMALBP Type-1. They compared the performance of the GA with a traditional heuristic based solution method: Ranked Positional Weight (RPW). In another study, Stockton et al. (2004b) performed computational experiments in order to define suitable genetic operators and parameter values. These two papers are adopted to complement each other.

Mendes et al. (2005) addressed MMALB problem using Simulated Annealing procedure. Its aim was maximizing the utilization rate of the assembly line for various demand cases. The first step of the study was based on the simulated annealing approach. In the second step, the solutions obtained by the first step were used as an input to discrete event simulation models. These simulation models were run to analyze several performance measures (e.g. resources utilization and flow times). The simulation study provided operational support and helps fine-tune the line configurations.

Agpak & Gökçen (2005) developed 0–1 integer programming models for solving simple assembly line balancing problems. The aim was to establish a balance of assembly line with minimum number of stations and resources.

Levitin et al. (2006) developed an effective method for the robotic assembly line balancing (RALB) problem to maximize the production rate of the line using a GA. Two different methods for fitting the GA to the RALB problem and assigning robots which have different abilities to stations are produced: a recursive assignment procedure and a consecutive assignment procedure. Implementation results showed that the proposed method performed better than the branch and bound procedure.

Lapierre, Ruiz & Soriano (2006) suggested a new Tabu Search (TS) method for solving the Type-1 standard ALBP and non-standard versions of this problem derived from actual life practices. This procedure explored two complementary neighborhoods and integrated several advanced features of TS to enhance its efficiency, robustness and adaptability to real industrial settings. The flexibility of meta-heuristics allowed them to easily adapt their algorithm to the new specifications. In more complex problems, TS has provided better results than several priority-based heuristics.

Bukchin & Rabinowitch (2006) addressed a MMALB problem with relax task assignment restriction. They allowed a common job to be assigned to various stations for various models. Minimizing the total costs of the stations and the task iteration was the objective. An algorithm based on the branch-and-bound procedure has been proposed and tested. Computational experiments showed that the proposed algorithm performed satisfactorily, and provides much better results than the other methods.

Gökçen, Agpak & Benzer (2006) studied on the SMALBP with parallel lines. Since the goal was to balance more than one assembly line together, it would be possible to assign tasks from each line to a multi-skilled operator. A binary integer-programming model was developed with the objective function of minimizing the number of workstations. The implementation results confirmed that the performance of proposed model were competence.

Rahimi-Vahed, Rabbani, Tavakkoli, Torabi & Jolai (2007) considered MMALB problem with three objectives: minimizing total utility work, total setup cost and total production rate. First, mathematical formulations with the considered objectives were provided, then a new approach named multi-objective scatter search (MOSS) was applied to procure various locally Pareto-optimal frontier for the problem. The results showed that the approach outperformed the existing GAs.

Bautista & Pereira (2007) studied a Time and Space constrained ALBPs which are related to the space available around the lines due to alterations in demand in the automobile industry. An ant colony algorithm was used for solving the problem under consideration. This algorithm was tested for two cases of the SALBP-1 and SALBP-2.

Toklu & Ozcan (2008) developed a fuzzy goal programming model with imprecise goal value for each objective for the simple U-shaped line balancing (SULB) problem. There were three fuzzy goals: minimization of number of workstations, sum of processing times and total number of tasks. The performance of the proposed model was compared with the results of the goal programming model proposed by Gökçen & Agpak (2002). The results indicated that the more realistic solutions were obtained in solving the SULB problem.

Corominas, Pastor & Plans (2008) considered the process of rebalancing the line at a motorcycle assembly plant. The plant had to rebalance its assembly line to meet the increasing production in summer period because of the seasonal variation of the demand.

Therefore, production was increased through the hiring of temporary workers. The goal of this study was to minimize the number of necessary temporal operators for a predetermined cycle time and the set of operators. The problem was modeled as a binary linear program (BLP) and solved optimally by using the ILOG CPLEX 9.0 optimizer.

Wu, Jin, Bao & Hu (2008) focused on the two-sided ALB problem (TALBP) with the objective of minimizing the number of opened stations. They proposed a branch and bound algorithm and carried out computational experiments. The results revealed that the proposed method performs well.

Gao, Sun, Wang & Gen (2009) proposed a GA for type–2 robotic assembly line balancing (rALB-II) problem. In this problem, tasks have to be assigned to stations and each station needs to select one of the robots to process the assigned tasks with the aim of minimum cycle time. Based on different neighborhood structures, five local search procedures were developed to raise the search ability of GA. The performance of the proposed hybrid GA was tested on 32 representative rALB-2. For small sized problems, proposed method gave an optimal solution.

A mathematical model that captures both operation time and physical workload was presented by Choi (2009). The author just didn't distribute equal workload by using operation times unlike the past researchers. Besides, he added one more step by considering the physical workloads implemented on the operators. This addressed problem was called Line Balancing Problem for Processing Time and Physical Workload (LBPT&PW). The goal programming model was formulated for the problem under consideration.

Kim, Song & Kim (2009) presented a mathematical formulation for TALBPs with the goal of minimizing the cycle time for a given number of mated stations. The mathematical model was used as a foundation for practical development in the design of two-sided assembly lines. Additionally, the authors constructed a GA to solve efficiently this problem within a reasonable computational time. They adopted the strategy of localized evolution and steady-state reproduction to improve population diversity and search efficiency.

Kara et al. (2009) considered both straight and U-shaped assembly lines. They presented IP models to minimize the number of workstations and cycle time in a fuzzy environment. In the proposed model for SALB, some constraints of the IP model developed by Talbot & Patterson (1986) were considered, and this model is extended by adding a new group of workstation constraints. Computational experiments were carried out to demonstrate the effectiveness of the proposed method, and to compare the performance of different line configurations. The authors claimed that the solution methods are effective and applicable for both straight and U-shaped ALBP.

Ozcan & Toklu (2009-a) presented a Mixed Integer Goal Programming model for precise goals and a Fuzzy Mixed Integer Goal Programming model for imprecise goals. Three objectives were considered in this study: minimization of the number of mated-stations, cycle time, and number of tasks which are assigned to each station. The proposed goal programming models were the first multiple criteria decision making approaches for two-sided ALBP with above-mentioned multiple objectives.

Ozcan & Toklu (2009-b) developed a Tabu Search algorithm for two-sided assembly line balancing problem with the objective of maximizing the line efficiency (i.e., minimizing the number of stations) and minimizing the smoothness index. This algorithm performed well and it found the optimal solutions for some problems.

Ozcan & Toklu (2009-c) also addressed TALBP with the aim of minimizing the number of mated-stations (i.e., the line length) as the primary objective and minimizing the number of stations (i.e., the number of operators) as the secondary objective. They presented a new mixed integer programming model with some additional constraints such as positional, zoning, and tasks constraints. Simulated annealing (SA) approach was also developed and applied to an example problem and tested on several test problems from the literature.

In another study, Ozcan, Cercioglu, Gökçen & Toklu (2009) proposed a TS algorithm for parallel assembly line balancing problem (PALBP) to minimize concurrently inequality of workloads and maximize line efficiency (minimizing number of stations). This study was based on the study of Gökçen et al. (2006), which was entitled “Balancing of parallel assembly lines”. The proposed approach was tested on 82 benchmark problems from the literature. Computational study showed that, the results of the developed solution method were better than Gökçen et al.’s (2006) results.

Ege et al. (2009) considered deterministic PALBP. They developed a branch and bound procedure for minimizing total equipment and workstation opening costs. As a result of their study, they found optimal solutions for medium sized problems and near optimal solutions for large sized problems in reasonable solution times.

Baykasoglu & Dereli (2009) used an ant colony algorithm for solving Simple and U-shaped Assembly Line Balancing Problem with the objective of maximizing line performance that is minimizing the number of workstations. Their suggested algorithm integrated COMSOAL (Computer Method of Sequencing Operations for Assembly Lines), Ranked Positional Weight heuristic and an Ant Colony Optimization based heuristic. The authors obtained promising results from the solution of the considered problems in most of the runs.

Becker & Scholl (2009) introduced an extension of SALBP for the automotive and other industries where large-sized high volume products such as trucks, cars, and machines are produced. In this assembly system, operators perform varied jobs on the same workpiece in parallel. The problem was formulated as a mixed-integer programming model and a solution algorithm was developed based on branch and bound algorithm. The problem considered in this study was called an assembly line balancing problem with variable workplaces (VWALBP).

Bautista & Pereira (2009) studied on the SALBP to find an assignment between tasks and stations while minimizing the number of required stations for a given cycle time (SALBP–1). They wanted to show how an algorithm based on Dynamic Programming (DP) can solve SALBP-1. Thus, they proposed a new procedure which was named Bounded Dynamic Programming. Term Bounded was associated not only with the use of bounds to reduce the state space, but also tried to reduce the solution space while using heuristic rules. This procedure provided an optimal solution with the rate 267 of 269 instances.

Kara et al. (2010) extended the mathematical model presented by Gökcen et al. (2006) for balancing parallel assembly lines with fuzzy goals. The authors presented two goal programming models. In this study, three objectives were considered: optimization of number of workstations, cycle time and task loads of workstations.

Zhang & Cheng (2010) studied on the U-shaped line balancing problem with fuzzy operation times and cycle time. An integer programming formulation for this problem is constructed and solved using LINGO optimization software.

Toksari, Isleyen, Guner & Baykoc (2010) dealt with an assembly line balancing problem involving deterioration tasks and learning effect. A mixed integer nonlinear programming model for this problem was proposed with the aim of minimizing the station number for a given cycle time. The authors compared the results of the solutions with the COMSOAL approach. They obtained the same results for Jackson 11 problem. Nevertheless, COMSOAL approach provided less numbers of stations for other test problems considered in their study.

Scholl et al. (2010) first focused on SALBP Type-1, then its enlargement with various types of assignment constraints. The second problem was named ARALBP-1 (assignment restricted ALBP-1). The ABSALOM methodology which was an extended procedure of the SALOME, a bidirectional branch-and-bound algorithm for the SALBP, was developed. This new methodology was effective in finding optimal or near-optimal solutions for GALBPs with additional assignment restrictions.

Ozcan (2010) considered the balancing of two-sided ALs which has stochastic operation times. A piecewise-linear, chance-constrained, mixed integer programming (CPMIP) model was developed. SA algorithm is proposed as a solution approach, and a heuristic method based on the COMSOAL (Computer Method of Sequencing Operations for Assembly Lines) was also used to compare the results for a set of test problems. We can state that this paper was the first study which was done for two-sided ALs with stochastic operation times.

Blum & Miralles (2010) presented an algorithm based on beam search to solve simultaneously worker assignment and line balancing problem. The problem was named as ALWABP-2. The goal was to find such solution that satisfies all the assignment constraints and minimize the cycle time. The obtained results showed that this algorithm was the best-performing method for the ALWABP-2 so far.

Zacharia & Nearchou (2010) addressed fuzzy assembly line balancing problem. Minimizing the fuzzy cycle time, the fuzzy delay time and the fuzzy smoothness index in the line was the aim of the study. In this work, the authors presented a new multi-objective GA. The computational study verified that multi-objective GA was a powerful approach to solve fuzzy line balancing problems.

Akpınar & Bayhan (2011) solved MMALBP using hybrid genetic algorithm (hGA) considering minimizing the number of workstations, maximizing the workload smoothness. They explored genetic algorithms by hybridizing the three well known heuristics; Kilbridge & Wester Heuristic, Phase-I of Moodie & Young Method and Ranked Positional Weight Technique. The proposed method was compared with these three heuristics, traditional GA, Simulated Annealing. The results showed that hGA outperformed the compared methods.

In another study, Bees Algorithm (BA) was adopted to solve TALBP with zoning constraint so as to minimize the number of stations for a given cycle time by Ozbakır & Tapkan (2011). The performance of the algorithm is compared with several algorithms from the literature such as ant colony optimization and tabu search, and exact solution approaches. The computational study was carried on two categories; without any constraints and with zoning constraint on test problems from the literature. BA performed considerably better than ant colony based approach in terms of number of workstations and CPU times.

Kılınçcı (2011) used a new heuristic method which was called Firing Sequence Backward (FSb) algorithm based on Petri Nets to solve the SALBP-1. The method's efficiency was tested on Talbot's and Hoffmann's benchmark datasets according to several performance measures. FSb showed the best performance in the single pass heuristics for all performance measures. The author also compared *FSb* with two Petri Net based heuristic approaches which is based on reachability analysis and P-invariants of the PN model. According to the performance analysis, FSb was the most efficient heuristic based on PNs to solve SALBP-1.

Nearchou (2011) presented a novel method based on Particle Swarm Optimization (PSO) for the SALBP. Two criteria were simultaneously considered: to maximize the production rate of the line and to maximize the workload smoothness. Four versions of the PSO algorithm, which differ in the weighted method used to estimate the weights in the evaluation function, were implemented. These four versions of the PSO algorithm and two existing multi objective GAs were compared on benchmark problems from the literature. The PSO algorithm gave better results in terms of solution quality for SALBPs.

Yagmahan (2011) dealt with MMALBP with the goals of minimizing the balance delay, the smoothness index between stations and the smoothness index within stations for a given cycle time. To solve the problem, multi objective ant colony optimization approach was used. In order to verify the performance of the proposed algorithm, computational experiments were conducted on test problems. Proposed algorithm performed better than ranked positional weight method, genetic algorithm and artificial immune algorithm.

Cakir, Altıparmak & Dengiz (2011) studied on multi-objective optimization of a single-model stochastic ALBP with parallel stations. The objectives were: minimization of the smoothness index and minimization of the design cost. They proposed new solution algorithm based on simulated annealing (SA), called m_SAA. This algorithm is constructed by a multinomial probability mass function approach, a search space and memory (which is called the tabu list), repair algorithms and a diversification strategy. The authors compared the performances of m_SAA and multi-objective simulated annealing (MOSA) method on 24 well known test problems in the ALBP literature. This comparison showed that m_SAA outperforms MOSA method.

Ozbakır, Baykasoglu, Gorkemli & Gorkemli (2011) developed multiple colony ant algorithm for balancing bi-objective parallel assembly lines considering minimizing the idle time of workstations and maximizing the line efficiency. The proposed approach was tested on the benchmark problems. Performance of the approach was compared with existing methods, i.e. the heuristic algorithm in Gokcen et al. (2006), the mathematical programming model in Scholl and Boysen (2009) , the branch and bound based exact solution procedure in Scholl and Boysen  (2008, 2009) and tabu search algorithm in Ozcan et al.(2009). The authors concluded that the proposed approach was very effective in solving the ALBP considered.

In the next study, Tapkan, Ozbakır & Baykasoglu (2012) considered balancing problem of a two-sided assembly line with positional, zoning and synchronous task constraints. First, they presented a mathematical programming model to describe the problem formally. Then, Bees Algorithm (BA) and Artificial Bee Colony (ABC) algorithm have been applied to the fully constrained two-sided assembly line balancing problem to obtain a balanced line. The aim of the study was to minimize the number of workstations. The authors compared the performances of BA and ABC algorithms and concluded that the two algorithms provided approximately same results.

In Chutima & Chimklai's study (2012), two-sided ALB problem was addressed with the aim of optimizing the number of mated stations, the number of workstations, and two conflicting sub-objectives to be optimized simultaneously, i.e. work relatedness and workload smoothness. The performance of Particle Swarm Optimization with negative knowledge (PSONK) is compared with COMSOAL, Non-dominated Sorting Genetic Algorithm-II, and Discrete Particle Swarm Optimization on several scenarios. The proposed method outperformed the other methods compared.

Hamzadayı & Yıldız (2012) presented Priority-Based Genetic Algorithm (PGA) for the mixed-model U-shape assembly line balancing and model sequencing problems (MMUL/BS) with parallel workstations and zoning constraints. Simulated annealing based fitness evaluation approach (SABFEA) was developed in order to make fitness function calculations. The new fitness function was adapted to MMULs for minimizing the number of workstations and smoothing the workload between-within workstations considering various cycle time considerations. The results indicated that SABFEA works with PGA very concordantly; and it was an effective method in solving MMUL/BS with parallel workstations and zoning constraints.

Chen et al. (2012) developed a Grouping Genetic Algorithm (GGA) for ALBP of sewing lines with different labor skill levels. GGA can allocate workload among machines as evenly as possible for different labor skill levels, so the mean absolute deviations (MAD) could be minimized. MAD, cycle time and operator utilization were used to evaluate GGA's performance. GGA's best parameter setting of population size, crossover rate and mutation rate were got through 100, 0.8 and 0.005, respectively.

Table 3.1 Literature review

| ***Year*** | ***Researchers*** | ***Problem Type*** | ***Objective function*** | ***Solution approach*** |
|---|---|---|---|---|
| 1995 | Rubinovitz & Levitin | SALBP | Type-2 | Genetic Algorithm |
| 1995 | Tsujimura et al. | SMALBP | Type-1 | Genetic Algorithm |
| 1995 | Sawik | MMALBP | Type-1 | Integer Programming |
| 1995 | Kabir & Tabucanon | MuMALBP | Type-1 | Multiattribute decision making approach |
| 1996 | Leu et al. | MMALBP | Equalizing inventory and work load | Genetic Algorithm |
| 1996 | Kim et al. | MMALBP | Type-1-2-3-4-5 | Genetic Algorithm |
| 1996 | Klein & Scholl | SALBP | Type-2 | Branch & Bound |
| 1996 | Suresh et al. | SMALBP | Type-1 | Genetic Algorithm |
| 1997 | Ugurdag et al. | SALBP | Type-2 | Integer programming |
| 1997 | Gökçen & Erel | MMALBP | Type-1 | Goal programming |
| 1998 | Kim et al. | SALBP | Type-2 | Genetic Algorithm |
| 1998 | Sarker & Pan | MMALBP | Minimizing the total cost of utility and idle times | Integer programming |
| 1998 | Gökçen & Erel | MMALBP | Type-1 | Binary integer programming |
| 1998 | Ajenblit & Wainwright | SMALBP | Type-1 | Genetic Algorithm |
| 1998 | Chan et al. | SMALBP | Type-1 | Genetic Algorithm |
| 1998 | Hyun et al. | MMALBP | Minimizing total utility work, keeping a constant rate of part usage and minimizing total setup cost | Genetic Algorithm |
| 1999 | Sarin et al. | SMALBP | Minimizing total labor cost and expected incompletion cost | Dynamic programming and branch and bound |
| 1999 | Tamura et al. | MMALBP | Levelling the part usage rates and workloads | Goal chasing method, tabu search and dynamic programming |
| 1999 | Scholl & Klein | SALBP | Type-1 | Branch & Bound |
| 1999 | Erel & Gökçen | MMALBP | Minimizing the sum of idle times | Network programming |

Table 3.1 (cont.) Literature review

| ***Year*** | ***Researchers*** | ***Problem Type*** | ***Objective function*** | ***Solution approach*** |
|---|---|---|---|---|
| 2000 | Bautista et al. | SALBP | Type-1-2 | Genetic Algorithm |
| 2000 | Frey | MMALBP | Finding a valid sequence | Petri nets-T invariants |
| 2000 | Kim et al. | SMALBP | Type-1 | Genetic Algorithm |
| 2000 | Sabuncuoglu et al. | SALBP | Type-1 | Genetic Algorithm |
| 2000 | Ponnambalam et al. | SALBP | Type-1-3 | Genetic Algorithm |
| 2000 | Sawik | Flexible ALBP | Balancing station workloads and minimizing total transportation time | Integer programming and LP based heuristic |
| 2001 | Carnahan et al. | SALBP | Type-2 | Genetic Algorithm |
| 2001 | Simaria & Vilarinho | MMALBP | Type-2 | Simulated annealing |
| 2001 | Ji et al. | SALBP | Type-2 | Integer programming, genetic algorithm |
| 2001 | Lee et al. | MMALBP | Maximizing work relatedness & slackness | Grouping assignment procedure |
| 2002 | Chen et al. | Assembly planning | Type-2 | A hybrid genetic algorithm |
| 2002 | Pastor et al. | MuMALBP | Type-2 | Tabu Search |
| 2002 | Nicosia et al. | SALBP | Minimizing the cost of the workstations | Dynamic programming and branch and bound |
| 2002 | Goncalves & De Almedia | SALBP | Type-1 | Genetic Algorithm |
| 2002 | Valente et al. | SMALBP | Type-2 | Genetic Algorithm |
| 2002 | Agpak & Gökçen | SULBP | Type-1 | Fuzzy integer programming |
| 2003 | McMullen & Tarasewich | MMALBP | Minimizing the required number of workers | Ant techniques |
| 2004 | Simaria & Vilarinho | MMALBP | Type-2 | Genetic Algorithm |
| 2004 | Brudaru & Valmar | SMALBP | Type-1 | A hybrid genetic algorithm |
| 2004 | Martinez & Duff | SMALBP | Type-1 | Genetic Algorithm |
| 2004 (a,b) | Stockton et al. | SALBP | Type-1 | Genetic Algorithm |
| 2005 | Mendes et al. | MMALBP | Maximizing the use of the AL for different demand scenarios | Simulated annealing |
| 2005 | Brown & Sumichrast | SALBP | Type-1 | Grouping Genetic Algorithm |
| 2005 | Agpak & Gökçen | SALBP | Type-1 | Integer Programming |
| 2006 | Levitin et al. | RALBP (Robotic ALBP) | Type-2 | Genetic Algorithm |

Table 3.1 (cont.) Literature review

| ***Year*** | ***Researchers*** | ***Problem Type*** | ***Objective function*** | ***Solution approach*** |
|---|---|---|---|---|
| 2006 | Lapierre et al. | SALBP | Type-1 | Tabu Search |
| 2006 | Bukchin & Rabinowitch | MMALBP | Minimizing the total costs of the workstations and the task iteration | Branch & Bound |
| 2006 | Gökçen et al. | PALBP (Parallel ALBP) | Type-1 | Binary Integer Programming |
| 2007 | Rahimi-Vahed et al. | MMALBP | Minimizing total utility work, total production rate and total setup cost | Multi-objective genetic algorithm |
| 2007 | Bautista & Pereira | SALBP | Type-1-2 | Ant algorithms |
| 2008 | Toklu & Ozcan | Simple U-Line Balancing (SULB) | Minimizing the number of workstations, sum of processing times and total number of tasks | Fuzzy goal programming |
| 2008 | Corominas et al. | MMALBP | Minimizing the number of necessary temporary operators, given a cycle time | Binary Linear Programming |
| 2008 | Wu et al. | TALBP | Minimizing the number of opened stations | Branch and Bound |
| 2009 | Gao et al. | Robotic Assembly Line Balancing Problem(RALB) | Type-2 | Genetic algorithms, Local search, Neighborhood structure |
| 2009-a | Ozcan & Toklu | TALBP | Minimizing the number of mated-stations, cycle time and number of tasks | Goal programming and fuzzy goal programming |
| 2009-b | Ozcan & Toklu | TALBP | Maximize the line efficiency and minimize the smoothness index | Tabu Search |
| 2009-c | Ozcan&Toklu | TALBP | Type-I | Integer programming and simulated annealing algorithm |
| 2009 | Choi | MMALBP | Minimizing the overload of the physical workload | Goal Programming |
| 2009 | Kim et al. | TALBP | Type-1 | Integer programming, genetic algorithm |
| 2009 | Kara et al. | SALBP-SULBP | Minimizing the combination of under achievement amount of the cycle time goal and number of workstations goal | Binary fuzzy goal programming |
| 2009 | Ozcan et al. | PALBP | Maximizing line efficiency and minimizing variation of workloads | Tabu Search |

Table 3.1 (cont.) Literature review

| ***Year*** | ***Researchers*** | ***Problem Type*** | ***Objective function*** | ***Solution approach*** |
|---|---|---|---|---|
| 2009 | Simaria & Vilarinho | TALBP | Type-I (with zoning restrictions) | Integer programming and ant colony algorithm |
| 2009 | Ege et al. | PALBP | Minimizing sum of station opening and tooling/equipment costs | Branch & Bound |
| 2009 | Baykasoglu & Dereli | SALBP and U-ALBP | Minimizing the number of workstations (or line efficiency) | Ant colony based algorithm, COMSOAL, Ranked Positional Weight |
| 2009 | Becker & Scholl | SALBP | Maximizing the line efficiency | Branch & Bound |
| 2009 | Bautista & Pereira | SALBP | Type-1 | Dynamic programming |
| 2010 | Kara et al. | PALBP | Minimizing the total idle times of assembly lines and total number of workstations | Fuzzy goal programming |
| 2010 | Zhang & Cheng | SULBP | Type-1 | Fuzzy integer programming |
| 2010 | Toksari et al. | SALBP | Type-1 | Mixed Integer Nonlinear Programming, COMSOAL approach |
| 2010 | Scholl et al. | SALBP | Type-1 | ABSALOM (Branch & Bound procedure) |
| 2010 | Ozcan | TALBP | Minimizing the number of mated-stations and minimizing the number of stations | Mixed Integer Programming, Simulated Annealing Algorithm, COMSOAL |
| 2010 | Blum & Miralles | Assembly line worker assignment and balancing problem (ALWABP) | Type-2 | Beam search |
| 2010 | Zacharia & Nearchou | Fuzzy SALBP | Type-2 | Genetic Algorithm |
| 2011 | Akpınar & Bayhan | MMALBP | minimizing the number of workstations, maximizing the workload smoothness between workstations and within workstations | Hybrid Genetic Algorithm |
| 2011 | Ozbakır & Tapkan | TALBP | Type-1 | Bees Algorithm |

Table 3.1 (cont.) Literature review

| ***Year*** | ***Researchers*** | ***Problem Type*** | ***Objective function*** | ***Solution approach*** |
|---|---|---|---|---|
| 2011 | Kılınççı | SALBP | Type-1 | Firing Sequence Backward based on Petri Net Approach |
| 2011 | Nearchou | SALBP | maximizing the production rate of the line and to maximizing the workload smoothing | Particle Swarm Optimization |
| 2011 | Yagmahan | MMALBP | minimizing the balance delay, the smoothness index between stations and within stations | Ant Colony Optimization |
| 2011 | Cakır et al | SALBP with Parallel Stations | minimization of the smoothness index and minimization of the design cost | Simulated Anneling |
| 2011 | Ozbakır et al. | PALBP | minimizing the idle time of workstations and maximizing the line efficiency | Multiple Colony Ant Algorithm |
| 2012 | Tapkan et al. | TALBP | Type-1 | Bees Algorithm, Artifical Bee Colony |
| 2012 | Chutima & Chimklai | TALBP | optimizing the number of mated stations, the number of workstations, work relatedness and workload smoothness simultaneously. | Particle Swarm Optimization |
| 2012 | Hamzadayı & Yıldız | MMALBP | minimizing the number of workstations and smoothing the workload between-within workstations | Priority-Based Genetic Algorithm |
| 2012 | Yolmeh & Kianfar | SUALBSP | minimizing cycle time for a given number of stations. | Hybrid Genetic Algorithm |
| 2012 | Chen et al. | SALBP | minimizing the mean absolute deviations | Grouping Genetic Algorithm |

## 3.2 Findings of the Literature Review

When we overview abovementioned papers in terms of problem types, we can see that Simple Assembly Line Balancing Problem was mostly studied problem, then the second one was Mixed Model Assembly Line Balancing Problem. In Figure 3.1, the problem types which handled by the researches are presented in a comparative manner.

SALBP➔ Simple Assembly Line Balancing Problem
GALBP➔ General Assembly Line Balancing Problem (Parallel ALBP, U-shaped ALBP, Two-sided ALBP, Robotic ALBP)
SMALBP➔ Single Model Assembly Line Balancing Problem
MMALBP➔Mixed Model Assembly Line Balancing Problem
MuMALBP➔ Multi Model Assembly Line Balancing Problem
ALWABP➔ Assembly Line Worker Assignment and Balancing Problem

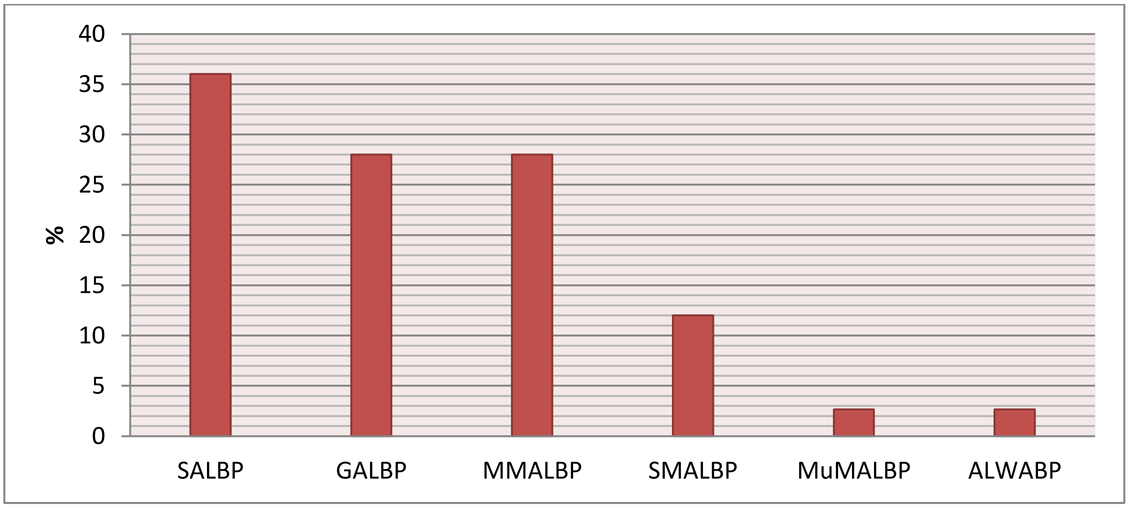


Figure 3.1 Problem types between 1995 and 2012

Figure 3.2 illustrates the percentages of solution approaches used in the studies between 1997 and 2012.

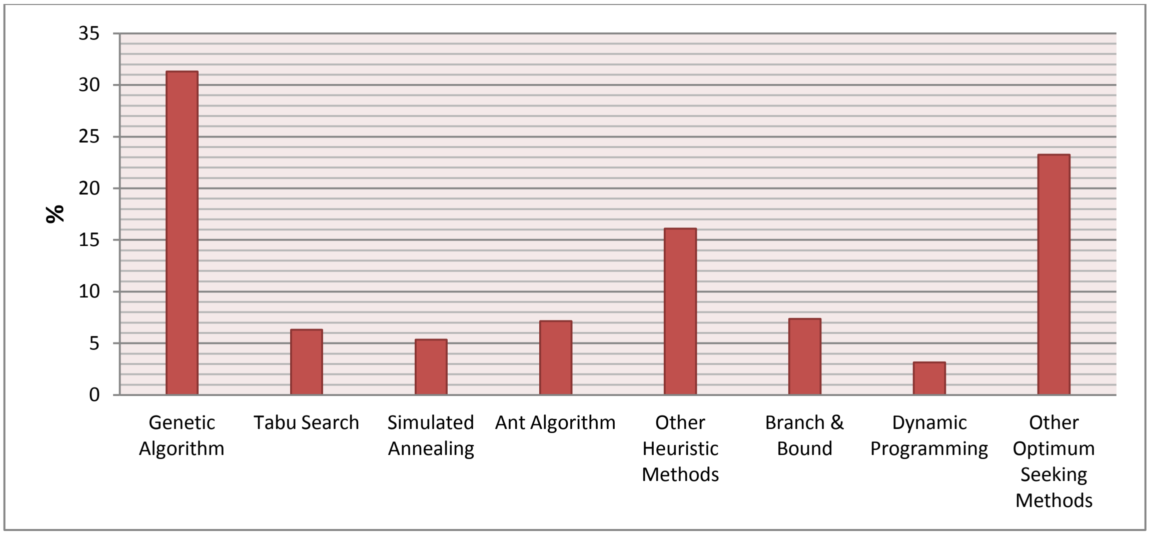


Figure 3.2 Solution approaches used in the literature between 1995 and 2012

As we defined as a general title on the graph of solution approaches, other heuristic methods contain beam search, bees algorithm, particle swarm optimization algorithm, simple heuristic methods and COMSOAL method. Additionally, the other optimum seeking methods contain integer programming, goal programming and linear programming.

As it can be seen in Figure 3.2, genetic algorithms are the most widely used solution method to solve ALBPs by the researchers. Mathematical programming, branch and bound and tabu search are the other common solution approaches.

Moreover, if we consider GALBP in detail, we can see that just few researchers in the literature considered two-sided ALBP. In these studies, the authors generally focused on minimizing number of workstations. Researchers widely used mathematical modeling, tabu search, simulated annealing and genetic algorithm for solving two-sided ALBP.

When we analyzed the studies in literature and implementations in practice, we can realize that applications of real and academic world differ from each other. Possible reasons of the gap between theoretical researches and practical applications are as follows:

- ✓ Researchers are interested in modeling simple problems with convenient assumptions to aid in solving & benchmarking solutions.
- ✓ Focal point is on “optimality” rather than on “practicality”.
- ✓ Scientific results could not be adapted to real cases (i.e., problems are not generic)
- ✓ The problems were covered, but could not be solved to satisfaction.
- ✓ Line balancing is not the only manufacturing problem (Time constraint is presented in the practical world limits good analysis)

The challenge lies in putting theory into practice, which involves simultaneously handling efficiency, practical assignment restrictions, competitiveness, and human-related factors. Figure 3.3 shows major differences between academic researches and real world application.

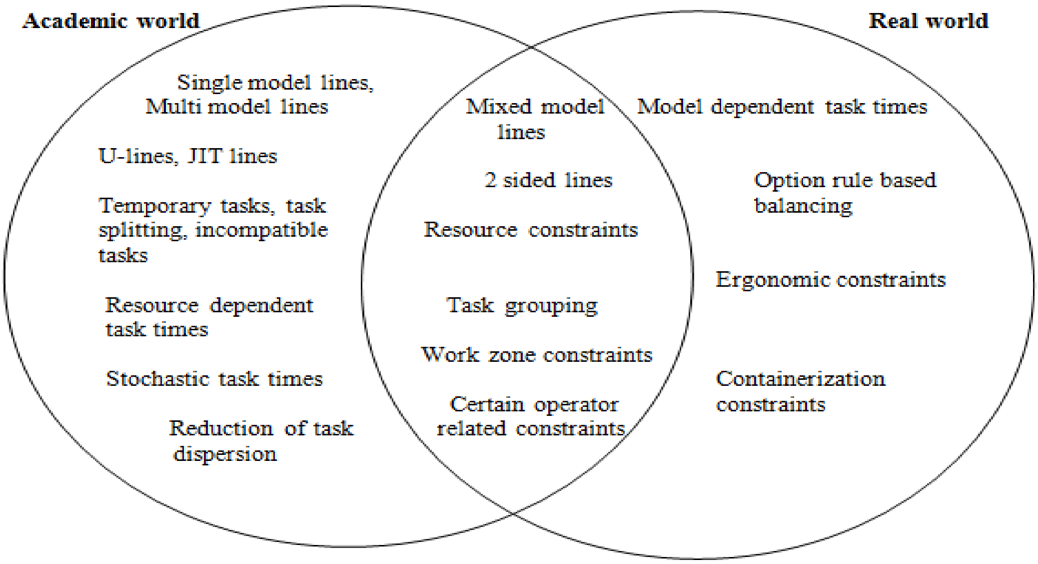


Figure 3.3 Line Balancing in Academic vs. Industry World (Sly PE, 2007)

# CHAPTER FOUR
# TWO-SIDED ASSEMBLY LINE BALANCING: AN APPLICATION IN AN INTERNATIONAL HOME APPLIANCES COMPANY

## 4.1 Introduction

In this chapter, we presented a two-sided assembly line balancing problem in a manufacturing system. First, we define the assembly line under consideration, data of the problem and determination of the constraints. Then, we applied two solution approaches, Group Assignment Procedure and Teaching-learning Based Optimization Algorithm, to achieve an efficient line balance for the relevant problem.

## 4.2 Problem Definition

The company is one of the leading manufacturers and distributors of major domestic appliances both in Turkey and Europe; especially in Italy, the UK, and Russia. Its products are fridges, freezers, dryers, washing machines, cookers, dishwashers, hoods, ovens and hobs. In Manisa plant, only cooling products are produced which can be ranged with 35 main models and totally 300 models. General classes of cooling products are as follows;

1. Coolers (Refrigerators)
2. Coolers-Freezers (Refrigerators)
    - No Frost System
    - Partial No Frost System
    - Turbo System
    - Static System
3. Vertical Freezers

*Cooler products* are divided into three main categories: single door, double door and combi models. In combi model, fridge part is at the upper side and freezer part is at the down side of the refrigerator. *Combi models* are also divided into three sub-categories: normal combi, 3D and 4D. Normal combi is composed of two doors: fridge and freezer. 3D models have one fridge door and two freezer drawers, and 4D models are composed of two fridge doors and two freezer drawers. In addition, external aesthetics of all models may differ according to customer demands.

In terms of product dimension, the product groups are also subdivided into three groups: 55, 60, and 70cm. Single door, double door and normal combi models (static, no frost or partial no frost system) are produced in the sizes of 55 cm and 60 cm. Also, double door, normal combi, 3D and 4D combi models (static and no frost system) are produced in the size of 70 cm.

The company has 15 % domestic market share and 85 % foreign market share. Mostly, products which have 70 cm platform are distributed to Turkey market.

In Manisa plant, there are six main divisions: plastic area, mechanic area, paint shop, door area, production area (pre-assembly area and assembly area) and packing area. In plastic area, plastics supplied by sub-industry are prepared as plastic plaques and moved to pre-assembly area to form the plaque and fill polyurethane. Steel roles supplied by sub-industry are cut and formed in accordance with models in the mechanic area. Metal sheets transferred from mechanic area are painted in the paint shop if white colored products are produced. The metal sheets of the other colored products are not painted. After cutting and forming of the roles, sheets are directly sent to pre-assembly area. In the door area, doors of products are prepared, filled by polyurethane, and then transferred to assembly area. Production area involves 7 assembly lines, and each assembly line is composed of two stages: pre-assembly and assembly line. In the first stage, products are processed on pre-assembly lines, where tasks are performed with inner liner of refrigerator and side-and-top panels. These tasks include drilling, banding, siliconizing, arranging, fixing operations, and placing some components on inner liner so that two opposite workers can execute various operations on the same individual item in parallel. After pre-assembly stage, products are transferred to assembly area where doors, internal and external accessories, thermodynamic components are installed. When finished products are ready, they are sent to test area. Products which pass all tests are transferred to packing area, and then to the warehouse.

In this study, we considered pre-assembly line for 4D combi model at Line 1. Two groups of tasks are performed in this line: assembly operations of plastic parts and assembly operations of panels. Figure 4.1 below illustrates the precedence diagram that indicates the order of performing each task. A detailed time study has been performed to determine the task times. Data of the pre-assembly line for the 4D model is presented in Table 4.1. There are 70 tasks; each of them has a specific operation direction (e.g., left, right or either side). Thus, the line can be considered as a two-sided assembly line. The second column lists operation times of each tasks in terms of cts, 1cts unit time equals 0.01 minute. Letters written in the third column indicate the direction of related operation. *L* shows that task should be assigned to the left side of the line, *R* shows that task should be assigned to the right side of the line, *E* represents that task can be assigned to either side of the line.

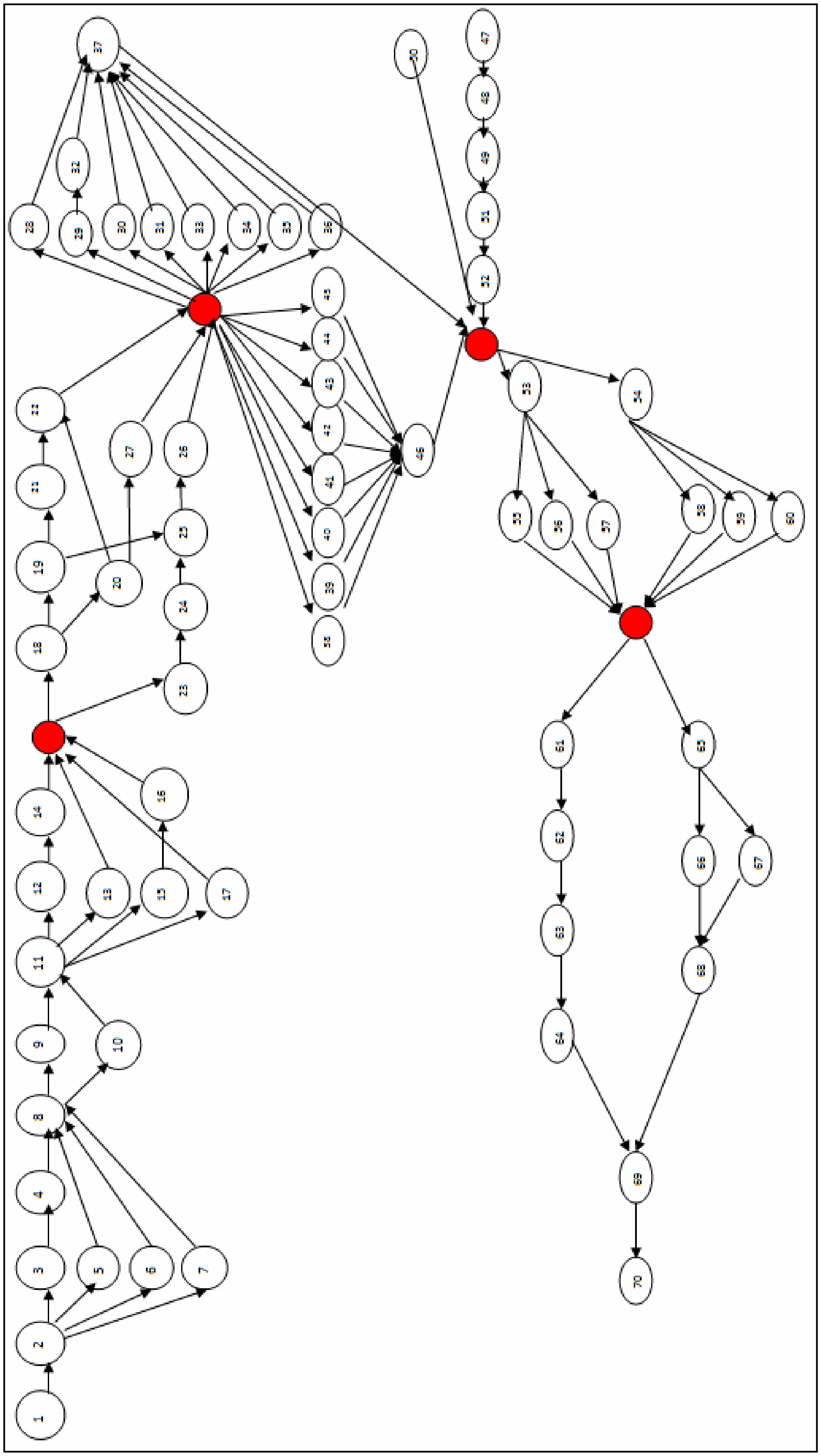


Figure 4.1 Precedence diagram

Table 4.1 Data of the problem

| Task No. | Processing Times- cts | Operation directions | Immediate precedence(s) |
|---|---|---|---|
| 1 | 82 | E | - |
| 2 | 34 | E | 1 |
| 3 | 8 | E | 2 |
| 4 | 11 | E | 3 |
| 5 | 22 | E | 2 |
| 6 | 11 | E | 2 |
| 7 | 9 | E | 2 |
| 8 | 30 | E | 4,5,6,7 |
| 9 | 13 | E | 8 |
| 10 | 38 | E | 8 |
| 11 | 24 | E | 9,10 |
| 12 | 24 | L | 11 |
| 13 | 20 | E | 11 |
| 14 | 10 | L | 12 |
| 15 | 18 | L | 11 |
| 16 | 10 | L | 15 |
| 17 | 16 | E | 11 |
| 18 | 37 | R | 13,14,16,17 |
| 19 | 34 | R | 18 |
| 20 | 23 | L | 18 |
| 21 | 32 | R | 19 |
| 22 | 16 | R | 20,21 |
| 23 | 29 | E | 13,14,16,17 |
| 24 | 47 | L | 18,23 |
| 25 | 7 | R | 19,24 |
| 26 | 9 | R | 25 |
| 27 | 36 | L | 20 |
| 28 | 10 | R | 22,26,27 |
| 29 | 17 | R | 22,26,27 |
| 30 | 22 | R | 22,26,27 |
| 31 | 9 | R | 22,26,27 |
| 32 | 38 | R | 29 |
| 33 | 8 | L | 22,26,27 |
| 34 | 11 | E | 22,26,27 |
| 35 | 18 | R | 22,26,27 |
| 36 | 31 | R | 22,26,27 |
| 37 | 10 | R | 28,30,31,32,33,34,35,36 |
| 38 | 10 | L | 22,26,27 |
| 39 | 15 | L | 22,26,27 |
| 40 | 29 | L | 22,26,27 |
| 41 | 34 | E | 22,26,27 |
| 42 | 26 | L | 22,26,27 |
| 43 | 16 | L | 22,26,27 |
| 44 | 12 | L | 22,26,27 |
| 45 | 13 | E | 22,26,27 |

Table 4.1 (cont.) Data of the problem

| **Task No.** | **Processing Times- cts** | **Operation directions** | **Immediate precedence(s)** |
|---|---|---|---|
| 46 | 10 | L | 38,39,40,41,42,43,45,44 |
| 47 | 19 | L | - |
| 48 | 10 | L | 47 |
| 49 | 51 | L | 48 |
| 50 | 9 | E | - |
| 51 | 69 | R | 49 |
| 52 | 53 | R | 51 |
| 53 | 6 | R | 37,44,50,52 |
| 54 | 25 | R | 53 |
| 55 | 15 | R | 53 |
| 56 | 15 | E | 53 |
| 57 | 26 | R | 54,55,56,62,63,64 |
| 58 | 25 | R | 57 |
| 59 | 10 | R | 58 |
| 60 | 20 | R | 59 |
| 61 | 6 | L | 37,44,50,52 |
| 62 | 14 | E | 61 |
| 63 | 14 | L | 61 |
| 64 | 28 | E | 61 |
| 65 | 38 | L | 54,55,56,62,63,64 |
| 66 | 24 | L | 65 |
| 67 | 15 | L | 65 |
| 68 | 8 | L | 66,67 |
| 69 | 156 | E | 60,68 |
| 70 | 162 | E | 69 |

In addition to the precedence relations among the tasks, the following assignment restrictions should also be taken into account while balancing the two-sided ALBP considered in this study:

1. Total task times in the first station has to be less or equal to 137 cts because of the machine capacity existed in this station.
2. Tasks 19 and 25 should be performed together and therefore assigned to the same station. (*inclusion/positive zoning constraint*)
3. Tasks 20 and 27 should also be performed together and therefore assigned to the same station. (*inclusion/positive zoning constraint*)
4. Tasks 69 and 70 represent quality control and fixture operations, respectively. Thus, their operation times are not certain; they are changeable in some cases. In

order to reduce the cycle time variation, these tasks should be executed on the different stations with no other tasks assigned. (*exclusion/negative zoning constraint*)

5. Right tasks have to be assigned to a station at the right side of the line, and left tasks have to be assigned to a left station. Tasks, which can be done at any side of the line, should be assigned to a right or left station considering sum of the task times at each station. (*side restrictions*)
6. Sum of the task times assigned to a station should not exceed the cycle time (*cycle time constraint*).

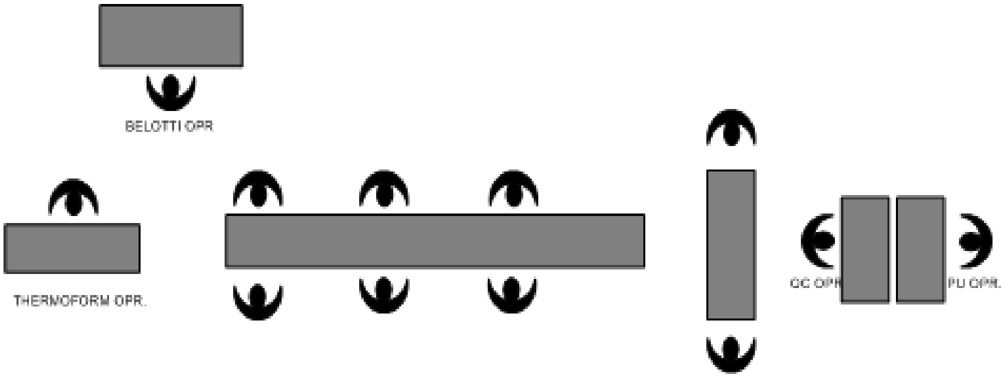


Figure 4.2 Current layout of pre-assembly line at Line 1

Current layout of the pre-assembly line, which has 12 workstations, is depicted in Figure 4.2. Initial workloads of the workstations are displayed in Figure 4.3. Cycle time (239cts) is marked with bold line in the graphic. It can be easily noticed that workstations are not fulfilled and there are quite idle times with respect to the cycle time of 239 cts. Due to the size of the parts, some of the stations are mated and the others work as single stations. Main goal is to improve the line balance implemented by the company for a given cycle time and also considering workload smoothness between workstations.

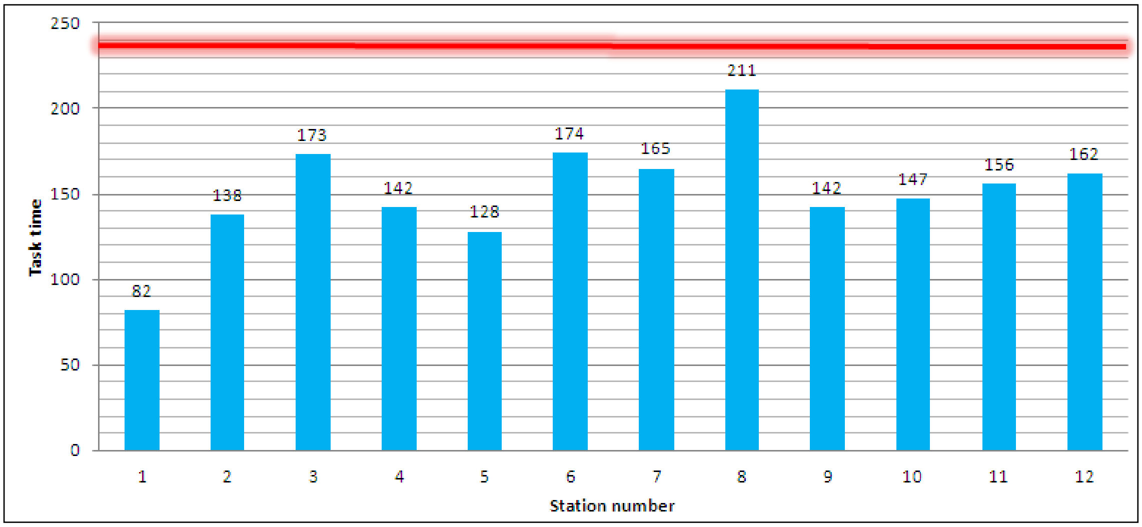


Figure 4.3 Initial workloads of the workstations

## 4.3 Balancing of the Two-sided Assembly Line

The essential difference between the assignment of tasks in one-sided lines and in two-sided lines is mainly related with the sequence in which the tasks are performed. In one-sided lines, the sequence of the tasks within a workstation is not important considering providing precedence relations. But in two-sided lines, sequencing is a critical job for an efficient assignment of tasks. Tasks performed opposite sides of the line can conflict with each other through precedence constraints which might cause idle time if a workstation needs to wait for a predecessor task to be completed at the opposite side of the line (Gunasekaran & Sandhu, 2010).

In this study, we first solved the line balancing problem with additional assignment restrictions by employing a heuristic method based on a group assignment procedure which is commonly used for two-sided ALB. Then, Teaching-Learning Based Optimization Algorithm, which has been recently developed as a solution method for mechanical design problems, was used to solve the problem under consideration. This algorithm is a population-based optimization algorithm like most of the other well-known meta-heuristics, such as Genetic Algorithm and Ant Colony Optimization Algorithm. The goal of choosing this algorithm as a solution approach to our problem was to investigate its performance for solving two-sided assembly line balancing problems.

In the following sections, detailed explanations relevant to these two approaches are given, and computational results are compared.

### *4.3.1 Group Assignment Procedure*

Group Assignment Procedure is first defined by Lee et al. (2001). The procedure begins with forming initial task groups considering operation directions of the tasks. It is disallowed for a group to contain both *L* and *R* tasks. If all tasks in a group are *E* tasks, the group can be allocated to any side. There are two rules (direction rules-DR) to determine the operation directions for such groups:

**DR 1:** Set the operation direction to the side where tasks can be started earlier.

**DR 2:** If the start time at both sides is the same, set the operation direction to the side where it is expected to carry out a less amount of tasks (total operation time of unassigned *L* or *R* tasks)

Then, the following sequencing rules (sequence rules-SR) are used to determine the sequence of tasks:

**SR 1:** Select the task whose start time is the earliest.

**SR 2:** Select the task (*i*) such that it has immediate succeeding tasks that are not contained in the task group currently considered and the operation directions of the succeeding tasks are either opposite to *i*’s operation direction or E-type. When the number of such tasks is more than one, the task with the largest operation time is selected (a task is selected at random to break ties).

The above procedure of constructing task groups is summarized below in an algorithmic form. Let *U* denote the set of tasks that are not assigned yet, and $G_i$ be a task group consisting of task *i* and all of its precedent tasks in *U*.

**Step 1.** Set *TU* = *U*, *FS* = empty.

**Step 2.** If *TU* = empty, then go to Step 7. Otherwise, choose an arbitrary task *i* among the tasks in *TU* and have no precedent task.

**Step 3.** Identify $G_i$. If $G_i$ contains both left and right tasks, then remove task *i*, and all its succeeding tasks from $T_u$ and go to Step 2.

**Step 4.** Determine the operation direction of $G_i$. If $G_i$ has no R-task (L-task), set its operation direction to left (right). Otherwise, all the tasks are E-tasks, determine its direction using direction rules DR 1 and DR 2.

**Step 5.** Determine the sequence of tasks in $G_i$ using sequencing rules SR1 and SR2.

**Step 6.** If the last task in $G_i$ can be completed before the cycle time, update *FS* as $FS=FS\ U\ \{G_i\}$, delete task *i* from $T_u$. And go to Step 2. Otherwise, remove task *i* and all its succeeding tasks from $T_u$ and go to Step 2.

**Step 7.** For every task group of *FS*, remove it from *FS* if it is a proper subset of another task group of *FS*.

The candidate task groups are produced by procedures which do not violate precedence relations, cycle time restriction, and operation direction constraints. Thus, the following assignment rules (AR) can be defined:

**AR 1.** Select the task group that can be started at the earliest time.

**AR 2.** Select the task group that involves the minimum delay.

**AR 3.** Select the task group that requires the maximum operation time.

Using the rules for grouping and assigning tasks, an iterative procedure is developed to solve two-sided ALBP. Let $j$ and $j'$, respectively, denote a left side and right side station of a mated-station, $S_j$ denote the start time at station $j$. $D_k$ and $T_k$, respectively, denote the amount of delay and total operation time required for performing the tasks in $G_k$. The iterative procedure developed by Lee et al. (2001) can be stated as follows:

**Step 1.** Set up $j$=1, $j'$=$j$+1, $S_j$= $S_{j'}$=0, and $U$=the set of all the tasks to be assigned.

**Step 2.** Run the task grouping procedure, which identifies $FS = \{G_1, G_2, \ldots G_k\}$. If $FS$=empty, go to Step 6.

**Step 3.** For every $G_k$, $k=1, 2, \ldots K$, compute $D_k$ and $T_k$.

**Step 4.** Identify one task group $G_r$ from $FS$ using the AR.

**Step 5.** Assign $G_r$ to a station $j$ (or $j'$) according to its operation direction, and update $S_j$=$S_j$+$D_r$+$T_r$ (or $S_{j'}$=$S_{j'}$+$D_r$+$T_r$). $U=U - \{G_r\}$, and go to Step 2.

**Step 6.** If $U$=empty, set $j$=$j'$+1, $j'$=$j$+1, $S_j$=$S_{j'}$=0, and go to Step 2. Otherwise, stop.

When we applied the above procedure to our problem, we acquired the implementation steps presented in Appendix-A. According to these steps, a new balance of the line, presented in Figure 4.4, is achieved. In this solution, number of opened workstations is 9, and smoothness index of the workload is equal to 0.14.

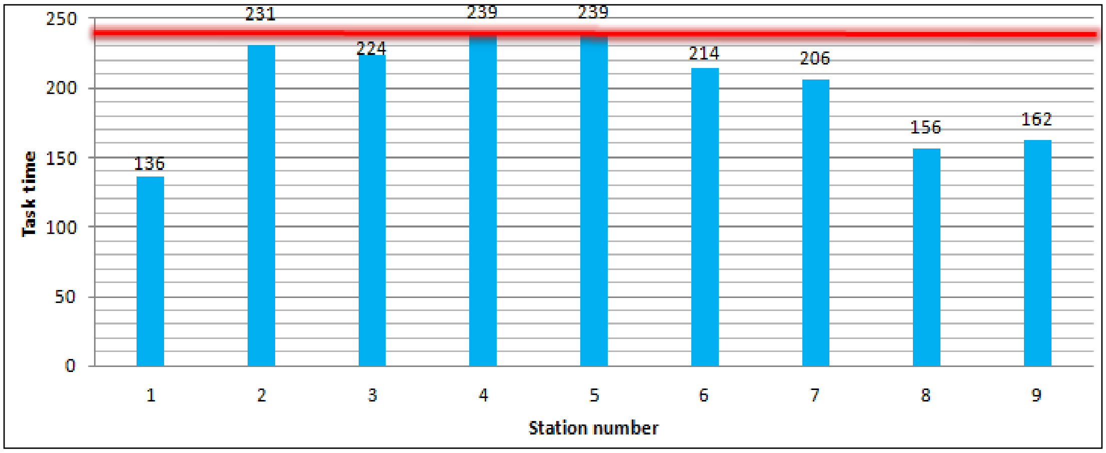


Figure 4.4 The line balance using group assignment procedure

The new layout according to the result of group assignment procedure is presented in Appendix B. Obviously, the new solution in Appendix B has the shorter length, and it is smoother than the current situation.

### *4.3.2 Teaching-learning Based Optimization Algorithm*

Teaching-Learning Based Optimization (TLBO) is a new method within well-known meta-heuristic methods such as Genetic Algorithm, Ant Colony Optimization, and Artificial Bee Colony. This method is based on philosophy of teaching and learning in a class. The basic philosophy of the method is based on the two factors: (i) influence of a teacher on the output of learners in terms of results or grades, and (ii) interaction between learners.

TLBO was proposed by Rao et al. (2011) for mechanical design problems. In the next study, Rao & Kalyankar (2011) addressed parameter optimization problem of advanced machining processes using TLBO algorithm. This method was tested and compared with the other well known meta-heuristics using non-linear optimization benchmark problems by Rao et al. (2012). In another study, Rao & Patel (2012) applied TLBO method to thermodynamic optimization of plate-fin heat exchanger. TLBO was also used on data clustering and discrete optimization in design of planar steel frames (Satapathy & Naik, 2011; Togan, 2012).

The TLBO method can be categorized into two parts as "*teacher phase*" and "*learner phase*".

*Teacher phase:*

In this phase, as shown in Figure 4.5, teacher tries to increase the mean of class from any value ($M_a$) to his/her level ($T_a$). But in real life, it is not completely possible and a teacher just can raise the mean of class from $M_a$ to $M_b$ which is higher than $M_a$. As in Gaussian law, there are a few students who can understand all knowledge acquired by the teacher (right side of Gaussian law). Some students can partially understand knowledge (mid part of Gaussian law) and the others' knowledge do not change with teacher's knowledge (left side of Gaussian law).

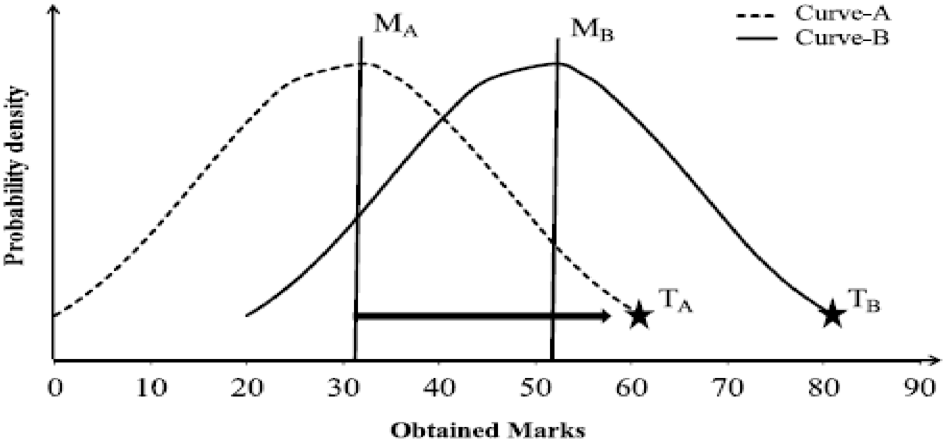


Figure 4.5 Model for obtained marks distribution for a group of learners (Rao et al., 2011)

In this aspect, teaching role is assigned to the person who has the best level in the class ($X_{teacher}$). The algorithm tries to increase other persons' level ($X_i$) to teacher's level considering the current mean ($X_{mean}$)

$X_{new} = X_i + r * (X_{teacher} - (T_f * X_{mean}))$

$T_f$ is the teaching factor which decides the value of mean to be changed, and $r_i$ is the random number in the range [0, 1]. Value of $T_f$ can be either 1 or 2.

*Learner phase:*

In this phase, $X_i$ tries to improve his/her knowledge by learning from $X_{ii}$ which is assumed to have better knowledge than $X_i$. In this case, $X_i$ is moved to $X_{ii}$. Otherwise, it means that $X_{ii}$ is not better than $X_i$, then $X_i$ is moved away from $X_{ii}$.

$X_{new} = X_i + r* (X_{ii} - X_i)$

$X_{new} = X_i + r* (X_i - X_{ii})$

The steps of the TLBO algorithm are given below:

1. Define the optimization problem and parameters (e.g., population size, number of generations, etc.)
2. Initialize population

   Generate random population according to the population size and the number of design ($P_n$ and $D$, respectively)

   $$\text{Population}= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ \vdots & & \ddots & \vdots \\ x_{Pn,1} & x_{Pn,2} & \cdots & x_{Pn,D} \end{bmatrix}$$

3. Teacher phase
4. Learner phase
5. Repeat procedure steps until the termination criteria is achieved.

Flow chart for TLBO is demonstrated in Figure 4.6.

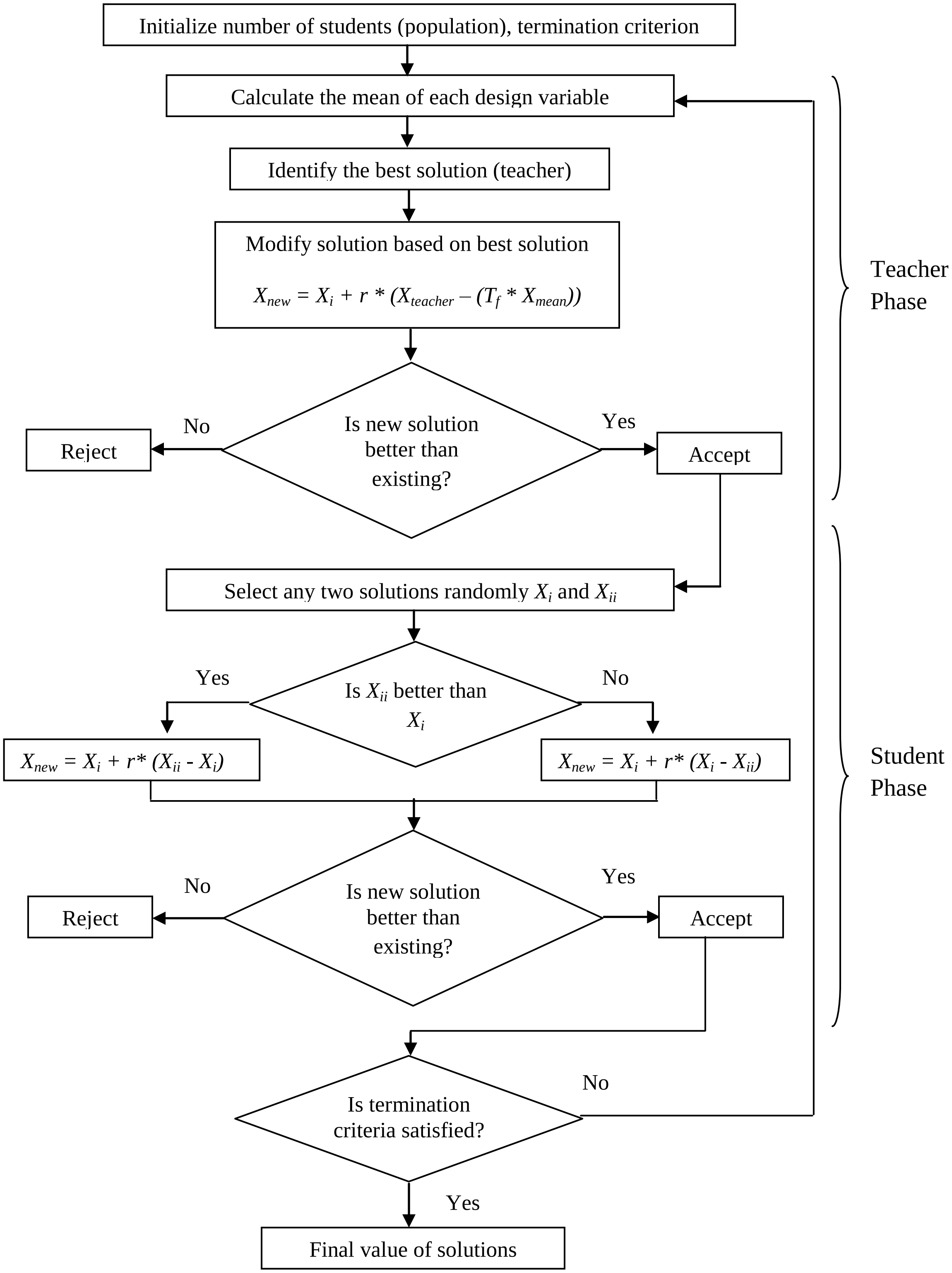


Figure 4.6 Flow chart of the TLBO algorithm (Rao et al., 2011)

In the application of this algorithm to line balancing problem, we used fitness function which is explained below. Our objective function is to minimize the number of workstations as a primary goal and to smooth the workload between workstations as a secondary goal given in Equation 4.3.

| | |
|---|---|
| $K$ | total number of workstations utilized on the line |
| $T$ | total idle time of all workstations |
| $C_b$ | smoothness index of the workload between workstations |
| $I_k$ | Idle time at workstation $k$ |
| $C$ | cycle time |
| $W_k$ | total workload assigned to workstation $k$ |

$$T = \sum_{k=1}^{K} I_k \qquad (4.1)$$

$$C_b = \sum_{k=1}^{K} [\ (I_k / T)\ - (1/K)\ ]^2 \qquad (4.2)$$

Total idle time of all workstations is calculated by Equation 4.1, and the smoothness index of the workload between workstations is computed according to Equation 4.2.

$$min\ z = K + C_b \qquad (4.3)$$

The TLBO algorithm is coded in Matlab 7.10.0. Population size, number of generations and number of replications are set to 40, 10, and 10, respectively. Using these parameters for 10 replications, we obtained the solutions given in Table 4.2.

The best result in the experiments is achieved on the first replication with 9 workstations and 0.131 of $C_b$. The line balance between workstations achieved by using TLBO algorithm is presented in Figure 4.7. The new configuration of the two-sided AL is illustrated in Appendix C.

Table 4.2 Implementation results obtained by TLBO algorithm

| **Replication no** | **Replication matrix** | **CPU-time-matrix** | **Min-value** | **Max-value** | **Mean value** |
|---|---|---|---|---|---|
| 1 | **9.131** | 20.6987 | **9.131** | 10.541 | 9.911 |
| 2 | 10.143 | 20.7376 | | | |
| 3 | 10.541 | 20.5989 | | | |
| 4 | 10.296 | 20.5804 | | | |
| 5 | 10.145 | 20.9565 | | | |
| 6 | 10.129 | 21.2433 | | | |
| 7 | 9.258 | 20.6703 | | | |
| 8 | 10.210 | 20.791 | | | |
| 9 | 9.216 | 20.7008 | | | |
| 10 | 10.340 | 20.6737 | | | |

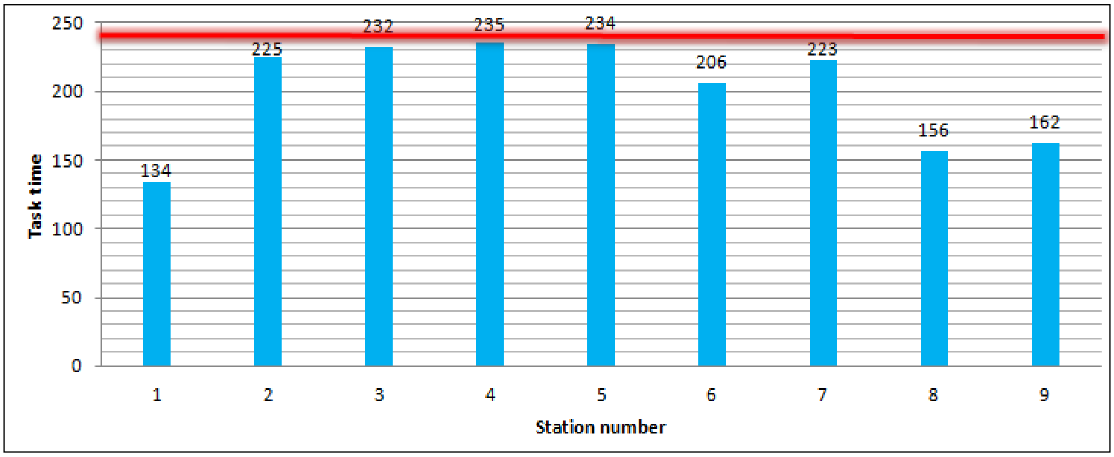


Figure 4.7 The line balance using teaching-learning based optimization algorithm

## 4.4 Computational Results

The solutions obtained by using the group assignment procedure and teaching-learning based optimization algorithm are compared with the current line balance in the production system in terms of the line efficiency, number of workers, and workloads between stations in Table 4.3.

Table 4.3 Summary of the computational results

| | **Lenght of the Line (number of workstations)** | **Number of Workers** | **Line Efficiency** | **Smoothing Index (Cb)** |
|---|---|---|---|---|
| **Initial Situation** | 9 | 12 | 63% | 0.20 |
| **Group Assignment Procedure** | 6 | 9 | 85% | 0.14 |
| **TLBO Algorithm** | 6 | 9 | 85% | 0.13 |

When we analyze the computational results, we can see that Group Assignment Procedure and TLBO algorithm provided a line balance with 9 workstations and less idle time at each workstation compared to the initial line configuration. Additionally, their smoothing indexes ($C_b$) are very close to each other; 0.14 and 0.13 respectively. We can conclude that, we have an improvement rate about 35% in terms of workloads between workstations. On the other hand, line efficiency increased from 63% to 85% with both of the solution approaches. According to theoretical efficiency, this value is ideally % 95. But in real-life, it can not be always possible due to the lots of practical constraints, e.g. resource, workzone, position, operator, ergonomic and internal logistics constraints. In this study, we considered practical extentions of two-sided assembly line balancing problem, and obtained high quality solutions in a very short computational time.

# CHAPTER FIVE
# CONCLUSION

In this thesis, we considered a real life two-sided ALBP with additional assignment restrictions. In this type of line balancing problem, there are three operation directions differently from the other problem types (e.g., one-sided or U-shaped lines). Some of the assembly operations should be performed at strictly one side of the line (i.e., right or left side), whereas the others can be performed at either side of the line. Therefore, we should also take into account operating sides of tasks in addition to precedence and cycle time constraints, when the allocation of the tasks to an ordered sequence of workstations is determined. Moreover, the problem considered in this thesis involves several compatible and incompatible zoning constraints. Accordingly, some groups of tasks must be executed together on the same station (compatible tasks) and other tasks were prevented from being assigned to the same station (incompatible tasks). In addition, one of the workstation has a different cycle time from the other stations because of machine capacity working in that station. Finally, each one of two tasks should be assigned to the different stations with no other tasks assigned (negative zoning constraint). Objective function was to minimize number of workstations and to ensure a smooth distribution of workload between workstations.

We used the group assignment procedure and teaching-learning based optimization algorithm to solve the line balancing problem under consideration. In both methods, current line composed of 12 workstations was balanced with 9 workstations in the new configurations with similar smoothness rates. We achieved 35% improvement in the distribution of workloads between workstations, and 25% reduction in total number of operators for pre-assembly line. However, 9 workstations for this industrial assembly system may be so strict when we consider real-life circumstances in practice. For instance, we assumed that there are no setup times in operations, operation times are deterministic, and there is no absence of the operator. Consequently, one worker, who assists to supplementary operations such as siliconizing, grouping, and seperating of parts, can be charged as a joker operator on-demand, if we consider the uncertainties and stochastic nature of real-life.

Two-sided assembly lines have more complicated structures than the other type of assembly lines. Additionally, we can face different conditions in real world such as symmetric tasks, synchronous tasks, task separation, station layout, parallel stations, setup times, multi or mixed model, capacity of machines in stations (local cycle times) and ergonomic constraints. Beyond benchmark/test problems, if real-world problems are addressed by including above conditions, gap between real world and academic world can be shortened.

Moreover, teaching-learning based optimization algorithm is a quite new method within the other meta-heuristics methods. In literature, we can see that this algorithm give promising results. As a further research area, its phases can be improved using different teachers in teacher phase and using tutorials or self-learning capability in learner phase.

# REFERENCES

Agpak, K. K., & Gökçen, H. (2005). Assembly line balancing: Two resource constraint cases. *International Journal of Production Economics*, *96*, 129-140.

Ajenblit, D.A., & Wainwright, R.L. (1998). Applying genetic algorithms to the u-shaped assembly line balancing problem. *In the Proceeding of the 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, 96-101.

Akpınar, S., & Bayhan, M. (2011). A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. *Engineering Applications of Artificial Intelligence*, *24*, 449–457.

Bartholdi, J.J. (1993). Balancing two-sided assembly lines: a case study. *International Journal of Production Research*, *31*, 2447–2461.

Bautista, J., Suarez, R., Mateo, M., & Companys, R. (2000). Local search heuristics for the assembly line balancing problem with incompatibilities between tasks. *In the Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2404-2409.

Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, *177*, 2016-2032.

Bautista, J., & Pereira, J. (2009). A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research*, *194*, 787-794.

Baykasoglu, A., & Dereli, T. (2009). Simple and U-type assembly line balancing by using an ant colony based algorithm. *Mathematical and Computational Applications, 14,* 1-12.

Becker, C., & Scholl, A. (2009). Balancing assembly lines with variable parallel workplaces: problem definition and effective solution procedure. *European Journal of Operational Research, 199,* 359-374.

Blum, C., & Miralles, C. (2010). On solving the assembly line worker assignment and balancing problem via beam search. *Computers & Operations Research, 38,* 328-339.

Boysen N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics, 111*, 509-528.

Brown, E. C., & Sumichrast, R. T. (2005). Evaluating performance advantages of grouping genetic algorithms. *Engineering Applications of Artificial Intelligence, 18*, 1-12.

Brudaru, O., & Valmar, B. (2004). Genetic algorithm with embryonic chromosomes for assembly line balancing with fuzzy processing times. *The 8th International Research/Expert Conference Trends in the Development of Machinery and Associated Technology*, TMT 2004, Neum, Bosnia and Herzegovina.

Bukchin, Y., & Rabinowitch, I. (2006). A branch and bound based solution approach for the mixed model assembly line balancing problem for minimizing stations and task duplication costs. *European Journal of Operational Research, 174,* 492-508.

Cakir, B., Altıparmak, F., & Dengiz, B. (2011). Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. *Computers & Industrial Engineering, 60*, 376–384.

Carnahan, B. J., Norman, B. A., & Redfern, M. S. (2001). Incorporating physical demand criteria into assembly line balancing. *IIE Transactions, 33,* 875-887.

Chan, C. C. K., Hui, P. C. L., Yeung, K. W., & Ng F. S. F. (1998). Handling the assembly line balancing problem in the clothing industry using a genetic algorithm. *International Journal of Clothing Science and Technology, 10*(1), 21- 37.

Chen, R. S., Lu, K. Y., & Yu, S. C. (2002). A hybrid genetic algorithm approach on multi-objective of assembly planning problem. *Engineering Applications of Artificial Intelligence, 15*, 447–457.

Chen, J.C., Chen, C.C., Su, L.H., Wu, H.B., & Sun, C.J. (2012). Assembly line balancing in garment industry. *Expert Systems with Applications, 39*, 10073–10081.

Choi, G. (2009). A goal programming mixed model line balancing for processing time and physical workload. *Computers and Industrial Engineering, 57*, 395-400.

Chutima, P., & Chimklai, P. (2011). Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. *Computers & Industrial Engineering, 62,* 39–55.

Corominas, A., Pastor, R., & Plans, J. (2008). Balancing assembly lines with skilled and unskilled workers. Omega, *International Journal of Management Science, 36*, 1126-1132.

Ege, Y., Azizoglu, M., & Ozdemirel, N.E. (2009). Assembly line balancing with station paralling. *Computers & Industrial Engineering, 57*, 1218-1225.

Erel, E., & Gökçen, H. (1999). Shortest-route formulation of mixed model assembly line balancing problem. *European Journal of Operational Research, 116*, 194-204.

Frey, G. (2000). Assembly line sequencing based on petri net t-invariants. *Control Engineering Practice, 8*, 63-69.

Gao, J., Sun, L., Wang, L., & Gen, M. (2009). An efficient approach for type-2 roboitc assembly line balancing problems. *Computers and Industrial Engineering, 56*, 1065-1080.

Gen, M., Cheng, R., & Lin, L. (2008). *Network Models and Optimization: Multiobjective Genetic Algorithm Approach.* Decision Engineering Series, Springer-Verlag, London, 477-550.

Gökçen, H., & Erel, E. (1997). A goal programming approach to mixed-model assembly line balancing problem. *International Journal of Production Economics, 48*, 177-185.

Gökçen, H., & Erel, E. (1998). Binary integer formulation for mixed model assembly line balancing problem. *Computers & Industrial Engineering, 34*, 451-461.

Gökçen, H., & Agpak, K. (2002). Basit U tipi montaj hattı dengeleme problemine bulanık programlama yaklaşımı. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 4*, 2, 29-40.

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

Goncalves, J. F., & De Almedia, J. R. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristic, 8,* 629-642.

Groover, M. P. (2001). *Automation, Production Systems, and Computer-Integrated Manufacturing.* 2nd Edition, Upper Saddle River, NJ: Prentice Hall.

Gunasekaran, A., & Sandhu, M. (2010). *Handbook on Business Information Systems.* World Scientific Publishing, Singapure.

Hamzadayi, A., & Yildiz, G. (2012). A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints. *Computers & Industrial Engineering, 62,* 206–215.

Hyun, C.J., Kim, Y., & Kim, Y.K.(1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers and Operations Research, 25,* 675-690.

Ji, P., Sze, M.T., & Lee, W.B. (2001). A genetic algorithm of determining cycle time for printed circuit board assembly lines. *European Journal of Operational Research, 128*, 175−184.

Kabir, A., & Tabucanon, M.T. (1995). Batch-model assembly line balancing: A multiattribute decision making approach. *International Journal of Production Economics, 41*, 193-201.

Kara, Y., Gökçen, H., & Atasagun, Y. (2010). Balancing parallel assembly lines with precise and fuzzy goals. *International Journal of Production Research, 48,* 6, 1685-1703.

Kılınççı, Ö. (2011). Firing sequences backward algorithm for simple assembly line balancing problem of type 1. *Computers & Industrial Engineering, 60,* 830–839.

Kim, Y.J., Kim, Y.K., & Cho, Y. (1998). A heuristic based genetic algorithm for workload smoothing in assembly lines. *Computers & Operations Research, 25,* 99-111.

Kim, Y.K., Kim, Y.J., & Kim, Y. (1996). Genetic algorithms for assembly line balancing with various objectives. *Computers & Industrial Engineering, 30*, 397-409.

Kim, Y. K., Kim, Y., & Kim, Y. J. (2000). Two-sided assembly line balancing: A genetic algorithm approach. *Production Planning and Control, 11*(1), 44-53.

Kim, Y.K., Song, W.S., & Kim, J.H. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers & Operations Research, 36*, 853-865.

Klein, R., & Scholl, A. (1996). Maximizing the production rate in simple assembly line balancing - A branch and bound procedure. *European Journal of Operational Research, 91*, 367-385.

Lapierre, S. D., Ruiz, A., & Soriano, P. (2006). Balancing assembly lines with tabu search. *European Journal of Operational Research, 168*, 826–837.

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

Leu, Y.Y., Matheson, L.A., & Rees, L.P. (1996). Sequencing mixed-model assembly lines with genetic algorithms. *Computers & Industrial Engineering, 30*, 1027-1036.

Levitin, G., Rubinovitz, J., & Shnits, B. (2006). A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research, 168*, 811-825.

Martinez, U., & Duff, W. S. (2004). Heuristic approaches to solve the U-shaped line balancing problem augmented by Genetic Algorithms. *In the Proceedings of the 2004 Systems and Information Engineering Design Symposium*, 287-293.

Mendes, A.R., Ramos, A.L., Simaria, A.S., & Vilarinho, P.M. (2005). Combining heuristic procedures and simulation models for balancing a pc camera assembly line. *Computers and Industrial Engineering*, *49*, 413-431.

McMullen, P. R., & Tarasewich, P. (2003). Using ant techniques to solve the assembly line balancing problem. *IIE Transactions: Design and Manufacturing, 35*(7), 605-617.

Nearchou, A.C (2011). Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *Int. J. Production Economics, 129*, 242–250.

Nicosia, G., Pacciarelli, D., & Pacifici, A. (2002). Optimally balancing assembly lines with different workstations. *Discrete Applied Mathematics, 118*, 99-113.

Ozbakır, L., & Tapkan, P. (2011). Bee colony intelligence in zone constrained two-sided assembly line balancing problem. *Expert Systems with Applications, 38*, 11947–11957.

Ozbakır,L., Baykasoglu, A., Gorkemli, B., & Gorkemli, L.(2011). Multiple-colony ant algorithm for parallel assembly line balancing problem. *Applied Soft Computing, 11,* 3186–3198.

Ozcan, U., & Toklu, U. (2009). Multiple-criteria decision making in two sided assembly line balancing: a goal programming and a fuzzy goal programming models. *Computers & Operations Research*, *36*, 1955-1965.

Ozcan, U. (2010). Balancing stochastic two-sided assembly lines: a chance constrained, piecewise linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research, 205*, 81-97.

Ozcan, U., Cercioglu, H., Gökçen, H., & Toklu, B. (2009). A Tabu Search Algorithm for the Parallel Assembly Line Balancing Problem. *Gazi University Journal of Science, 22*, 313-323.

Ozcan, U., & Toklu, B. (2009). A tabu search algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, *43*, 822-829.

Ozgormus, E., (2007). Ergonomik Koşullar Altında Montaj Hattı Dengeleme. Postgraduate thesis, Pamukkale Üniversitesi, Denizli, Turkey.

Ozmehmet Tasan, S. (2007). Solving Simple And Mixed-Model Assembly Line Balancing Problems Using Hybrid Meta-Heuristic Approaches. Ph.D. Thesis, Dokuz Eylül University, Izmir, Turkey.

Pastor, R, Andris, C., Duran, A., & Pirez, M. (2002). Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of the Operational Research Society, 53*(12), 1317-1323.

Ponnambalam, S. G., Aravindan, P., Naidu, G., & Mogileeswar, G. (2000). Multiobjective genetic algorithm for solving assembly line balancing problem. *International Journal of Advanced Manufacturing Technology, 16*(5), 341-352.

Rahimi-Vahed, A.R., Rabbani, M., Tavakkoli-Moghaddam R., Torabi, S.A., & Jolai, F. (2007). A multi-objective scatter search for a mixed model assembly line sequencing problem. *Advanced Engineering Informatics*, *21*, 85-99.

Rekiek, B., & Delchambre, A. (2006). *Assembly Line Design: The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms*. Springer Series in Advanced Manufacturing, Springer London, 13-27.

Rao, R.V.,Savsani, V.J., & Vakharia, D.P. (2011). Teaching-learning based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*, 303-315.

Rao, R.V., & Kalyankar, V.D. (2011). Parameters optimization of advanced machining processes using TLBO algorithm. *EPPM: International Conference on Engineering, Project, and Production Managament*, September 20-21, 2011, Singapore.

Rao, R.V.,Savsani, V.J., & Vakharia, D.P. (2012). Teaching-learning based optimization: An optimization method for continuous non-linear large scale problems. *Information Sciences, 183*, 1-15.

Rao, R.V., & Patel, V.(2012). Multi-objective optimization of combined Brayton and inverse Brayton cycles using advanced optimization algorithms. *Engineering Optimization. 44*(8), 965-983.

Rubinovitz, J. & Levitin, G. (1995). Genetic algorithm for assembly line balancing. *International Journal of Production Economics, 41*, 343-354.

Sabuncuoglu, I., Erel, E., & Tanyer, M. (2000). Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing, 11*(3), 295-310.

Sarin, S.C., Erel, E., & Dar-El, E.M.(1999). A methodology for solving single model, stochastic assembly line balancing problem. Omega, *International Journal of Management Science, 27*, 525-535.

Sarker, B.R., & Pan, H. (1998). Designing a mixed model assembly line to minimize the costs of idle and utility times. *Computers & Industrial Engineering, 34*, 609-628.

Satapathy, S. C., & Naik, A. (2011). Data clustering based on teaching-learning-based optimization. In: B.K. Panigrahi et al. (Eds.), SEMCCO 2011, Part II, LNCS 7077, 148-156.

Sawik, T. (1995). Integer programming models for the design and balancing of flexible assembly systems. *Mathematical and Computer Modelling, 21*, 1-12.

Sawik, T. (2000). An LP-based approach for loading and routing in a flexible assembly line. *International Journal of Production Economics, 64,* 49-58.

Scholl, A., & Boysen, N. (2008). The multiproduct parallel assembly lines balancing problem: model and optimization procedure. *Working and Discussion Paper Series School of Economics and Business Administration Friedrich-Schiller-University Jena,* ISSN 1864-3108.

Scholl, A., & Boysen, N. (2009). Designing parallel assembly lines with split workplaces: Model and optimization procedure. *International Journal of Production Economics, 119,* 90–100.

Scholl, A., Fliedner, M., & Boysen, N. (2010). ABSALOM: Balancing assembly lines with assignment restrictions. *European Journal of Operational Research, 200,* 688–701.

Scholl, A. (1999). *Balancing and sequencing of assembly lines* (2nd Ed.). New York: Springer-Verlag.

Scholl, A., & Klein, R. (1999). Balancing assembly lines effectively - a computational comparison. *European Journal of Operational Research, 114,* 50-58.

Simaria, A. S., & Vilarinho, P. M. (2001). The simple assembly line balancing problem with parallel workstations- a simulated annealing approach. *International Journal of Industrial Engineering, 8*(3), 230-240.

Simaria, A. S., & Vilarinho, P. M. (2004). A genetic algorithm based approach to mixed model assembly line balancing problem of type II. *Computers and Industrial Engineering, 47,* 391-407.

Simaria, A. S., & Vilarinho, P. M. (2009). 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. *Computers and Industrial Engineering, 56 (2),* 489-506.

Sly, D.P (2007). *A Practical Approach to solving Multi-objective Line Balancing Problem*. Retrieved May 12, 2010, from http://www.proplanner.com/documents/filelibrary/products/probalance/documents/Line_Balance_PDF_EA88A25C4FF47.pdf.

Stockton, D. J., Quinn, L., & Khalil, R. A. (2004a). Use of genetic algorithms in operations management Part 1: applications. Proceeding of the Institution of Mechanical Engineers-Part B: *Journal of Engineering Manufacture, 218*(3), 315-327.

Stockton, D. J., Quinn, L., & Khalil, R. A. (2004b). Use of genetic algorithms in operations management Part 2: results. *Proceeding of the Institution of Mechanical Engineers-Part B: Journal of Engineering Manufacture, 218*(3), 329-343.

Suresh, G., Vinod V.V., & Sahu S. (1996). A genetic algorithm for assembly line balancing. *Production Planning & Control, 7,* 38-46.

Tamura, T., Long, H., & Ohno, K. (1999). A sequencing problem to level part usage rates and work loads for a mixed model assembly line with a bypass subline. *International Journal of Production Economics, 60-61*, 557-564.

Tapkan, P, Ozbakır, L., & Baykasoglu, A. (2012). Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms. *Applied Soft Computing, Volume 12, Issue 11,* 3343-3355.

Togan, V. (2012). Design of planar steel frames using teaching–learning based optimization, *Engineering Structures, 34*, 225–232.

Toklu, B., & Ozcan, U. (2008). A fuzzy goal programming model for the simple U-line balancing problem with multiple objecitves. *Engineering Optimization, 40*, 3, 191-204.

Toksarı, M.D., Isleyen, S.K., Güner, E., & Baykoc, O.F. (2010). Assembly line balancing problem with deterioration tasks and learning effect. *Expert Systems with Applications, 37*, 1223-1228.

Tsujimura, Y., Gen, M., & Li, Y. (1995). Fuzzy assembly line balancing using genetic algorithms. *Computers & Industrial Engineering, 31*, 631-634.

Ugurdag, H.F., Rachamadugu, R., & Papachristou, C.A. (1997). Designing paced assembly lines with fixed number of stations. *European Journal of Operational Research, 102*, 488-501.

Valente, S. A., Lopes, H. S., & Arruda, L. V. R. (2002). *Genetic algorithms for the assembly line balancing problem: a real-world automotive application.* In: Roy, R., Köppen, M., Ovaska, S., Fukuhashi, T., Hoffman, F. Soft Computing in Industry - Recent Applications. Berlin: Springer-Verlag, 319-328.

Wu, E.F., Jin, Y., Bao, J.S., & Hu, X.F. (2008). A branch and bound algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology, 39*, 1009-1015.

Yagmahan, B. (2011). Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. *Expert Systems with Applications*, *38*, 12453–12461.

Zacharia, P.T., & Nearchou, A.C. (2012). Multi-objective fuzzy assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing, 23*, 615-627.

Zhang, Z., & Cheng, W. (2010). Solving fuzzy U-shaped line balancing problem with exact method. *Applied Mechanics and Materials, 26-28*, 1046-1051.

# APPENDICES

# Appendix- A: Solution Steps of Group Assignment Procedure

for **j= 1, S1=0**

| Iteration | Gk | Dk | Tk | Direction | Gr | U | Note |
|---|---|---|---|---|---|---|---|
| **1** | G1=1,2,3,4 | 0 | 135 | E | Gr=G2 (AR3) | G2 is assigned to S1<br>U=U-{1,2,6,7} | due to machine constraint, task time of this sta has to be equal or less than 137 and it has to be beginning of the line differently from L and R station |
| | **G2=1,2,6,7** | 0 | **136** | **E** | | | |
| | G3=1,2,3,7 | 0 | 133 | E | | | |
| | G4=1,2,3,6 | 0 | 135 | E | | | |
| | G5=47,48,49,50 | 0 | 89 | L | | | |

for **j= 2, j'=3, S2=0, S3=0**

| Iteration | Gk | Dk | Tk | Direction | Gr | U | Note |
|---|---|---|---|---|---|---|---|
| **2** | G1=3,4,5,8,9 | 136 | 84 | E | Gr=G3 (AR1-2) | G3 is assigned to S3<br>U=U-{47,48,49,50} | - |
| | G2=3,4,5,8,10 | 136 | 109 | R | | | |
| | **G3=47,48,49,50** | 0 | **89** | **L** | | | |
| **3** | G1=3,4,5,8,9 | 136 | 84 | E | Gr=G2 (AR3) | G2 is assigned to S2<br>U=U-{3,4,5,8,10} | - |
| | **G2=3,4,5,8,10** | 136 | **109** | **R** | | | |
| **4** | G1=9,11,12,13,14,15,16,17 | 207 | 135 | L | Gr=G2 (AR2) | G2 is assigned to S3<br>U=U-{51,52} | - |
| | **G2=51,52** | 80 | **122** | **R** | | | |
| **5** | G1=9,11,12,13,14,15,16,17 | 207 | 135 | L | Gr=G1 | G1 is assigned to S2<br>U=U-{9,11,12,13,14,15,16,17} | - |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 4, j'=5, S4=0, S5=0**

| Iteration | Gk | Dk | Tk | Direction | Gr | U | Note |
|---|---|---|---|---|---|---|---|
| **6** | **G1=18,19,21** | 0 | **103** | **R** | Gr=G1 (AR3) | G1 is assigned to S5<br>U=U-{18,19,21} | - |
| | G2=23 | 0 | 29 | E | | | |
| **7** | G1=20,27 | 37 | 59 | L | Gr=G2 (AR2) | G2 is assigned to S4<br>U=U-{23,24} | - |
| | **G2=23,24** | 0 | **76** | **L** | | | |
| **8** | **G1=20,27** | 0 | **59** | **L** | Gr=G1 (AR3) | G1 is assigned to S4<br>U=U-{20,27} | - |
| | G2=25,26 | 0 | 16 | R | | | |
| **9** | G1=22,25,26 | 0 | 32 | R | Gr=G1 | G1 is assigned to S5<br>U=U-{22,25,26} | - |
| **10** | G1=28 | 0 | 10 | R | Gr=G2 (AR3) | G2 is assigned to S5<br>U=U-{29,32} | - |
| | **G2=29,32** | 0 | **55** | **R** | | | |
| | G3=30 | 0 | 22 | R | | | |
| | G4=31 | 0 | 9 | R | | | |
| | G5=33 | 0 | 8 | L | | | |
| | G6=34 | 0 | 11 | E | | | |
| | G7=35 | 0 | 18 | R | | | |
| | G8=36 | 0 | 31 | R | | | |
| | G9=38 | 0 | 10 | L | | | |
| | G10=39 | 0 | 15 | L | | | |
| | G11=40 | 0 | 29 | L | | | |
| | G12=41 | 0 | 34 | E | | | |
| | G13=42 | 0 | 26 | L | | | |
| | G14=43 | 0 | 16 | L | | | |
| | G15=44 | 0 | 12 | L | | | |
| | G16=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

| for **j= 4, j'=5, S4=0, S5=0** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
| **11** | G1=28 | 0 | 10 | R | Gr=G11 (AR3) | G11 is assigned to S4<br>U=U-{41} | Total operation time of unassigned L tasks is 311, that of R tasks is 349, so the operation direction of G11 is L. |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=35 | 0 | 18 | R | | | |
| | G7=36 | 0 | 31 | R | | | |
| | G8=38 | 0 | 10 | L | | | |
| | G9=39 | 0 | 15 | L | | | |
| | G10=40 | 0 | 29 | L | | | |
| | **G11=41** | **0** | **34** | **E** | | | |
| | G12=42 | 0 | 26 | L | | | |
| | G13=43 | 0 | 16 | L | | | |
| | G14=44 | 0 | 12 | L | | | |
| | G15=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 4, j'=5, S4=0, S5=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **12** | G1=28 | 0 | 10 | R | Gr=G7 (AR3) | G7 is assigned to S5<br>U=U-{36} | - |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=35 | 0 | 18 | R | | | |
| | **G7=36** | **0** | **31** | **R** | | | |
| | G8=38 | 0 | 10 | L | | | |
| | G9=39 | 0 | 15 | L | | | |
| | G10=40 | 0 | 29 | L | | | |
| | G11=42 | 0 | 26 | L | | | |
| | G12=43 | 0 | 16 | L | | | |
| | G13=44 | 0 | 12 | L | | | |
| | G14=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

| for **j= 4, j'=5, S4=0, S5=0** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
| **13** | G1=28 | 0 | 10 | R | Gr=G9 (AR3) | G9 is assigned to S4<br>U=U-{40} | - |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=35 | 0 | 18 | R | | | |
| | G7=38 | 0 | 10 | L | | | |
| | G8=39 | 0 | 15 | L | | | |
| | **G9=40** | **0** | **29** | **L** | | | |
| | G10=42 | 0 | 26 | L | | | |
| | G11=43 | 0 | 16 | L | | | |
| | G12=44 | 0 | 12 | L | | | |
| | G13=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 4, j'=5, S4=0, S5=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **14** | G1=28 | 0 | 10 | R | Gr=G10 (AR3) | G10 is assigned to S4<br>U=U-{42} | - |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=35 | 0 | 18 | R | | | |
| | G7=38 | 0 | 10 | L | | | |
| | G8=39 | 0 | 15 | L | | | |
| | **G10=42** | **0** | **26** | **L** | | | |
| | G11=43 | 0 | 16 | L | | | |
| | G12=44 | 0 | 12 | L | | | |
| | G13=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 4, j'=5, S4=0, S5=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **15** | G1=28 | 0 | 10 | R | Gr=G6(AR3) | G6 is assigned to S5<br>U=U-{35} | G2 has max operation time but if we select G2 to set of Gr, total task time of S5 is exceeded cycle time so we select G6 |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | **G6=35** | **0** | **18** | **R** | | | |
| | G7=38 | 0 | 10 | L | | | |
| | G8=39 | 0 | 15 | L | | | |
| | G11=43 | 0 | 16 | L | | | |
| | G12=44 | 0 | 12 | L | | | |
| | G13=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 4, j'=5, S4=0, S5=0**

| Iteration | Gk | Dk | Tk | Direction | Gr | U | Note |
|---|---|---|---|---|---|---|---|
| **16** | G1=28 | 0 | 10 | R | Gr=G7(AR3) | G7 is assigned to S5<br>U=U-{39} | G2 has max oper.time but its direction is R and in iteration 15, S5 is fulled.<br>Then, G8 has max operation time for S4 but if we select G8 to set of Gr, total task time of S4 is exceeded cycle time so we select G7 |
| | G2=30 | 0 | 22 | R | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=38 | 0 | 10 | L | | | |
| | **G7=39** | **0** | **15** | **L** | | | |
| | G8=43 | 0 | 16 | L | | | |
| | G9=44 | 0 | 12 | L | | | |
| | G10=45 | 0 | 13 | R | | | |

for **j= 6, j'=7, S6=0, S7=0**

| Iteration | Gk | Dk | Tk | Direction | Gr | U | Note |
|---|---|---|---|---|---|---|---|
| **17** | G1=28 | 0 | 10 | R | Gr=G2(AR3) | G2 is assigned to S7<br>U=U-{39} | - |
| | **G2=30** | 0 | **22** | **R** | | | |
| | G3=31 | 0 | 9 | R | | | |
| | G4=33 | 0 | 8 | L | | | |
| | G5=34 | 0 | 11 | E | | | |
| | G6=38 | 0 | 10 | L | | | |
| | G7=43 | 0 | 16 | L | | | |
| | G8=44 | 0 | 12 | L | | | |
| | G9=45 | 0 | 13 | R | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 6, j'=7, S6=0, S7=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **18** | G1=28 | 0 | 10 | R | Gr=G6(AR3) | G6 is assigned to S6<br>U=U-{43} | **-** |
| | G2=31 | 0 | 9 | R | | | |
| | G3=33 | 0 | 8 | L | | | |
| | G4=34 | 0 | 11 | E | | | |
| | G5=38 | 0 | 10 | L | | | |
| | **G6=43** | **0** | **16** | **L** | | | |
| | G7=44 | 0 | 12 | L | | | |
| | G8=45 | 0 | 13 | R | | | |
| **19** | G1=28 | 0 | 10 | R | Gr=G7(AR3) | G7 is assigned to S7<br>U=U-{45} | **-** |
| | G2=31 | 0 | 9 | R | | | |
| | G3=33 | 0 | 8 | L | | | |
| | G4=34 | 0 | 11 | E | | | |
| | G5=38 | 0 | 10 | L | | | |
| | G6=44 | 0 | 12 | L | | | |
| | **G7=45** | **0** | **13** | **R** | | | |
| **20** | G1=28 | 0 | 10 | R | Gr=G6(AR3) | G6 is assigned to S6<br>U=U-{44} | **-** |
| | G2=31 | 0 | 9 | R | | | |
| | G3=33 | 0 | 8 | L | | | |
| | G4=34 | 0 | 11 | E | | | |
| | G5=38 | 0 | 10 | L | | | |
| | **G6=44** | **0** | **12** | **L** | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 6, j'=7, S6=0, S7=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **21** | G1=28 | 0 | 10 | R | Gr=G4(AR3) | G4 is assigned to S6<br>U=U-{34} | Total operation time of unassigned L tasks is 133, that of R tasks is 156, so the operation direction of G4 is L. |
| | G2=31 | 0 | 9 | R | | | |
| | G3=33 | 0 | 8 | L | | | |
| | **G4=34** | 0 | **11** | **E** | | | |
| | G5=38 | 0 | 10 | L | | | |
| **22** | **G1=28** | 0 | **10** | **R** | Gr=G1(AR1) | G4 is assigned to S7<br>U=U-{28} | - |
| | G2=31 | 0 | 9 | R | | | |
| | G3=33 | 0 | 8 | L | | | |
| | G4=38 | 0 | 10 | L | | | |
| **23** | G1=31 | 0 | 9 | R | Gr=G3(AR3) | G3 is assigned to S6<br>U=U-{38} | - |
| | G2=33 | 0 | 8 | L | | | |
| | **G3=38** | 0 | **10** | **L** | | | |
| **24** | **G1=31** | 0 | **9** | **R** | Gr=G1(AR3) | G1 is assigned to S7<br>U=U-{31} | - |
| | G2=33 | 0 | 8 | L | | | |
| **25** | G1=33 | 0 | 8 | L | Gr=G1 | G1 is assigned to S6<br>U=U-{33} | - |
| **26** | G1=53,55,56,57 | 3 | 61 | R | Gr=G2(AR2) | G2 is assigned to S6<br>U=U-{54,58,59,60} | - |
| | G2=54,58,59,60 | 0 | 62 | L | | | |

Appendix- A (cont): Solution Steps of Group Assignment Procedure

for **j= 6, j'=7, S6=0, S7=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **27** | **G1=53,55,56,57** | **3** | **61** | **R** | Gr=G1 | G1 is assigned to S7<br>U=U-{53,55,56,57} | **-** |
| **28** | G1=61,62,63,64 | 1 | 81 | R | Gr=G2(AR2) | G2 is assigned to S6<br>U=U-{65,66,67,68} | **-** |
| | **G2=65,66,67,68** | **0** | **85** | **L** | | | |
| **29** | **G1=61,62,63,64** | **1** | **81** | **R** | Gr=G1 | G1 is assigned to S7<br>U=U-{61,62,63,64} | **-** |

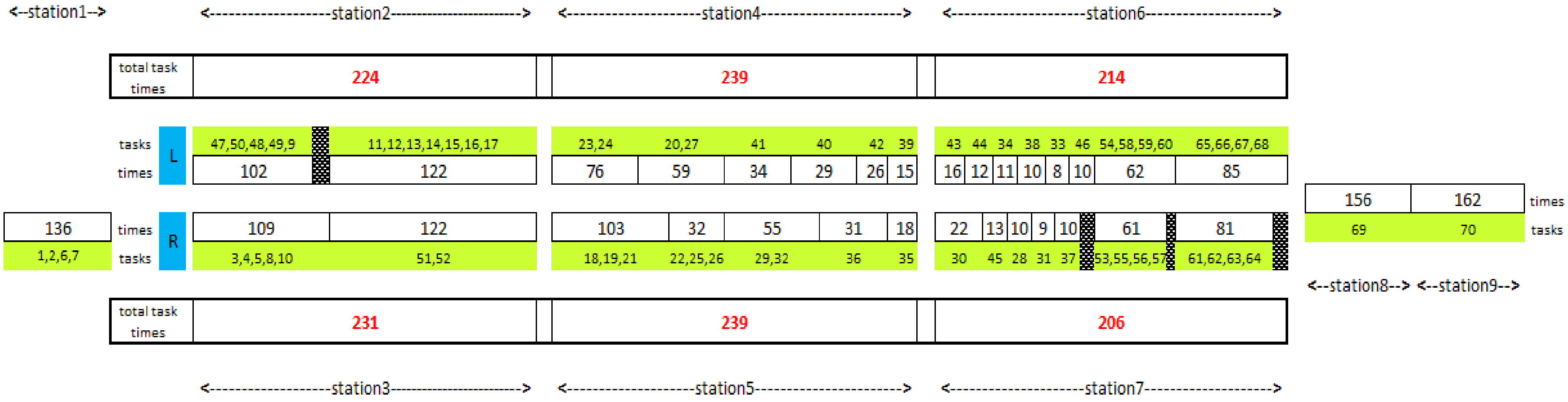
for **j= 8, S8=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **30** | G1=69 | 0 | 156 | E | Gr=G1 | G1 is assigned to S8<br>U=U-{69} | Task 69 is performed in a station alone due to machine constraint. |

for **j= 9, S9=0**

| **Iteration** | **Gk** | **Dk** | **Tk** | **Direction** | **Gr** | **U** | **Note** |
|---|---|---|---|---|---|---|---|
| **31** | G1=70 | 0 | 162 | E | Gr=G1 | G1 is assigned to S9<br>U=U-{70} | Task 70 is performed in a station alone due to quality constraint |

Appendix-B: New Layout According to the Result of Group Assignment Procedure



<--station1-->

| | |
|---|---|
| 136 | times |
| 1,2,6,7 | tasks |

<-------------------station2------------------------->

| | | |
|---|---|---|
| total task times | 224 | |
| tasks | 47,50,48,49,9 | 11,12,13,14,15,16,17 |
| times | 102 | 122 |

<-------------------station3------------------------->

| | | |
|---|---|---|
| times | 109 | 122 |
| tasks | 3,4,5,8,10 | 51,52 |
| total task times | 231 | |

L R

<--------------------station4---------------------->

| | | | | | | |
|---|---|---|---|---|---|---|
| total task times | 239 | | | | | |
| tasks | 23,24 | 20,27 | 41 | 40 | 42 | 39 |
| times | 76 | 59 | 34 | 29 | 26 | 15 |

<-------------------station5---------------------->

| | | | | | |
|---|---|---|---|---|---|
| times | 103 | 32 | 55 | 31 | 18 |
| tasks | 18,19,21 | 22,25,26 | 29,32 | 36 | 35 |
| total task times | 239 | | | | |

<--------------------station6-------------------->

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| total task times | 214 | | | | | | | |
| tasks | 43 | 44 | 34 | 38 | 33 | 46 | 54,58,59,60 | 65,66,67,68 |
| times | 16 | 12 | 11 | 10 | 8 | 10 | 62 | 85 |

<--------------------station7-------------------->

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| times | 22 | 13 | 10 | 9 | 10 | 61 | 81 |
| tasks | 30 | 45 | 28 | 31 | 37 | 53,55,56,57 | 61,62,63,64 |
| total task times | 206 | | | | | | |

<--station8--> <--station9-->

| | | |
|---|---|---|
| 156 | 162 | times |
| 69 | 70 | tasks |

# Appendix-C: New Layout According to the Result of Teaching-Learning Based Optimization Algorithm