ÇANKAYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
INDUSTRIAL ENGINEERING

MASTER THESIS

BALANCING AND SEQUENCING MULTI ZONE MIX MODEL ASSEMBLY LINES

NİZAMETTİN DOĞAN GÜNER

SEPTEMBER 2013

Title of the Thesis: **Balancing and Sequencing Multi-Zone Mix Model Assembly Lines**

Submitted by: **Nizamettin Doğan GÜNER**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Levent KANDİLLER

Assist. Prof. Dr. Benhür SATIR

Examination Date: **20.09.2013**

Examining Committee Members

Assist. Prof. Dr. Nureddin KIRKAVAK	(Çankaya University)	
Assist. Prof. Dr. Benhür SATIR	(Çankaya University)	
Prof. Dr. Levent KANDİLLER	(Yaşar University)	
Assist. Prof. Dr. Haluk AYGÜNEŞ	(Çankaya University)	
Assist. Prof. Dr. Alp ERTEM	(Çankaya University)	

STATEMENT OF NONPLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Nizamettin Doğan GÜNER

Signature :

Date : 20.09.2013

ABSTRACT

BALANCING AND SEQUENCING
MULTI-ZONE MIX MODEL ASSEMBLY LINES

GÜNER, Nizamettin Doğan

M.Sc, Department of Industrial Engineering
Supervisor: Asst. Prof. Benhür SATIR
Co-Supervisor: Prof. Dr. Levent KANDİLLER

September 2013, 83 pages

Assembly based production has importance for large scale industries of developing countries. Satisfying demand and meeting production due dates of those kind of products are the prime targets of companies. One of the largest bus production facilities of Turkey, MAN Türkiye A.Ş., aims to minimize delivery costs and to meet daily production amounts by preventing delays in assembly operations. In this study, an assembly line design is suggested in order to achieve the company goals by studying single model line balancing and mixed model sequencing problems. Based on the characteristics of real life problem, three mathematical models for line balancing and sequencing are constructed. Furthermore, a heuristic algorithm is proposed for single model line balancing problem. This study is a part of a project[1] that is partially supported by SANTEZ Program of Ministry of Science, Industry and Technology of Republic of Turkey.

Keywords: Mixed model assembly lines, line balancing, model sequencing

---

[1] Project name is “Decision Support System ofMulti Worker Multi Sided Mixed Model Assembly Line Balancing for MAN Türkiye A.Ş.” and is supported by SANTEZ with project number 00695.STZ.2010-2.

ÖZ

ÇOK TARAFLI KARIŞIK MODELLİ MONTAJ HATTI DENGELENMESİ
VE MODEL SIRALAMASI

GÜNER, Nizamettin Doğan

Yüksek Lisans, Endüstri Mühendisliği Ana Bilim .Dalı
Tez Yöneticisi: Yrd. Doç. Dr. Benhür SATIR
Ortak Tez Yöneticisi: Prof. Dr. Levent KANDİLLER
Eylül 2013, 83 sayfa

Montaja dayalı üretim, günümüz gelişen ekonomilerinin büyük ölçekli sanayilerinde önemli bir yer tutmaktadır. Geçmişten günümüze, montaj işlerinin hedeflenen zamanda bitirilebilmesi için iyileştirme çalışmaları yapmak, firmaların öncelikli hedefi olmuştur. Bu doğrultuda, Türkiye'nin en büyük üretim tesislerinden biri olan otobüs üreticisi MAN Türkiye A.Ş., siparişlerle belirlenen günlük üretim miktarı hedefine ulaşarak termin süresini tam zamanında karşılayabilmeyi ve bu sayede montajdaki gecikmeleri engelleyerek gecikme maliyetlerini en aza indirmeyi hedeflemektedir. Bu tez kapsamında, üç ana tip otobüs, hedeflenen çevrim süresi ve sabit istasyon sayısı koşulları altında, montaj işlerinin en verimli şekilde istasyonlara dağıtılması ve gecikmeler en aza indirilerek araçların hatta ilerletilebilmesi için, matematiksel modelleme ve sezgisel metot yöntemleri kullanılarak montaj hattı dengeleme ve model sıralaması çalışmaları önerilmektedir. Bu çalışma, Türkiye Cumhuriyeti Bilim, Teknoloji ve Sanayi Bakanlığı SANTEZ programı kapsamında desteklenen bir projenin[2] parçasıdır.

Anahtar Kelimeler: Karışık montaj hatları, hat dengeleme, model sıralama

[2] Projenin adı 'MAN Türkiye A.Ş. için Çok işçili Çok Taraflı Karışık Modelli Üretim Hattı Dengelenmesi Karar Destek Sisteminin Geliştirilmesi'dir ve SANTEZ destek numarası 00695.STZ.2010-2'dir.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisors Prof. Dr. Levent Kandiller and Asst. Prof. Dr. Benhür Satır for their valuable guidance through all the stages of my M.Sc. study with wholehearted patience and sincerity.

I am grateful to my examining committee members, Asst. Prof. Dr. Nureddin Kırkavak, and Asst. Prof. Dr. Haluk Aygüneş and Asst. Prof. Dr. Mustafa Alp Ertem for accepting to be on my committee and for their valuable suggestions on my research.

I would also like to express my deepest thanks to my wife Funda Güner for all her endless academic and moral support during all my desperate times.

I would also express my thanks to my family for all the supports that they provided to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Assembly lines have a great importance for the firms compete in the market especially for the ones that show their activity in automobile industry. In automobile industry, assembly line differs from other assembly lines due to the multi-manned workers work simultaneously at same workstations. Therefore, assembly line balancing problems, on these lines, are not the classical defined problems in the literature. They are called general assembly line balancing problems. Not only assembly line balancing problem that is assigning tasks to the workstations on the assembly line, but also model sequencing problem arises when there is a product mix. Product mix means that product variations are manufactured on the same assembly lines. Because of the different workstation time requirements of each product variations at workstations, there exists overloaded and under loaded workstations. To overcome these issues, model sequencing approaches are used. In addition to these two concepts occurring at assembly lines, workforce scheduling problem should be taken into consideration. Combining these concepts (line balancing, model sequencing, workforce scheduling), firms become able to compete in the market. Figure 1 illustrates the relation between these three problems.

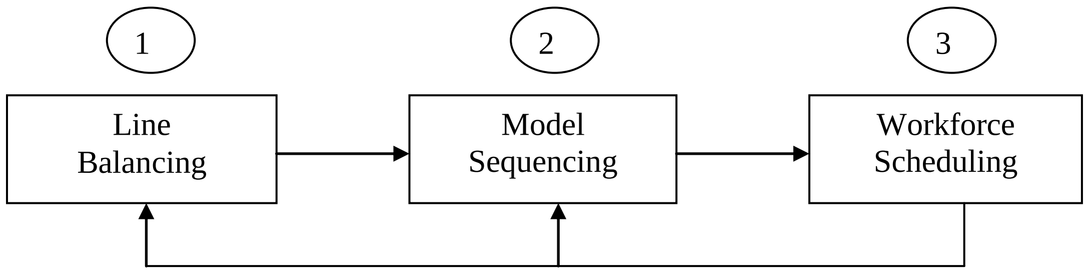


Figure 1: Operational Planning of Mixed-Model Assembly Line

As depicted in Figure 1 the first problem here is defined as line balancing problem that is assigning tasks to workstations in a simple setting. However, this study aims to cope with the overall optimization of assembly line defined in literature as the general assembly line problem. Generality comes from multi worker workstations on the assembly line. It means that workers perform their work simultaneously at workstations. This idea is a relatively new idea in assembly line literature

(Dimitriadis, 2006). Then, by using the results of line balancing problem, if the line is mixed model line that is varying products are manufactured on same line over large number of stations, the model sequencing problem must be solved to find the best production sequence. Bear in mind that, varying products require different workloads at the same workstations. This situation results in work overloads, blocking and starving, hence delays in production. Therefore, the model sequencing problem tries to find a good product sequence. As a final step workforce management is crucial as other two problems. This problem is not a long term decision problem like others. It should be solved on daily basis. As a recent study on workforce scheduling in assembly line, studied by (Karabak, 2012) as master thesis which was studied within the same project[3] with our study and partially supported by SANTEZ Program of Ministry of Science, Industry and Technology of Republic of Turkey. Our study and Karabak (2012) work on the same system with different problems however these problems are related with each other. Karabak (2012) interested in third problem in Figure 1. According to this figure, results of the line balancing (assigned tasks) and model sequencing (sequenced products) are the inputs for workforce scheduling. For the workforce assignment in a multi-worker multi-sided mixed model assembly line balancing problem, Karabak (2012) presents a mathematical formulation and a heuristic algorithm.

In the scope of this thesis study, line balancing and model sequencing problems are studied. Our study includes real life application at MAN Türkiye A.Ş. In Chapter 1, assembly line balancing and model sequencing related literature is reviewed. In Chapter 2, problem environment is described the problems are defined. Consequently, mathematical formulations are constructed. Because of the size of the real life problems, heuristic approaches are developed to overcome the intractability. In Chapter 3, algorithms using general solvers intaking mathematical formulations and heuristic approaches developed in Chapter 2 are compared by means of the designed data sets and real life data sets.

[3] Project name is “Decision Support System ofMulti Worker Multi Sided Mixed Model Assembly Line Balancing for MAN Türkiye A.Ş.” and is supported by SANTEZ with project number 00695.STZ.2010-2.

CHAPTER 1
LITERATURE REVIEW

In this chapter, literature is reviewed for line balancing and model sequencing in correspondence with explanations of Chapter 1 and Figure 1. First, line balancing literature is given by summarizing the studies according to their interest areas. Then, model sequencing literature is presented with its all details.

1.1 LINE BALANCING

Assigning all tasks to the workstations by considering cycle time and precedence relations between all defined tasks is called as the assembly line balancing problem. In this context, it can be seen that many scientist have been dealt with assembly line balancing problem since Henry Ford's study on Ford T models which is the first and known assembly line related work published in 1915. On this assembly line, only single is produced. Thus, that assembly line is called as single model assembly line. For this type of assembly lines, Salveson (1955) presented the first mathematical model for single model assembly lines to balance the lines to minimize number of workstations in 1955. Also, assumptions for the single model assembly line balancing are as follows (Baybars, 1986):

- Mass production of one type of product
- All tasks are processed in predetermined mode
- Paced line with a fixed common cycle time according to a desired output quantity
- The line is considered to be serial with no feeder lines or parallel elements
- The processing sequence of tasks is subject to precedence relations
- Deterministic task times
- No further assignment restrictions except precedence relations
- A task cannot be split among two or more workstations
- All workstations are equally equipped with respect to machines and workers

When the above restrictions are modified according to the problem studied on hand, then it is called as the general assembly line balancing problem. We classified general assembly line balancing problems based on three characteristics of assembly lines pointed out by Boysen et. al. (2007).

1.1.1 Precedence Graph Characteristics

Here, six different attributes are seen for the classification. These are;

1. *Product specific precedence graph:* This attribute considers that whether single product with single precedence graph, similar precedence graphs for more products or totally different products with different precedence graphs. Therefore, under this attribute, it is seen that there are mixed model lines with varying models, multi model lines with different models and single model. The firms should respond to the customer needs in a very short time. In this manner, applying mixed model assembly lines is the most commonly used competitive product markets. Mixed model assembly line is the assembly line with variations on the main product in such a way that various model variations are manufactured on the same assembly line. It can be said that single model assembly lines (SAL) have been switched to a more flexible system, mixed model assembly lines (MMAL). After the first mathematical model, many academicians have been interested in assembly line problems. But it is seen that most of the researches reported are for the single model assembly lines. When we concentrate on the model assembly lines we see first, Thompoulus (1967) on the mixed model assembly lines. Thompoulus (1970) solved the balancing problem of mixed model lines by coming up a joint precedence diagram by considering all precedence diagrams of the other models can be used. Due to the difference of each different model variation in mixed model assembly lines, Becker and Scholl (2006) indicated that by using joint precedence diagrams, work overloads or idle times can occur on the line. In this manner, mixed model assembly line problems are harder than that of single model assembly line problems. Single model assembly line problems are known NP-Hard (Garey and Jhonson, 1979), therefore mixed model assembly line problems are also NP-Hard. Boysen et. al. (2008) propose a study to find which model to use.
2. *Structure of the precedence graph:* It is better to classify the precedence graph to get efficient specialized algorithms. Some precedence graphs can be linear, diverging and converging or can have any acyclic structure.

3. *Processing times:* For this attribute, it is seen in stochastic processing times, dynamic variations of processing times (due to learning effects) and static and deterministic processing times.
4. *Sequence dependent task time increment:* On the line, there can be some tasks whose sequence affects the line system by considering their processing times. This effect can be direct or indirect.
5. *Assignment restrictions:* It is possible to see on the assembly lines that subsets of tasks are linked such that these tasks must be assigned to the same workstation (task marriage) or it is possible to see incompatible tasks that must not be assigned at same workstation (task divorce). Other assignment restrictions can be assignment of tasks is based on constraints on the cumulated value of particular task time. Zoning restriction is that for a task should or should not be assigned to a workstation with truly equipped or not. Another type of zoning restriction is that minimum and maximum distance to other tasks in time or space should be considered.
6. *Processing alternatives:* If there are processing alternatives, it means that some of the precedence graph can be changed, sub graphs can be changed and time and cost of tasks can be changed.

1.1.2 Station and Line Characteristics

There are six different attributes with respect to station and line characteristics. These are as follows.

1. *Movement of work piece:* Work piece can be moved by strictly obeying cycle time restriction with or without a probability or it can be moved to another place such as buffers.
2. *Line layout:* Line can be serial, or U-shaped.
3. *Parallelization:* Assembly lines can be specialized by considering parallel workplaces on the same workstation. It means that, more than one worker can perform their work simultaneously on the same workstation. The first studies have been made on the two sided assembly lines defined by the Bartholdi (1993). The main idea behind this logic is decreasing the number of workstation on the assembly lines. Özcan and Toklu (2009) study two sided assembly line balancing problem. They propose a mathematical formulation and simulated annealing algorithm for two sided assembly line balancing

problem. Kim (2009) proposed a mathematical model and genetic algorithm for two sided assembly line balancing problem. Özcan (2010) studied two sided stochastic assembly line balancing problem. Purnomo et. al. (2013) propose a mathematical model for two sided assembly line balancing by aiming minimizing cycle time while balancing workstations simultaneously. Tapkana et. al. (2012) modeled and solved constrained two sided assembly line balancing problem with bee algorithms. Özbakır and Tapkan (2011) presented their work using bee colony in zone constrained two sided assembly line balancing problem. The more workplaces in a workstation have been studied by Becker and Scholl (2009) concentrating on the automobile assembly industry. The last study on parallelization of workstation was made by Kellegöz and Toklu (2012). They proposed an efficient branch and bound algorithm for the defined problem. Guresky et. al. (2013) study a general assembly line balancing problem that has several workplaces at each workstation. Abdolreza et. al. (2013) report simulated annealing algorithm for multi-manned assembly line balancing problem.

4. *Resource assignment:* To perform tasks at workstations, multiple workers, machines and tools can be necessary.
5. *Workstation dependent time increment:* At workstations, some activities such as walking, transferring work piece to another place are time consuming. Hence, these are workstation dependent activities.
6. *Additional aspects of line configuration:* Buffer, feeder, etc.

1.1.3 Objectives

As a third characteristic, a classification can be made with respect to the objectives that evaluate the solution. These are as follows:

1. Minimizing number of workstations
2. Minimizing number of cycle time or maximizing the production rate
3. Maximizing the line efficiency
4. Cost minimization
5. Profit maximization
6. Smoothing workstation times
7. Balancing workstation times over all assembly line

8. Minimizing or maximizing a score defined by bottleneck aspects, required grip strengths, quality of work piece position changes etc.

Besides single objectives two and more of above ones can be thought as multi objective.

In Table 1 on next page, the summary of the literature is made.

Table 1: Summary of the Literature

Author Name	Line Balancing Problem Type
Bartholdi (1993)	Two sided assembly line balancing problem
Özcan and Toklu (2009)	Two sided assembly line balancing problem
Becker and Scholl (2009)	Single model assembly line balancing problem with variable workplaces
Thompohoulus (1967)	Mixed model assembly line balancing
Thompohoulus (1970)	Mixed model assembly line balancing
Kellegöz and Toklu (2012)	Single model assembly line balancing with parallel multimanned workstations
Baybars (1986)	Single model assembly line balancing problem
Boysen et. al. (2007)	Assembly line balancing problem
Boysen et. al. (2008)	Assembly line balancing problem
Purnomo et. al. (2013)	Two sided assembly line balancing problem
Gurseky et. al. (2013)	General assembly line balancing problem
Abdolreza et. al. (2013)	Multimanned assembly line balancing problem
Tapkana et. al. (2012)	Two sided assembly line balancing problem
Özbakır and Tapkan (2011)	Two sided assembly line balancing problem
Özcan (2010)	Two sided assembly line balancing problem
Kim (2009)	Two sided assembly line balancing problem

To compare our study with the studies in Table 1, it includes single model assembly line balancing problem with multi zone workstations which have different qualifications for the tasks to be assigned. The nearest study from Table 1 seems as Kellegöz and Toklu (2012) in the scope of single model assembly line balancing with parallel multimanned workstations but this study does not consider different qualifications for multimanned workstations.

1.2 MODEL SEQUENCING

In order to compete in the market, companies must implement mixed model assembly line strategies where variations of base models are produced on the same line with a mixed manner. However, by implementing mixed model assembly line approach, due to the different processing time of models at each station, work overload or underload situations occur in workstations. To avoid from these overloads and idle times, better sequencing of models are required. Hence, mixed model sequencing problems are among the short term decisions problems that give sequence of models for which demand of models over specified time period as elaborated in Boysen et. al. (2009).

Mixed model sequencing problems assume that line balance and the layout of the line are given, and the planning period is short term period (a week, a day). Usually launching discipline is fixed rate launching, that is work piece is launched in regular intervals. However, in the scope of this research, sequencing for unpaced lines whose workstations have buffers that hold work piece in the case of next workstation's work is not completed. This type of problems is studied by Scholl (1999).

Thompoholus (1967) proposed a sequencing procedure for determining the order of models that are fed to the assembly line. Dar-El et. al. (1975) proposed an algorithm to minimize the overall assembly line-length for no operator interference by considering closed and open workstations. They give lower bounds for the overall line-length. Dar-El and Navidi (1981) applied line balancing and mixed-model sequencing theory to the assembly of 'frames' used in the production of telephone exchanges. They used an approach that is based on the Dar-El/Cother mixed-model sequencing algorithm developed in 1975, and extended it to include an additional station lower bound. Milternburg (1989) developed a theoretical basis for scheduling Just-in-Time (JIT) production systems, and presented new scheduling algorithms and heuristics. Inman and Bulfin (1991) presented a new formulation and solution procedure to sequence a mixed model JIT assembly system. They provided a polynomial algorithm to determine the optimal sequence for an objective function that is mathematically different, but intuitively similar to the objective functions of previous researchers. Ding and Cheng (1993) proposed an effective algorithm for mixed model sequencing for JIT production systems. For the paced assembly lines Yano and Bolat (1989) presented a survey on mixed model sequencing problems and an approach. In 1992, Bard et. al. proposed a solution technique that gives optimal solutions for mixed model sequencing problems. Tsai (1995) studied mixed model sequencing problem to minimize utility work and risk of conveyor stoppage and also proved that mixed model sequencing problem is NP-Hard. Yano and Rachamadugu (1991) addressed the problem of sequencing operation, each of which is characterized by one of a large number of possible combinations of customer-specified options, on a paced assembly line. Boysen et. al. (2007) presented work overloads or idle times occur between the workstations in mixed model assembly line environment because of the diversity between the products. Therefore, production sequence of variations of a model should be decided. Work overloads occur when intensive model variations are produced consecutively in the corresponding stations. These work overloads are eliminated by utility workers. Utility workers are highly qualified workers that can handle each type of task (Scholl, 1999). The large variation reduces production efficiency considerably and may even cause a line stoppage.

Kim and Jeong (2007) aimed to determine the sequence of models to minimize the total unfinished work within their work zone. A generalized formulation of the product sequencing problem in MMAL is presented and developed an optimal procedure using Branch & Bound technique and a heuristic procedure using lower bound and local search.

Mirzapour and Aryanezhad (2009) proposed a hybrid algorithm based on Genetic Algorithm (GA) and event based procedure was developed to solve the as mixed model line balancing problem considering sub-lines.

Fattahi and Selahi (2009) presented a mathematical formulation and its solutions for small-sized problems and develop a hybrid metaheuristic algorithm based on simulated annealing and a heuristic algorithm as well.

Compared to all these cited literature above, this study includes multi zone line balancing problem with workstation qualification and model sequencing problem. Moreover, the solution procedure consists of separate approaches for these problems and hierarchical solutions.

CHAPTER 2

PROBLEM DEFINITION AND SOLUTION APPROACH

In the scope of this thesis, problems occurred in assembly lines are considered. Specifically, these problems have huge importance for the productivity of the firms. One of them is line balancing problem which includes allocating tasks to workstations under some objectives. Other one is model sequencing problem that finds better sequence for the given product mix. This study is valuable on the side of integrating these two problems with proposed solution approaches. Because in recent literature that was mentioned in the previous sections there are not many studies on the integrated field which the two approaches are used together. To the best of our knowledge the study proposed by Cevikcan (2009) includes a heuristic method for line balancing and model sequencing for truck assembly. However, problem definitions for line balancing and model sequencing in terms of constraints and included parameters are different than our's study.

2.1. LINE BALANCING

As known, line balancing problems are interest for the researchers from 1910, therefore up to now, we see so many different problem definitions and solution approaches. However, with the improving technologies and the market conditions, stimulated companies and researchers define new assembly line balancing problems. For example, size of the product transforms classical assembly line balancing problem to a general assembly line balancing problem by defining workstations with multi zones. This provides workers to perform their assigned tasks simultaneously at the same workstation on different zones. Moreover, for the workstations and zones, there could be assigning restrictions. In this study these type of restrictions called as qualifications.

In the scope of this study, a real life system is considered, therefore our problem definition for line balancing problem is formed by this real life system. In this system product is huge, and assembly operations are performed in the different zones of the product. This situation makes our problem multizonned. Besides, due to the physical conditions of the assembly line, we consider qualifications for the workstations for each task. At last, the real system contains more than one product model to be produced. Due to the structure of the relations between these product models, which are explained in detail in next section, we consider single line balancing problem for each model.

2.1.1 Mathematical Formulation

Before modeling the problem of assembly line balancing, some assumptions are made in the light of observations we made at the company and the literature review. These are,

- Each task can be assigned to only one zone.
- Only one task can be performed at a time at each zone.
- Processing times are deterministic and integer.
- Single product represents one of the models.
- Serial line layout is used.
- Travel and setup times between tasks are ignored.
- All necessary equipment is available at required workstations at all times.

Moreover, it is possible to see the assumptions by illustrating on Figure-2 below.

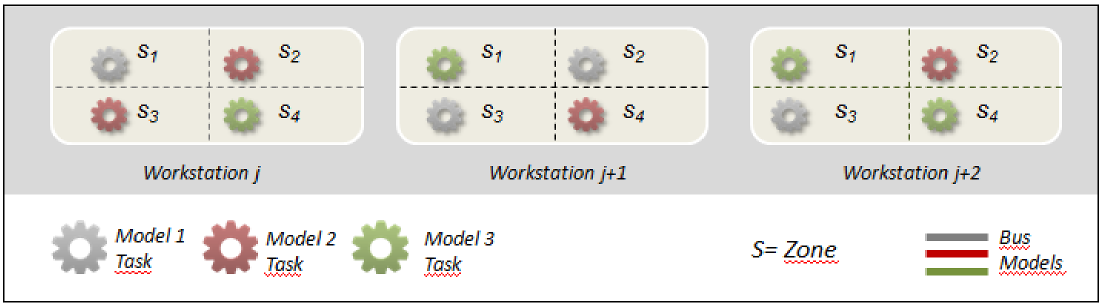


Figure 2: Line Balancing Assumptions

The current system has 3 main models of products, namely A78, R07 and R14. Because of this, by adding a new index for models, mathematical model is extended to balance mixed model assembly lines. A joint precedence diagram should be composed in order to solve single assembly line balancing problem (SAP). Same tasks of different model of products are taken as one task in joint precedence diagrams. In current system, all assembly is classified in terms of processes and tasks. A process is defined as group of tasks. Amount of tasks and processes of current system amounts can be achieved from Table 2. Table 2 also shows the number of common processes between three main models. For the all three models and two model combinations of products, there are very few processes in common.

Table 2: Number of common processes between three main models

	Tasks	Process
A78	996	105
R07	1074	116
R14	977	117
	3047	338

# of Common Processes	
All models	5
A78-R07	4
A78-R14	7
R07-R14	30

The following mathematical model formulation is constructed for the defined SAP by using mixed integer programming.

Indices:

i,h=index for tasks

s,t=index for bus zones

j,g=index for workstations

P_i= set of tasks that have precedence relations between each other tasks

Parameters:

$TIME_i$ = Processing time of task i

$$PRE_{ih} = \begin{cases} 1;\text{if task i is immediately performed before task h} \\ 0;\text{otherwise} \end{cases}$$

$$wQUAL_{isj} = \begin{cases} 1;\text{if task i can be processed in zone s in workstation j} \\ 0;\text{otherwise} \end{cases}$$

CT=Cycle Time

V=Sufficiently large number

Decision Variables:

$$WASG_{isj} = \begin{cases} 1;\text{if task i is assigned to zone s in workstation j} \\ 0;\text{otherwise} \end{cases}$$

B_i =Start time of task i

$$Z_{ih} = \begin{cases} 1;\text{if task i is immediately assigned before task h} \\ 0;\text{otherwise} \end{cases}$$

FINISH: Max. finish time of any task

The LB Model:

Minimize FINISH *1.0*

$subject\ to$

$$\sum_s \sum_j wQUAL_{isj}.WASG_{isj} = 1 \qquad \forall i \qquad 1.1$$

$$\sum_g \sum_t g.WASG_{htg} - \sum_i \sum_s j.WASG_{isj} \leq 0 \qquad \forall i \in P_i \qquad 1.2$$

$$\mathrm{B_h} + V(3 - WASG_{isj} - WASG_{htg} - Z_{ih}) \geq B_i + TIME_i \qquad i \notin P_i \qquad 1.3$$

$$\mathrm{B_i} + V(2 - WASG_{isj} - WASG_{htg} + Z_{ih}) \geq B_h + TIME_h \qquad i \notin P_i \qquad 1.4$$

$$B_h \geq B_i + TIME_i \qquad \forall i, h \qquad 1.5$$

$$CT - TIME_i \geq B_i \qquad \forall i \qquad 1.6$$

$$FINISH \geq B_i + TIME_i \qquad \forall i \qquad 1.7$$

$$FINISH \geq 0 \qquad 1.8$$

$$Z_{ih} = \{0,1\} \qquad \forall i, h \qquad 1.9$$

$$WASG_{isj} = \{0,1\} \qquad \forall i, s, j \qquad 1.10$$

$$B_i \geq 0 \qquad \forall i \qquad 1.11$$

The objective function is the minimization of the finish time of all tasks; cycle time shown as in 1.0. Equation 1.1 says that only one task should be assigned to one zone in a workstation. 1.2 is used for the precedence relations between tasks for the workstations. Equations 1.3, 1.4 and 1.5 are used for the start time of tasks that have no precedence relations. 1.3 says that if task *i* is assigned before task *h* at same workstation, start time of task *h* is greater than finish time of task *i*. 1.6 indicates that completion time of each task should be less than the cycle time. Equation 1.7 is used for the calculation of finish time of tasks. In 1.8 through 1.11 the binary and non-negativity constraints are shown.

2.1.2 A Heuristic Approach - Line Balancing

Using the mathematical formulation developed in the previous section or another similar model, getting optimal solutions in a reasonable time via a general solver like

CPLEX is impossible, since the model is NP-Hard. Therefore, there is a need for a heuristic approach for the defined problem for real life instances. In this section, a heuristic approach is going to be explained in detailed. Three construction heuristics are defined as follows: Construction_1 based on longest processing time, Construction_2 based on processing time of all follower tasks, Construction_3 based on number of followers. Then, an improvement method based on a local search algorithm is developed.

2.1.2.1 Construction_1

This construction algorithm is based on largest processing time. Steps of the algorithm are defined below. Figure 2 shows the steps on a flow diagram.

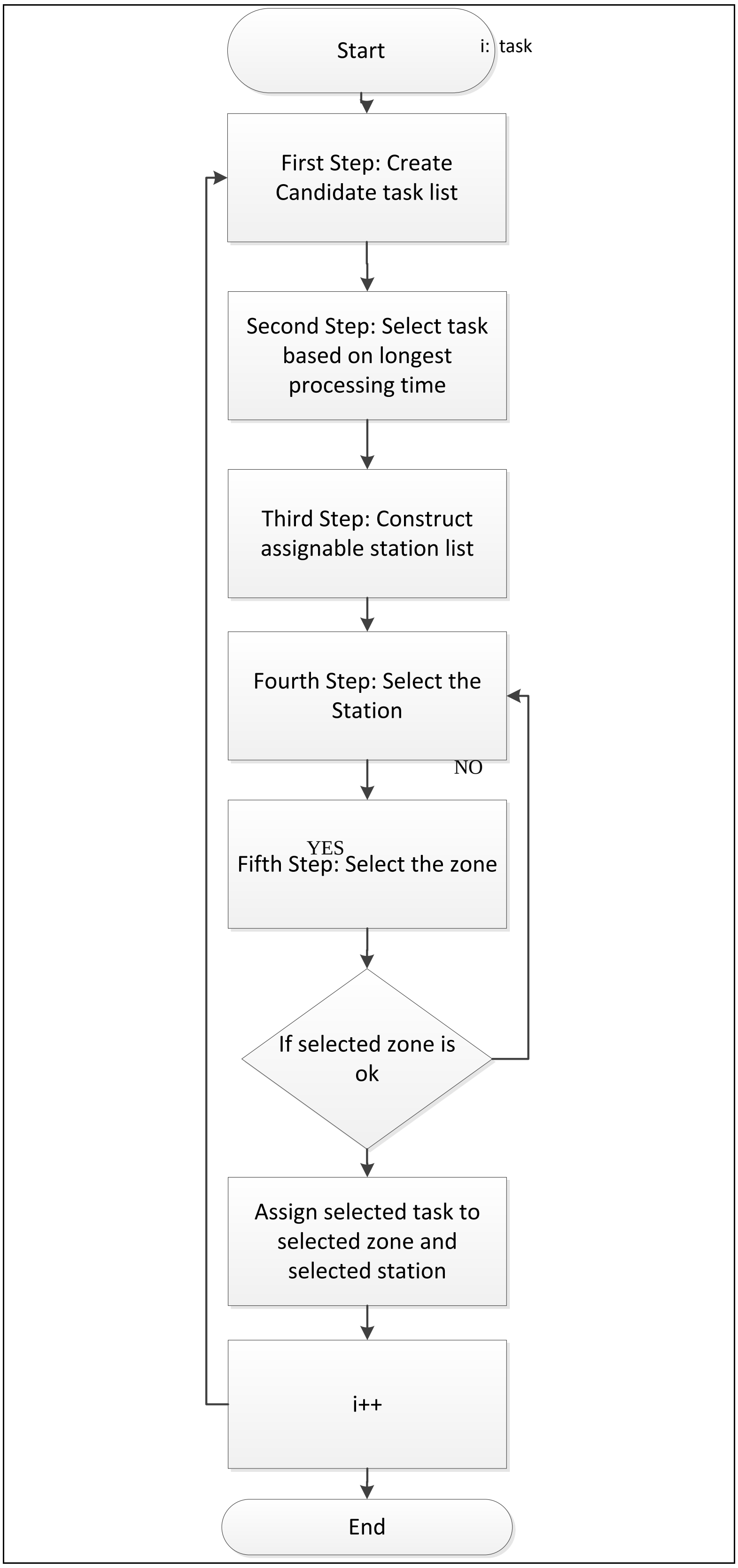


Figure 3: Steps of Construction_1 Heuristic

First Step: Represents the constructing candidate task list to be assigned to zones in workstations by using precedence relations.

Second Step: Represents the selected task that should be assigned first based on a priority rule that is the longest processing time.

Third Step: Represents the constructing station list based on the station-qualifications for selected task.

Fourth Step: Selection rule of one of the station from station list. That selection rule says that one should be selected among the candidate stations with minimum station id number and until number of total assigned task is equal to number of assigned tasks. Then, select the next station with minimum id. This is required to get minimum cycle time.

Fifth Step: Selection for zone under selected station.

Sixth Step: Zone selection rule is represented. Selection is performed based on qualification and zone workload.

Seventh Step: Represents the procedure whether zone is selected or not based on a rule applied. And if the zone is not selected, algorithm says that go on next station from the candidate station list. If it is selected, assign the selected task and go on with the next task from candidate task list.

Eighth Step: Procedure should be applied until all tasks are assigned.

2.1.2.2 Construction_2

The only difference from Construction_1 occurs in Step 2 such that tasks are selected for the candidate list based on a priority rule. Here the priority rule is total processing time of all followers of a task. The rest is the same with Construction_1.

2.1.2.3 Construction_3

In this heuristic, our priority rule is total number of followers of a task. The rest is the same with Construction_1.

2.1.2.4 Improvement Algorithm

The improvement algorithm is a local search algorithm where the neighborhood of the solution is searched and insertion used. Pseudo code is shown Figure 3.

```
Initialization.
Iteration=1
for (iteration =1, iteration <= number of zone, iteration ++)
{
            Sort load of zones in descending order
            Max load zone = sortedzones[iteration]
    for (all tasks of maxload zone)
    {
            for (all zones)
            {
              Insert task by checking constraints that are
              zoneQUAL, stationQUAL, PRE starting from min
              load zone.

              Calculate new cycle time,

              If new cycle time < current cycle time accept
              insertion process
            }
    }
}
End
```

Figure 4: Pseudo Code for Improvement Algorithm

Example:

It is better to define small example to understand how proposed algorithms work. Below, parameters are tabulated. Table 3 shows the processing times of 10 tasks to be assigned to workstations. Table 4 presents precedence matrix for tasks. Table 5 and Table 6 are for qualification of zones and workstations for tasks, respectively.

Parameters:

Table 3: Processing Times of 10 tasks

Task Number	Processing Time
1	19
2	26
3	10
4	46
5	23
6	21
7	31
8	35
9	30
10	8

Table 4: Precedence Matrix

T/T	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Table 5: Zone qualification

Z/T	1	2	3	4	5	6	7	8	9	10
z1	1	1	0	0	1	0	0	0	0	1
z2	0	0	0	1	0	1	1	0	0	0
z3	0	0	1	0	0	0	0	1	1	0

Table 6: Workstation qualification

S/T	1	2	3	4	5	6	7	8	9	10
s1	1	1	1	1	1	1	1	1	1	1
s2	1	1	1	1	1	1	1	1	1	1
s3	1	1	1	1	1	1	1	1	1	1

First Iteration:

First Step: Candidate Task List: {1, 3, 6 and 9}

Second Step: Order processing times descending: {9, 6, 1, 3}

Task to be assigned is 9.

Third Step: Candidate workstation list: {1, 2, 3}

Fourth Step: Selected workstation with minimum station id is 1.

Fifth and Sixth Step: In workstation 1, there is 3 zones. Between them one should be selected by considering qualification of zones and workloads.

Zone for task 9 is zone 3 with workload is 0. Assign task 9 to zone 3 in station 1.

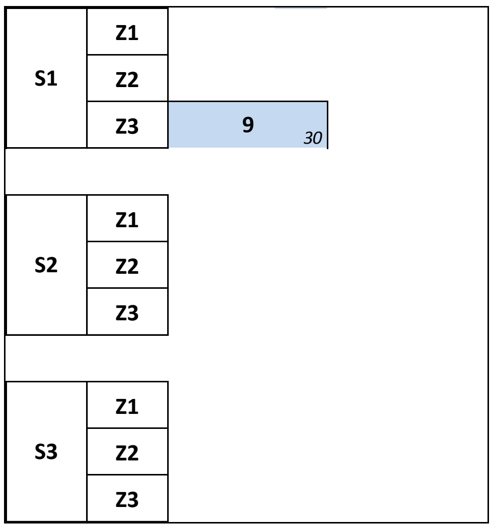


Figure 5: Scheme after task 9 is assigned

Second Iteration:

First Step: Candidate Task List: {1, 3, 6 and 10}

Second Step: Order processing times descending: {6, 1, 3, 10}

Task to be assigned is 6.

Third Step: Candidate workstation list: {1, 2, 3}

Forth Step: Selected workstation with minimum workstation id is 1.

Fifth and Sixth Step: In workstation 1, there are 3 zones. Between them one should be selected by considering qualification of zones and workloads.

Zone for task 6 is zone 2 with workload is 0. Assign task 6 to zone 2 in station 1.

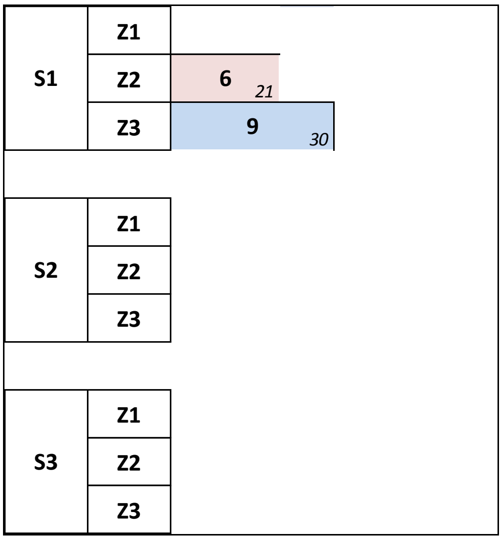


Figure 6: Scheme after task 6 is assigned

Third Iteration:

First Step: Candidate Task List: {1, 3 and 10}

Second Step: Order processing times descending: {1, 3, 10}

Task to be assigned is 1.

Third Step: Candidate workstation list: {1, 2, 3}

Fourth Step: Selected workstation with minimum workstation id is 1.

Fifth and Sixth Step: In workstation 1, there are 3 zones. Between them one should be selected by considering qualification of zones and workloads.

Zone for task 1 is zone 1 with workload is 0. Assign task 1 to zone 1 in station 1.

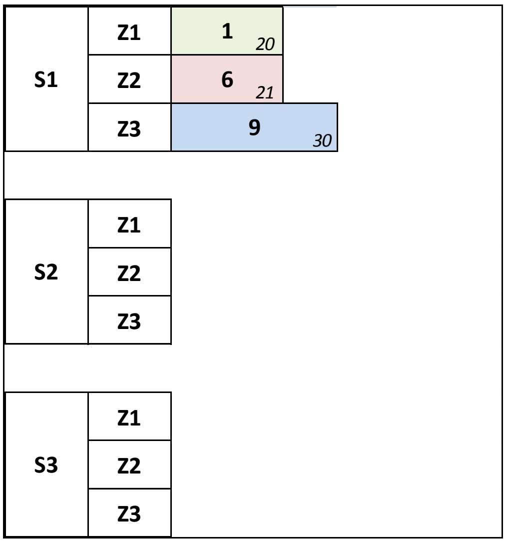


Figure 7. Scheme after task 1 is assigned

Fourth Iteration:

First Step: Candidate Task List: {2, 3 and 10}

Second step: Order processing times descending: {2, 3, 10}

Task to be assigned is 2.

Third Step: Candidate workstation list: {1, 2, 3}

Fourth Step: Selected workstation with minimum workstation id is 2. Because number of tasks assigned to station 1 is equal to number of zones in station 1.

Fifth and Sixth Step: In workstation 2, there are 3 zones. Between them one should be selected by considering qualification of zones and workloads.

Zone for task 2 is zone 1 with workload is 0. Assign task 2 to zone 1 in station 2.

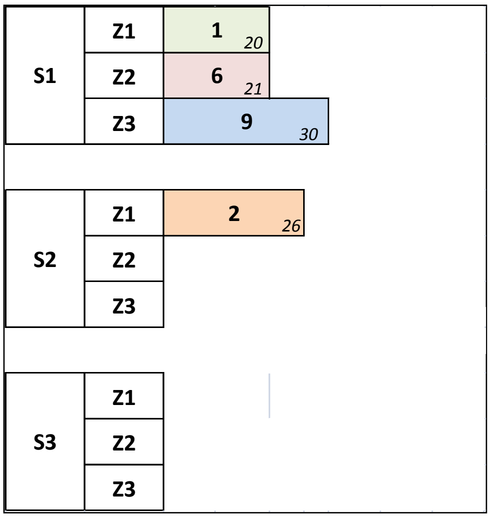


Figure 8: Scheme after task 2 is assigned

Note: Other remaining tasks are assigned by using the same manner with the assigned tasks. Figure-9 presents the final assignment.

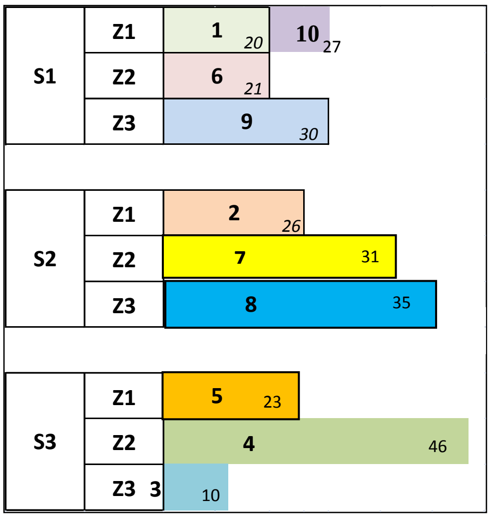


Figure 9: Final Assignment

An Example (Improvement)

To clarify the steps of improvement algorithm, a different instance used instead of using the instance of Construction Heuristic example.

To identify how the algorithm works it is better to show the steps on below example as an output of construction algorithm. Below, parameters are defined.

Parameters:

Table 7: Processing Times of 10 tasks

Task Number	Processing Time
1	5
2	3
3	10
4	12
5	4
6	9

Table 8: Precedence Matrix

T/T	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Table 9: Zone qualification

Z/T	1	2	3	4	5	6
z1	1	1	0	0	1	0
z2	0	0	0	1	0	1
z3	0	0	1	0	0	0

Table 10: Workstation qualification

S/T	1	2	3	4	5	6
s1	1	1	1	1	0	1
s2	1	0	1	1	1	1

In Figure 10, an assignment part for the workstations defined above (S1, and S2) with their zones (z1, z2, z3) is illustrated. Also, workloads of each zone are shown.

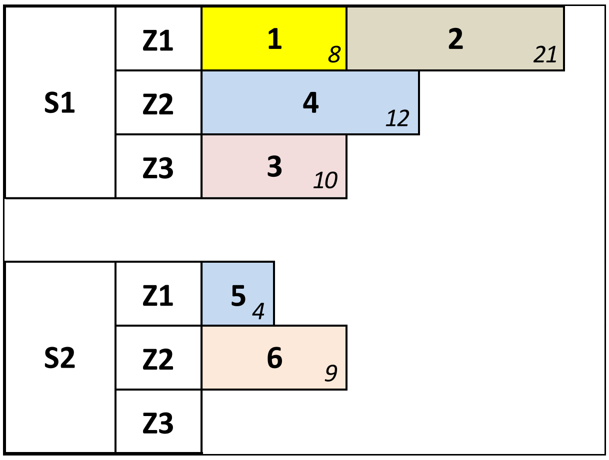


Figure 10: Scheme before improvement algorithm

According to the improvement algorithm;

First Step: Sort the zones based on their workload in descending order:

S1-Z1, S1-Z2, S1-Z3, S2-Z2, S2-Z1.

Find the maximum load zone with workstation id. It is zone 1 in Workstation 1.

Second Step: For all tasks in zone 1 at Workstation 1, search for the better place to reduce cycle time (maximum finish time) by considering zone and workstation qualifications and precedence relations.

Here, Task 1 in zone 1 at workstation 1 is candidate task to change its position. When qualifications and precedence relations are checked, it can be seen that Task 1 can be transferred to zone 1 in Workstation 2. Then reduced cycle time will be 13 instead of 21. Figure 11 shows the scheme after improvement algorithm is implied.

This procedure is applied for all tasks in all zones until getting better cycle time values.

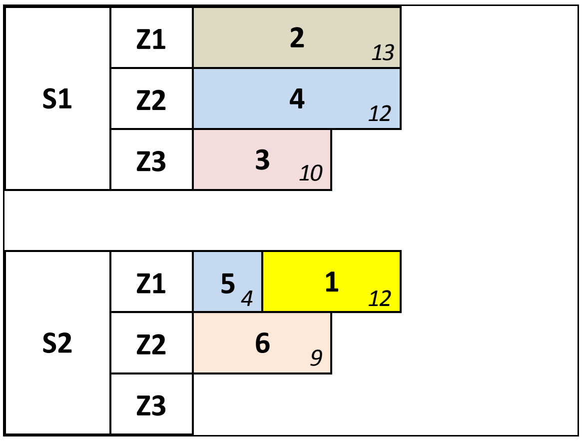


Figure 11: Scheme after improvement algorithm

2.1.3 Experimental Design

In this section, experimental designs for the proposed problem are made. Then, the results are taken from the solution of proposed mathematical formulations and heuristic procedures. As a solution methodology for the line balancing problem, a mixed integer linear mathematical formulation coded with GAMS 23.0 and heuristic algorithms (three construction algorithms with one improvement algorithm) coded in C language are proposed. In this section, theoretical and implementation results of these algorithms are presented. An example GAMS code for small sized problem is given in Appendix A.

2.1.3.1 Small Sized Problems (30 Tasks)

To the best of our knowledge the defined problem that is single model line balancing problem with station-zone qualification in multi-manned workstations is an original problem, not studied in the literature. Therefore, finding benchmarking problem instances becomes impossible. Due to this reason, test problems' data are generated for the stated problem. Below, data generation process is explained in detail.

Number of Tasks (I): 30 tasks are selected as the number of tasks for the line balancing procedure.

Processing time of tasks ($TIME_i$): Processing times of tasks are assumed to be deterministic and generated by uniformly distributed random integer numbers between 5-20 minutes.

Number of workstations: It is assumed that whole system has 3 workstations.

Number of zones: It is assumed that in each workstation 4 zones are available.

Qualification Matrix for Station-Zone (WQUAL): It is a binary matrix, in this matrix each task can only be performed in only one zone. However, each task is performed in 2/3 of all workstations.

Precedence Matrix (PRE): Flexibility Ratio (F-ratio) is used to construct a precedence matrix. F-ratio measures the precedence relations between tasks and represented by;

$$F-ratio=\frac{2\times B}{I(I-1)} \text{ where B is the number of 0's in the upper triangular matrix}$$

For the stated number of tasks, F-ratio is set to 70%.

All in all, 20 problem instances are generated. Each instance can be considered as a separate model, since has its own processing times.

2.1.3.2 Medium Sized Problems (60 Tasks)

20 problem instances are generated similar to small sized problems with the following differences: number of tasks is 60 and there are 4 workstations.

2.1.3.3 Large Sized Problems (90 Tasks)

As said before, the problem comes from the real life. Therefore, it is important that the proposed heuristic algorithm gives acceptable results for a large size data set. In this context, a data set for so called large size problem for 20 problem instances are generated similar to small sized problems with the following differences: number of tasks is 90 and there are 5 workstations.

2.1.3.4 Computational Results

For the generated data sets, three construction heuristics (Construction 1, Construction 2, Construction 3) are run with improvement heuristic. All heuristic results and optimal results are shown in Table 11, 12 and 13 with associated CPU times.

Table 11: Small Sized Results for Construction_1, 2, 3 and Improvement

Instance #	*GAMS Solution*	*GAMS Best Possible*	*CPU (sec)*	*Construction_1*			*Construction_1+ Improvement*			*Construction_2*			*Construction_2+ Improvement*			*Construction_3*			*Construction_3+ Improvement*			*Best of Construction*	
				Solution	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*Which Algorithm*
1	68	68	29.89	No Solution																			
2	50	50	82.94	69	28%	0.666	69	0%	0.753	74	32%	0.987	74	0%	0.847	59	15%	0.679	59	0%	0.835	59	C3
3	58	58	316.3	86	33%	0.696	86	0%	0.623	84	31%	0.812	84	0%	0.832	64	9%	0.758	64	0%	0.795	64	C3
4	57	57	598.27	76	25%	0.528	76	0%	0.703	65	12%	0.847	65	0%	0.697	66	14%	0.802	66	0%	0.789	65	C3+i
5	63	57	7200	78	19%	0.498	78	0%	0.758	72	13%	0.797	72	0%	0.784	63	0%	0.68	63	0%	0.863	63	C3
6	52	52	56.42	58	10%	0.561	58	0%	0.759	66	21%	0.963	66	0%	0.845	52	0%	0.896	52	0%	0.913	52	C3
7	59	58	7200	72	18%	0.718	72	0%	0.749	72	18%	0.851	72	0%	0.727	60	2%	0.899	60	0%	0.91	60	C3
8	55	55	78.25	87	37%	0.639	87	0%	0.689	63	13%	0.989	63	0%	0.881	63	13%	0.925	63	0%	0.812	63	C3
9	58	51	7200	73	21%	0.653	73	0%	0.678	66	12%	0.855	66	0%	0.852	58	0%	0.798	58	0%	0.865	58	C3
10	47	47	141.81	66	29%	0.686	66	0%	0.737	55	15%	0.765	55	0%	0.742	53	11%	0.821	53	0%	0.929	53	C3
11	55	55	213.67	68	19%	0.565	68	0%	0.757	68	19%	0.892	68	0%	0.658	62	11%	0.688	62	0%	0.858	62	C3
12	52	52	160.25	69	25%	0.622	69	0%	0.739	68	24%	0.679	68	0%	0.815	53	2%	0.952	53	0%	0.865	53	C3
13	58	58	369.81	59	2%	0.717	59	0%	0.713	78	26%	0.976	78	0%	0.598	63	8%	0.836	63	0%	0.803	59	C3+i
14	48	48	356.99	68	29%	0.624	68	0%	0.759	62	23%	0.844	62	0%	0.821	57	16%	0.722	57	0%	0.899	57	C3
15	59	59	1279.64	84	30%	0.642	84	0%	0.635	68	13%	0.817	68	0%	0.812	59	0%	0.684	59	0%	0.962	59	C3
16	55	55	489.5	66	17%	0.581	66	0%	0.705	59	7%	0.582	59	0%	0.86	No Solution			53	0%	0.9	53	C3
17	53	53	166.83	74	28%	0.486	74	0%	0.623	65	18%	0.92	65	0%	0.594	58	9%	0.954	58	0%	0.956	58	C3
18	63	51	7200	92	32%	0.569	92	0%	0.643	67	6%	0.73	67	0%	0.675	60	-5%	0.771	60	0%	0.84	60	C3
19	55	54	7200	65	15%	0.493	65	0%	0.695	69	20%	0.745	69	0%	0.75	63	13%	0.871	63	0%	0.856	63	C3
20	49	49	592.44	67	27%	0.699	67	0%	0.674	61	20%	0.586	61	0%	0.863	51	4%	0.819	51	0%	0.959	51	C3

Table 12: Medium Sized Results for Construction_1, 2, 3 and improvement

Instance #	*GAMS Solution*	*GAMS Best Possible*	*CPU (sec)*	*Construction_1*			*Construction_1 + Improvement*			*Construction_2*			*Construction_2 + Improvement*			*Construction_3*			*Construction_3 + Improvement*			*Best of Construction*	
				Solution	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*Which Algorithm*
1	106	91	7200	131	19%	1.721	131	0%	2.112	141	25%	2.018	136	4%	2.294	136	22%	4.934	136	0%	2.441	131	C1
2	120	93	7200	142	15%	1.692	142	0%	1.807	121	1%	1.966	121	0%	2.567	107	-12%	2.189	107	0%	2.851	107	C3
3	122	97	7200	170	28%	1.529	170	0%	2.069	161	24%	1.851	161	0%	2.178	125	2%	1.937	125	0%	2.902	125	C3
4	*Out of memory*			146	100%	1.528	146	0%	2.183	148	100%	1.942	148	0%	2.143	122	100%	2.054	122	0%	2.405	122	C3
5	158	82	7200	102	-55%	1.600	102	0%	1.845	111	-42%	1.944	111	0%	2.41	100	-58%	2.096	100	0%	2.242	100	C3
6	*Out of memory*			156	100%	1.596	156	0%	1.882	141	100%	2.066	141	0%	2.169	127	100%	1.840	127	0%	2.32	127	C3
7	114	83	7200	126	10%	1.600	126	0%	2.584	139	18%	1.911	139	0%	2.296	120	5%	1.781	120	0%	2.312	120	C3
8	141	96	7200	130	-8%	1.581	130	0%	2.11	138	-2%	1.906	138	0%	2.466	108	-31%	1.828	108	0%	2.505	108	C3
9	135	84	7200	130	-4%	1.622	130	0%	1.902	120	-13%	1.858	120	0%	2.534	121	-12%	1.946	114	6%	2.2	114	C3
10	114	86	7200	120	5%	4.331	120	0%	4.95	144	21%	5.54	144	0%	2.14	133	14%	5.401	133	0%	9.122	120	C3+i
11	120	94	7200	154	22%	3.741	154	0%	5.523	141	15%	6.284	141	0%	6.64	114	-5%	5.206	114	0%	7.024	114	C3
12	119	93	7200	167	29%	1.408	167	0%	2.019	131	9%	1.749	131	0%	2.332	114	-4%	2.164	114	0%	2.686	114	C3
13	118	96	7200	146	19%	6.250	146	0%	6.747	131	10%	5.26	131	0%	6.727	114	-4%	5.36	114	0%	5.132	114	C3
14	140	93	7200	141	1%	4.319	136	4%	5.096	125	-12%	5.204	125	0%	7.252	110	-27%	5.409	110	0%	6.804	110	C3
15	100	78	7200	133	25%	1.501	133	0%	1.857	113	12%	1.938	113	0%	2.117	104	4%	2.137	104	0%	2.217	104	C3
16	*Out of memory*			154	100%	1.439	154	0%	1.633	131	100%	1.793	131	0%	2.113	118	100%	1.919	118	0%	2.535	118	C3
17	125	103	7200	121	-3%	1.420	121	0%	2.111	126	1%	1.938	126	0%	2.414	131	5%	1.911	131	0%	2.29	121	C3+i
18	125	91	7200	137	9%	1.399	137	0%	1.762	108	-16%	1.805	108	0%	2.783	103	-21%	1.887	103	0%	2.816	103	C3
19	117	86	7200	109	-7%	1.519	109	0%	1.757	125	6%	2.023	125	0%	2.027	107	-9%	1.918	107	0%	2.442	107	C3
20	106	91	7200	131	19%	1.721	131	0%	2.112	141	25%	2.018	136	4%	2.294	136	22%	4.934	136	0%	2.441	131	C3+i

Table 13: Large Sized Results for Construction_1, 2, 3 and improvement

Instance #	*GAMS Solution*	*GAMS Best Possible*	*CPU (sec)*	*Construction_1*			*Construction_1+ Improvement*			*Construction_2*			*Construction_2+ Improvement*			*Construction_3*			*Construction_3+ Improvement*			*Best of Construction*	
				Solution	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*%GAP*	*CPU (sec)*	*Solution*	*% Improvement*	*CPU (sec)*	*Solution*	*Which Algorithm*
1	*Out of memory*			173		2.971	173	0%	14.901	184		9.987	184	0%	13.008	166		10.512	166	0%	15.182	166	C3
2	*Out of memory*			211		8.161	211	0%	13.102	155		10.669	155	0%	14.501	159		10.596	159	0%	12.31	155	C2
3	*Out of memory*			190		8.159	181	5%	14.412	191		10.774	191	0%	14.747	172		10.408	172	0%	15.604	172	C3
4	*Out of memory*			237		8.044	237	0%	14.102	210		10.445	210	0%	14.515	197		10.581	197	0%	13.555	197	C3
5	*Out of memory*			198		8.234	198	0%	9.813	161		9.928	161	0%	12.32	148		10.309	148	0%	15.242	148	C3
6	*Out of memory*			187		2.805	187	0%	12.314	168		10.473	168	0%	13.787	161		10.6	161	0%	13.702	161	C3
7	*Out of memory*			151		2.772	151	0%	3.308	185		3.456	185	0%	5.222	165		3.595	165	0%	4.585	151	C2
8	*Out of memory*			184		2.84	184	0%	3.726	198		3.523	198	0%	4.721	181		3.473	181	0%	4.378	181	C3
9	*Out of memory*			236		2.844	236	0%	3.089	203		3.679	203	0%	4.296	237		3.502	237	0%	4.432	203	C2
10	*Out of memory*			229		3.286	229	0%	5.402	189		2.833	189	0%	4.385	189		3.563	189	0%	5.620	189	C3
11	*Out of memory*			171		3.298	171	0%	5.702	158		4.093	158	0%	4.907	158		3.603	158	0%	5.214	158	C2
12	*Out of memory*			157		2.608	157	0%	3.111	184		3.432	184	0%	5.604	145		3.464	145	0%	4.900	145	C3
13	*Out of memory*			189		2.688	189	0%	3.607	196		3.52	196	0%	4.740	151		3.484	151	0%	4.422	151	C3
14	*Out of memory*			191		3.015	191	0%	3.856	192		3.518	192	0%	4.456	163		3.508	163	0%	4.478	163	C3
15	*Out of memory*			204		2.683	204	0%	3.759	188		3.586	188	0%	4.457	181		3.614	181	0%	4.511	181	C3
16	*Out of memory*			227		2.732	165	38%	4.021	206		3.434	206	0%	4.739	165		3.382	165	0%	4.368	165	C3
17	*Out of memory*			217		2.982	217	0%	3.418	196		3.537	196	0%	3.924	146		3.516	146	0%	4.911	146	C3
18	*Out of memory*			186		2.817	186	0%	3.248	176		3.629	176	0%	4.248	156		3.793	156	0%	4.502	156	C3
19	*Out of memory*			176		2.903	176	0%	3.271	172		3.468	172	0%	4.568	166		3.518	166	0%	4.556	166	C3
20	*Out of memory*			194		2.901	194	0%	3.337	198		3.575	198	0%	4.782	149		3.46	149	0%	4.411	149	C3

Table 11 shows the results of 20 small sized instances for optimal seeking and proposed heuristic algorithms. According to the table, it can be said that Construction 3 and Construction 3 with improvement algorithms gives better results than other heuristic algorithms. Also, CPU times for the proposed methods represented on the table.

Table 12 presents the results of 20 medium sized instances for optimal seeking and proposed heuristic algorithms. Based on the table, Construction 2, Construction 3 and Construction 3 with improvement algorithms over perform the Construction 1 algorithm with improvement.

Table13 tabulates the results of 20 large sized instances for optimal seeking and proposed heuristic algorithms. As Table 12, Construction 2 and Construction 3 with improvement have better performance than Construction 1 with improvement.

As a summary, algorithms Construction 3 and Construction 3 with improvement has better effect on the solution of instances than other proposed heuristic algorithms.

2.2. MODEL SEQUENCING

In real system, large sized products are produced, with great variety. These varying product models are launched in a mix manner to the assembly line under the same cycle time. Therefore, it is possible to observe delays on assembly line which has no buffers between workstation. Thus, high amount of delays occur on those workstations and assembly line becomes asynchronous. Therefore, other aim of this study is sequencing different types of products for daily, weekly and monthly time periods by satisfying their demand amounts with the minimum amount of delays.

In the assembly line, for each period, products enter the frontier workstations. Due to this situation, stating time of a product in a workstation is depended on the frontier product in the assembly line.

In the scope of this study, two approaches for sequencing problem is considered. In the first approach, synchronous scenario, it is aimed to measure delays based on the pre-determined cycle time. For this scenario, also of synchronous structure of the line is evaluated. In the second approach, asynchronous scenario, delays on assembly line is based on starting and finishing times of each product.

Synchronous and asynchronous assembly line flows are shown in Figure 10 and 11 below.

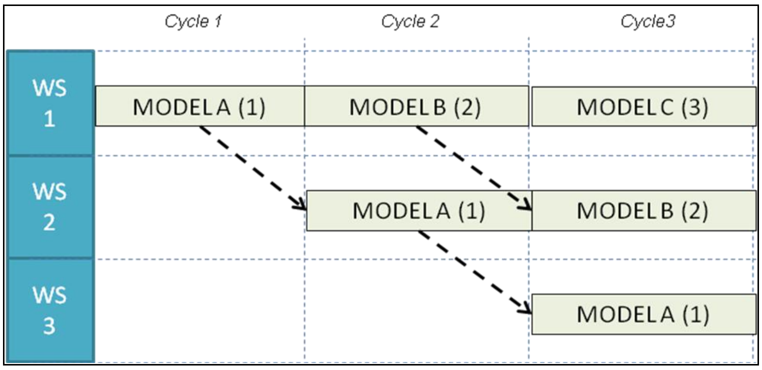


Figure 12: Synchronous assembly line diagram

In Figure 12, numbers that are written in brackets indicate the sequence position of that product. At each cycle, products enter next workstations, consecutively. Cycles that are shown in the figure are predetermined. In other words, if only all workloads are equal or below from cycle time, products can move to next workstation according to these cycle times. For each cycle, all products on the assembly line moves on same time is called as synchronous.

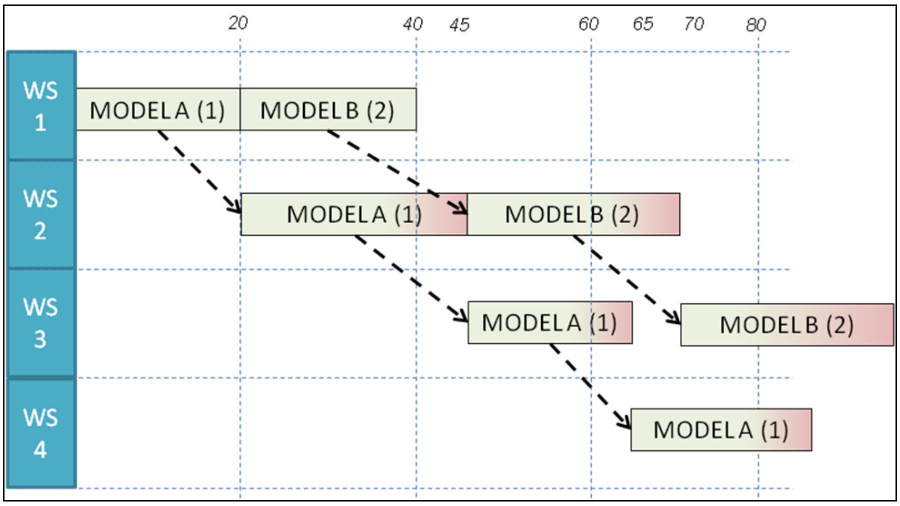


Figure 13: Asynchronous assembly line diagram

But in real life, when a products moves to next station if its assembly is completed, next workstation can be free. In other words, assembly of frontier products is being completed first. This situation can be seen in Figure 13. In Figure 13, “B” corresponds to blocking situation and “ST” is used to show starving situation.

In real life, company managers want to have a synchronous assembly line but in fact line flows asynchronous because of the unbalanced workloads. In that direction, both scenarios are considered.

1. *Synchronous Line Scenario:* There is a predetermined cycle time. Delays are defined as difference between cumulative cycle time and starting time of a product on a workstation. This type of line is used to evaluate *tardiness* amount if a cycle time is determined.
2. *Asynchronous Line Scenario:* There is not a cycle time constraint. Delays are defined as *blocking* and *starving* times.

If there is a predetermined cycle time, it is expected that line should flow according to this cycle time, in other words products move synchronously. But due to the overloads on workstation times, *tardiness* occurs on overloaded workstations. This situation is shown in Figure 14.

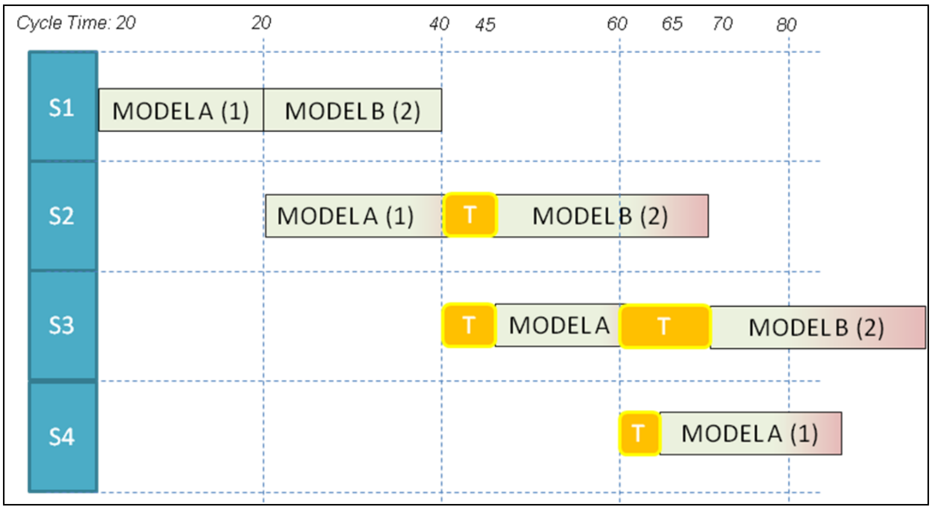


Figure 14: Tardiness situation

Tardiness time: It is defined as difference between cycle time and starting time. If starting time of position ***t*** on workstation ***k*** is greater than cumulative cycle time on that cycle, tardiness occurs. As seen in Figure 14, starting time of Model B in position (2) on workstation 2 is 45. But cumulative cycle time is 40 for that cycle. So, tardiness is 5 (45 – 40 = 5) for position (2) on workstation 2.

Blocking and *starving* situations occur in asynchronous lines. Figure 13 shows how blocking and starving times are generated, with the help of a numerical illustration.

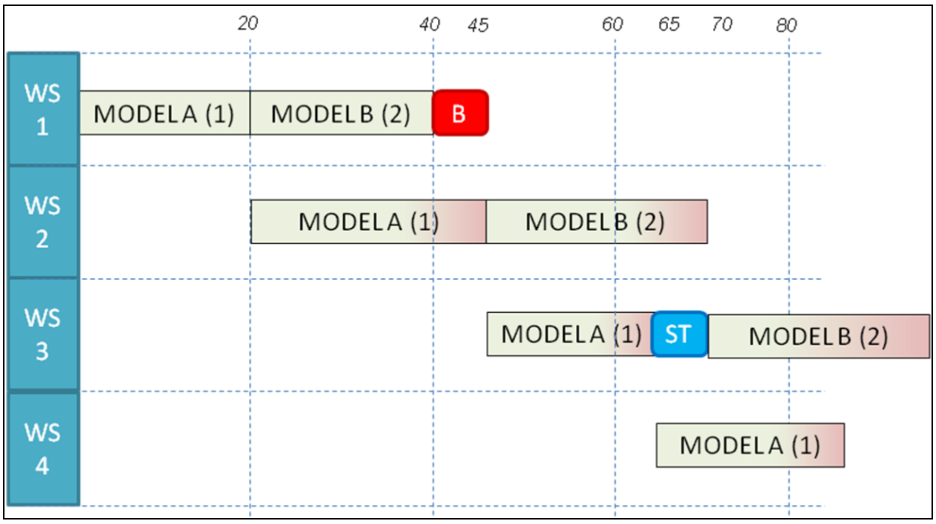


Figure 15: Blocking and starving situations

Blocking time: If assembly of position ***t+1*** is completed on workstation ***k*** but assembly of position t is not completed on workstation ***k+1*** or it cannot move to next workstation, blocking occurs for workstation ***k+1*** for position ***t+1***. As seen in Figure 15, Model B in position (2) and workstation 1 is completed at time 40. But frontier products, Model A, in position (1) and workstation 2 is completed at time 45 and then it moves to next workstation. So, blocking time can be calculated as 45 – 40 = 5 for position (2) on workstation 2.

Starving time: In contrast of blocking, if assembly of position ***t*** is completed on workstation ***k+1*** and it move to next workstation but assembly of position ***t+1*** is not completed on workstation ***k***, starving occurs for workstation ***k+1*** for position ***t+1***. As seen in Figure 15, Model A in position (1) and workstation 3 is completed at time 50. But previous products, Model B, in position (2) and workstation 2 is completed at time 52 and then it moves to next workstation. So, starving time can be calculated as 52 – 50 = 2 for position (2) on workstation 3.

To evaluate the *synchronous* and *asynchronous* lines, total *tardiness* in synchronous line and total *lateness* (which is the sum of total blocking and starving times) in asynchronous line are considered.

In the following sections, we propose model formulations for each scenario.

2.2.1 Mathematical Formulations

Index sets, parameters and decision variables for the following two models are given below.

Index Sets:

i= index for vehicle types, where $i \in \{1,\ldots,I\}$;

k = index for workstations, where $k \in \{1,\ldots,K\}$;

t= index for sequence positions, where $t \in \{1,\ldots,T\}$;

Parameters:

d_i=demand for vehicle type i

P_{ik}=station loads of each vehicle type i for each workstation

CT = cycle time

Decision Variables:

$$X_{it} = \begin{cases} 1 & \text{if vehicle type i is assigned to position t} \\ 0 & \text{otherwise} \end{cases}$$

SS_{kt} = starting time of position t for workstation k

PP_{kt} = processing time of position t for workstation k

ST_{kt} = starving time of workstation k for position t

B_{kt} = blocking time of workstation k for position t

W_{kt} =tardiness from cycle time of position t for workstation k

2.2.1.1 MS-Sync Model

Min z

$$\sum_{k=1}^{K}\sum_{t=1}^{T} W_{kt} \qquad (2.0)$$

Subject to

$$\sum_{t=1}^{T} X_{it} \geq d_i \qquad \forall i \qquad (2.1)$$

$$\sum_{i=1}^{I} X_{it} = 1 \qquad \forall t \qquad (2.2)$$

$$PP_{kt} = \sum_{i=1}^{I} P_{ik}.X_{it} \qquad \forall k,t \qquad (2.3)$$

$$W_{kt} \geq SS_{kt} - CT.(k+t-2) \qquad \forall k>1, t>1 \qquad (2.4)$$

$$W_{kt} \geq SS_{kt} - CT.(k-1) \qquad \forall k>1, t=1 \qquad (2.5)$$

$$SS_{kt} \geq PP_{k,t-1} + SS_{k,t-1} \qquad \forall k, t>1 \qquad (2.6)$$

$$SS_{kt} \geq PP_{k-1,t} + SS_{k-1,t} \qquad \forall k>1, t \qquad (2.7)$$

$$SS_{kt} \geq PP_{k-1,t} + SS_{k-1,t} \qquad \forall k>1, t=1 \qquad (2.8)$$

$$SS_{kt} \geq SS_{k+1,t-1} \qquad \forall k=1, t>1 \qquad (2.9)$$

$$X_{it} \in 0,1 \qquad \forall i,t \qquad (2.10)$$

$$W_{kt} \geq 0 \qquad \forall k,t \qquad (2.11)$$

$$SS_{kt} \geq 0 \qquad \forall k,t \qquad (2.12)$$

$$PP_{kt} \geq 0 \qquad \forall k,t \qquad (2.13)$$

Equation 2.0 is the objective function that minimizes the total tardiness. Constraint 2.1 is demand satisfaction constraint. 2.2 is used for satisfying that only one vehicle can be assigned to one position. 2.3 is used for determining processing times of assigned vehicles. Constraints are 2.4 and 2.5 are used to determine tardiness for each position and workstation. In synchronous lines, the aim is processing all vehicles according to determined cycle time. Because of this, for each cycle, difference between starting time of a vehicle on any workstation and cycle time is defined as tardiness. It shows how much the vehicle is behind the cycle time. 2.6, 2.7, 2.8 and 2.9 are used for calculating starting times. These constraints ensure that a vehicle can enter following workstations when frontier vehicle exits from this workstation. The rest of the constraints are binary and non-negativity constraints.

2.2.1.2 MS-Async Model

Min z

$$\sum_{k=1}^{K}\sum_{t=1}^{T} B_{kt} + \sum_{k=1}^{K}\sum_{t=1}^{T} ST_{kt} \tag{3.0}$$

Subject to

$$\sum_{t=1}^{T} X_{it} \geq d_i \qquad \forall i \tag{3.1}$$

$$\sum_{i=1}^{I} X_{it} = 1 \qquad \forall t \tag{3.2}$$

$$PP_{kt} = \sum_{i=1}^{I} P_{ik}.X_{it} \qquad \forall k,t \tag{3.3}$$

$$B_{kt} \geq SS_{kt} - SS_{k-1,t} - PP_{k-1,t} \qquad \forall k > 1, t \tag{3.4}$$

$$ST_{kt} \geq SS_{kt} - SS_{k,t-1} - PP_{k,t-1} \qquad \forall k, t > 1 \tag{3.5}$$

$$SS_{kt} \geq PP_{k,t-1} + SS_{k,t-1} \qquad \forall k, t > 1 \tag{3.6}$$

$$SS_{kt} \geq PP_{k-1,t} + SS_{k-1,t} \qquad \forall k > 1, t \tag{3.7}$$

$$SS_{kt} \geq PP_{k-1,t} + SS_{k-1,t} \qquad \forall k > 1, t = 1 \tag{3.8}$$

$$SS_{kt} \geq SS_{k+1,t-1} \qquad \forall k = 1, t > 1 \tag{3.9}$$

$$B_{k,1} = 0 \qquad \forall k \tag{3.10}$$

$$ST_{k,1} = 0 \qquad \forall k \tag{3.11}$$

$$X_{it} \in 0,1 \qquad \forall i,t \tag{3.12}$$

$$B_{kt} \geq 0 \qquad \forall k,t \tag{3.13}$$

$$ST_{kt} \geq 0 \qquad \forall k,t \tag{3.14}$$

$$SS_{kt} \geq 0 \qquad \forall k,t \qquad (3.15)$$

$$PP_{kt} \geq 0 \qquad \forall k,t \qquad (3.16)$$

Equation 3.0 is the objective function that minimizes the total blocking and starving times. Blocking and starving times are determined on additional constraints 3.4 and 3.5. 3.10 and 3.11 indicate that blocking and starving times of the vehicle on first position are zero. The rest of all constraints are the same with Model 1. The proposed mathematical models are solved using GAMS 23.0. The models have been tested by generated data based on the real life problem structure in the following section.

2.2.2 Experimental Design

For the model sequencing problem, two different mixed integer linear programming formulations are constructed for synchronous and asynchronous assembly lines, respectively. Codes for these mathematical formulations are developed within GAMS 23.0. Since we are getting optimal solutions in a reasonable time for the model sequencing problem, there is no need for heuristic approaches. Example GAMS codes for synchronous and asynchronous sequencing problems are given in Appendix B.

As known from previous sections, model sequencing problem uses results of line balancing problems. Therefore, results of above experimental design for the line balancing problem are used to design experiments on model sequencing. For asynchronous scenario and synchronous scenario, total 18 instances (6 for small sized, 6 for medium sized and 6 for large sized problems) with multi models (5 models, 10 models and 20 models for small, medium and large sizes, respectively) are randomly selected.

In other words;

Number of product types (*i*)= as number of product types 5, 10 and 20 is used.

Number of workstations (*k*)= as number of workstation 3, 4 and 5 are used.

Number of sequence position (*t*)= 50 and 60 are used for number of sequence positions.

Number of demands for vehicle type *i* (*di*)= For each type of product demands quantity is selected without exceeding sequence position.

Station loads of each vehicle type i for each workstation (Pik) = Comes from the results of experimental design of line balancing.

Cycle time (CT) = As cycle time 65 min is selected.

All in all, 18 problem instances are generated.

2.2.2.1 Computational Results

Model 1 (Synchronous)

Table 14. Results for Synchronous Model

Instance #	*GAMS Solution*	*GAMS Best Possible*	*CPU (sec)*	*Number of Tasks*	*Number of Models*	*Number of WS*	*Number of Demands*
1	814	814	0.349	30	5	3	10 per model
2	411	411	0.559	30	10	3	5 per model
3	418	418	1.199	30	20	3	3 per model
4	277388	277388	37.285	60	5	4	10 per model
5	246858	246858	46.788	60	10	4	5 per model
6	327010	327010	12:59.0	60	20	4	3 per model
7	614467	614467	0.702	90	5	5	10 per model
8	654081	654081	5.445	90	10	5	5 per model
9	842830	841918	2 hours lim.	90	20	5	3 per model

Table 14 shows the results for synchronous model. 9 instances are run by using GAMS 22.6 to get optimal solutions. According to the table, if number of models increases computational complexity of the problem increases, therefore, CPU time also increases.

Model 2 (Asynchronous)

Table 15: Results for Asynchronous Model

Instance #	*GAMS Solution*	*GAMS Best Possible*	*CPU (sec)*	*Number of Tasks*	*Number of Models*	*Number of WS*	*Number of Demands*
1	5381	5381	2.873	30	5	3	10 per model
2	1211	927.7	2 hours lim	30	10	3	5 per model
3	618	186	2 hours lim	30	20	3	3 per model
4	1512	1099	2 hours lim	60	5	4	10 per model
5	1362	292	2 hours lim.	60	10	4	5 per model
6	2330	99.7	2 hours lim.	60	20	4	3 per model
7	4324	4284	2 hours lim.	90	5	5	10 per model
8	2562	1006	2 hours lim.	90	10	5	5 per model
9	4174	3983	2 hours lim.	90	20	5	3 per model

In Table 15, results for asynchronous model can be seen. Based on the table, in asynchronous model number of workstation increases computational complexity of the problem increases, hence CPU time also increases.

CHAPTER 3
INDUSTRIAL APPLICATION

This study is originated from a senior graduation project held at MAN Türkiye A.Ş. by the senior students and academics of Çankaya University Industrial Engineering Department within the academic year of 2009-2010. With the success of the senior project, a wider project is formulated for MAN Türkiye A.Ş. and submitted for financial support. The proposal is accepted and two Master of Science thesis studies are conducted under this project. The first thesis dealt with the problem for MAN Türkiye A.Ş.'s assembly line workforce scheduling. Instead of dealing with the overall assembly line problem, only the third part of the overall problem (depicted in Figure 1) is of concern for the first thesis study. At workstations of assembly line of MAN Türkiye A.Ş., where the product is a huge or space considerations in the facility, workers perform their tasks simultaneously in teams. MAN Türkiye A.Ş. produces buses in make to order manner. Thus, different types of buses are fed to line in a mix sequence. As a result of this, unbalanced task assignments and work overload and underloads are observed. Hence, it can be said that the majority of the symptoms pointed out are line balancing and model sequencing problems.

In the scope of this study, we aimed to balance workloads between workstations, and then by using the results of the line balancing as input, we fired to come up better model sequencing alternatives to prevent work overloads, production delays occurred at workstations. Hence, two sets of solution methodologies and tools are developed for these defined problems.

As seen from Figure 16, this study focuses to get solutions get separately and hierarchically, and to come up the integrated solution by the feedback mechanisms. First, line balancing problem is solved, then, by using the assignment results of line balancing problem, model sequencing approach is performed.

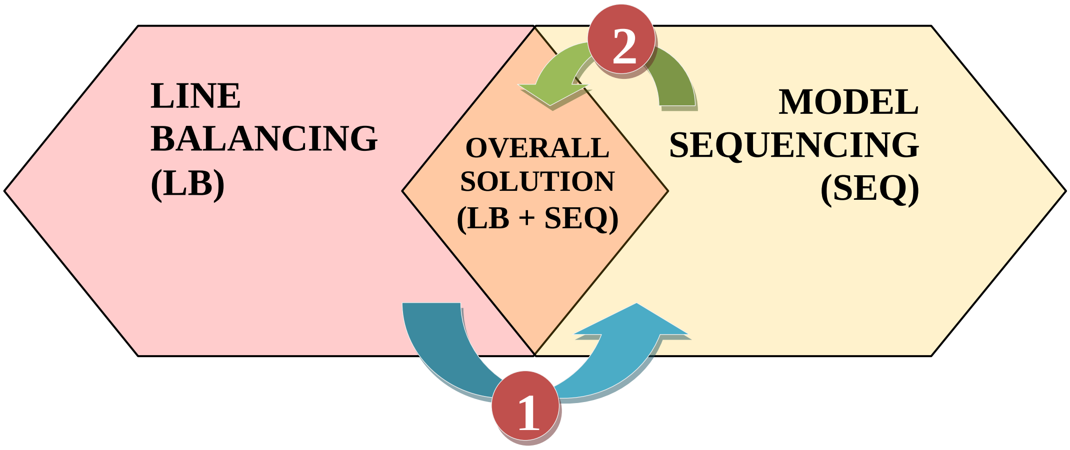


Figure 16: Operational Planning of Study

MAN Türkiye A.Ş. is divided into three production buildings named as Ü1, Ü2 and Ü3. There are four production sections in these buildings; Framework, Painting, Assembly and Finishing. A sketch of these sections and buildings can be seen in Figure 17.

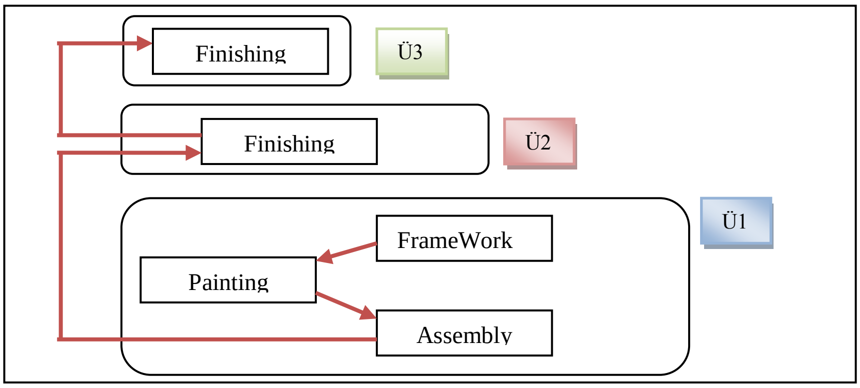


Figure 17: Production buildings and sections

In these buildings, there are approximately 110 workstations (in scope of this study, only 10 of them are considered because the project with held in MAN Türkiye A.Ş. includes only a part (10 workstations) of all workstations (110 Workstations) that represent the whole system.) where the predefined operations are performed by the workers and final product appears at the last station. Then, workstations can be divided into two categories on the production line (flowing) and preparation workstations that feed them. These workstations are explained in Figure 18.

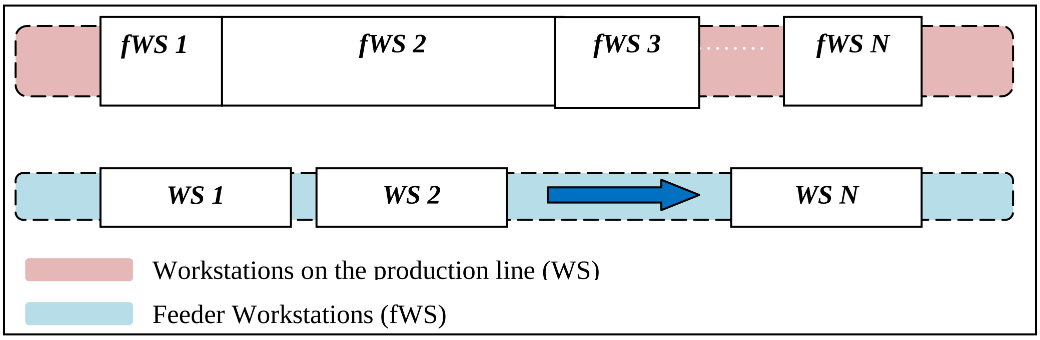


Figure 18: Flowing and Feeder Workstations

In MAN Türkiye A.Ş., three definitions are used for classification of processes. These are ***Process*** and ***Task***. This classification can be seen in Figure 18.

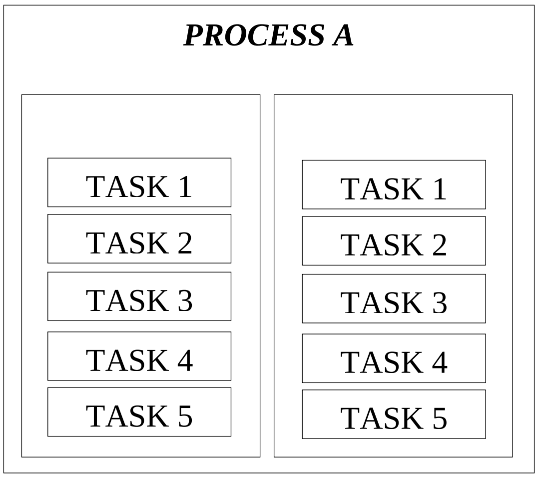


Figure 19: Process Classification

Sub groups of processes are tasks. For example, in "seat assembly" process, "moving seats from feeder workstation to bus" is a task. In this study, **processes are taken into consideration,** however, for easy understanding and continuing common usage in literature, processes are called as *tasks*.

Product types: There are three main products that manufactured by MAN Türkiye A.Ş. However, nearly 20 variations of these three main products are possible products that produced on the assembly line. In addition to this, customer needs may diversify the product variations more than 20.

The cycle time for each station is fixed. But, it is varied over time horizon. The operations that are assigned to the workstations have to be completed in the given cycle time. Because of large dimensions of the product, more than one worker

(worker teams) work simultaneously at different locations of the bus. These bus locations are defined as '***zones***'.

Precedence Relations

In the current system, precedence relations of tasks are not defined or they are not well defined. Therefore, precedence relations of processes are determined in the scope of this study by analyzing the gathered data.

Workstation and zone restrictions of tasks

Zone is a region of a bus where a certain task is processed. This information is required to assign tasks in parallel. In the real system, a task can be operated on more than one zone, but, in this study, it is assumed as a task can be operated in only one zone. It is required to define the information about which task can be assigned to which workstations. As an example, 'Assembly of Air Passage' task can only be assigned in 'Workstation 1' because of the tooling, location and environment of this workstation. As a result, there are some workstation and zone restrictions for tasks.

After all these analysis on current assembly line system of the MAN Türkiye A.Ş., the following symptom is observed: some tasks that are assigned to workstations are not completed within the cycle time. Because of this, delays occur on some workstations. In other words, tasks assigned to workstations are not balanced and current assembly line does not work efficiently.

3.1 CURRENT SOLUTION

First, it is better to see the whole picture for line balancing and model sequencing for MAN Türkiye A.Ş. Below sections shows the whole picture.

3.1.1 Current Balance (CB)

Current line balance and workstation (from G01 to G10) times for 3 models (A78, R07, R14) are as following. These station times (in terms of minutes) are determined by calculating maximum loaded workers in each workstation.

Table 16: Workloads - CB

	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10
A78	51.70	70.80	68.88	83.83	109.98	109.32	75.70	70.63	58.07	73.73
R07	75.97	80.68	68.45	80.47	86.53	79.57	69.67	72.45	91.03	68.38
R14	76.40	69.93	78.45	79.95	93.00	97.15	76.18	75.13	76.28	69.12

3.1.2 Current Balance – Adjusted (CBA)

Overload tolerance*:* In the current balance, there are a lot of station times that exceed fixed cycle time, which is 65 minutes and set by MAN Türkiye A.Ş. Because of this, an adjustment is required to rebalance these workstations. We follow Çevikcan et. al., (2009) for re-balancing. Before re-balance these stations, an overload tolerance is determined to remain in real life case as much as possible. In that direction, station times are allowed to exceed fixed cycle time within a tolerance, up to allowable station times. Tolerance ratio is taken as 20%. Allowable station time is calculated according to the below formulation:

CT: Cycle Time

$$\text{Allowable station time} = CT + (CT*Tolerance) \quad (4.0)$$

Then, allowable station time is found as follows:

CT= 65 min., Tolerance: 20%

Allowable station time: 65+(65*0,20)=78 min.

In the Table 17, red shaded cells show overloaded workstations that exceed allowable station time. All overloaded station times in current balance are adjusted by re-balancing task. This balancing is made by making little assignment changes in overloaded workstations. In example, for A78 in G04, last 3 task of most loaded zone are taken from this zone and assigned to another zone in another workstation that has most free time.

Table 17: Overloaded Workstations

	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10
A78	51.70	70.80	68.88	83.83	109.98	109.32	75.70	70.63	58.07	73.73
R07	75.97	80.68	68.45	80.47	86.53	79.57	69.67	72.45	91.03	68.38
R14	76.40	69.93	78.45	79.95	93.00	97.15	76.18	75.13	76.28	69.12

New adjusted station times are shown with green shaded cells in table below. For A78 in G05, station time still exceeds allowable station time, 78 min. this is because of the assignment restrictions on that workstation.

Table 18: Adjusted Workloads - CBA

	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10
A78	51.70	70.80	68.88	74.33	85.25	76.32	75.70	70.63	58.07	73.73
R07	75.97	77.23	68.45	75.95	77.83	72.23	69.67	72.45	77.92	68.38
R14	76.40	69.93	78.45	77.72	76.88	75.78	76.18	75.13	76.28	69.12

3.1.3 Current Sequence (CS)

Sequences are taken from 2011 master data (Demand data and production planning of buses of MAN Türkiye A.Ş. for 2011). The real sequence of May 2011 is taken as the current sequence (CS) for the following parts of the study. Also, demands of May 2011 are used for sequencing approaches.

Monthly Demands

Demands are taken from master data of company for the first seven months of 2011. They are shown in the following tables. In Table 20, demands of first seven months of 2011 for each bus model are given. As seen from the table, the most demanded bus model is R07 with 81 percent.

Table 19: Demands of each model

2011 - first 7 months		
A78	62	6%
R07	808	81%
R14	131	13%
Total	1001	100%

Table 20: Monthly Demands

	January		*February*		*March*		*April*		*May*		*June*		*July*	
	∑	%	∑	%	∑	%	∑	%	∑	%	∑	%	∑	%
A78	0	0%	0	0%	2	1%	24	17%	15	10%	2	1%	19	20%
R07	106	84%	108	85%	170	93%	87	63%	112	70%	158	92%	67	70%
R14	20	16%	19	15%	10	6%	28	20%	32	20%	12	7%	10	10%
Total	126	100%	127	100%	182	100%	139	100%	159	100%	172	100%	96	100%

3.2 PROPOSED SOLUTION

In this section proposed solution for the industrial application is presented. First, line balancing approaches then the model sequencing approaches with different scenarios are explained.

3.2.1 Line Balancing Approach

In this section, heuristic balance results are tabulated and illustrated by plots and commented on them.

3.2.1.1 Heuristic Balance (HB)

Line balancing model and heuristic algorithm are explained in Chapter 2. For each model of buses, station times are found by using heuristics, i.e. combined Construction_3 and improvement algorithms.

As stated before, in the current sub system, there are three models of buses and ten workstations. Task times are deterministic. Each model of buses has approximately 100 tasks.

Results of these algorithms are named as Heuristic balance (HB) and its results are shown in Table 21 below.

Table 21: Heuristic Balance (HB) Results

		G01	G02	G03	G04	G05	G06	G07	G08	G09	G10
A78	Load	57.68	65.27	64.12	60.67	67.72	66.90	67.93	65.48	64.98	64.73
	Improvement	*-10%*	*8%*	*7%*	*23%*	*26%*	*14%*	*11%*	*8%*	*-11%*	*14%*
R07	Load	63.59	65.15	59.05	61.95	62.92	56.01	64.50	64.93	74.37	66.08
	Improvement	*19%*	*19%*	*16%*	*23%*	*24%*	*29%*	*8%*	*12%*	*5%*	*3%*
R14	Load	59.10	67.23	67.93	62.87	71.40	60.29	60.35	62.18	69.37	57.08
	Improvement	*29%*	*4%*	*15%*	*24%*	*8%*	*26%*	*26%*	*21%*	*10%*	*21%*

As seen from the table, workloads are smoothened and decreased. Average workloads of each model of buses in minutes are determined as 64.55’ for A78, 63.86’ for R07 and 63.78’ for R14. Improvement ratios are determined after comparison with CBA values which are given in Table 18. These improvements are also shown in following Figures 20, 21 and 22. As seen from these figures, model A78 is the most smoothened model.

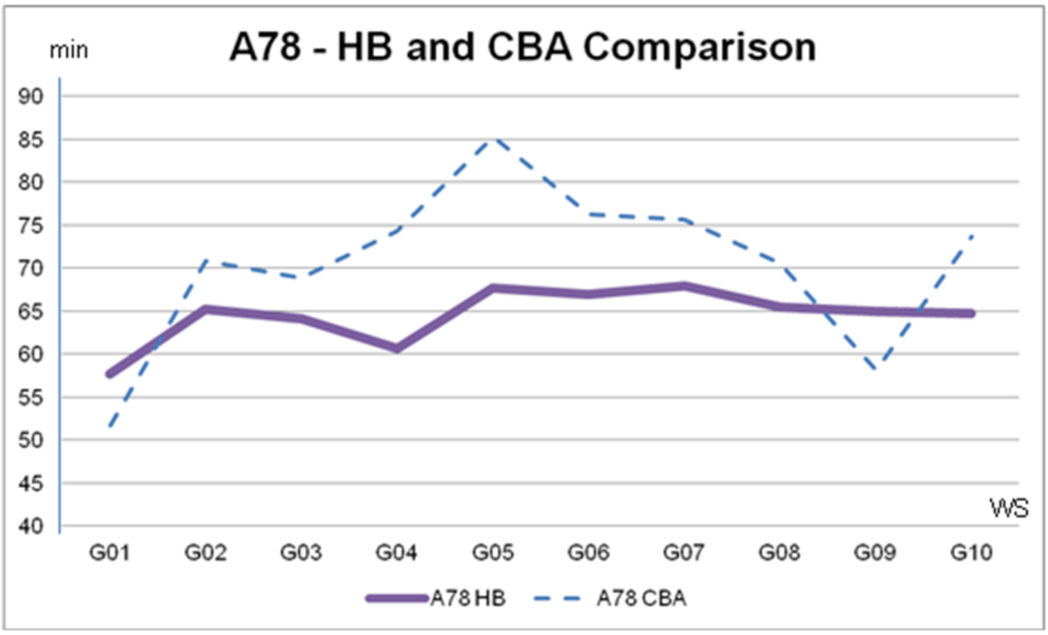


Figure 20: A78 - HB and CBA Comparison

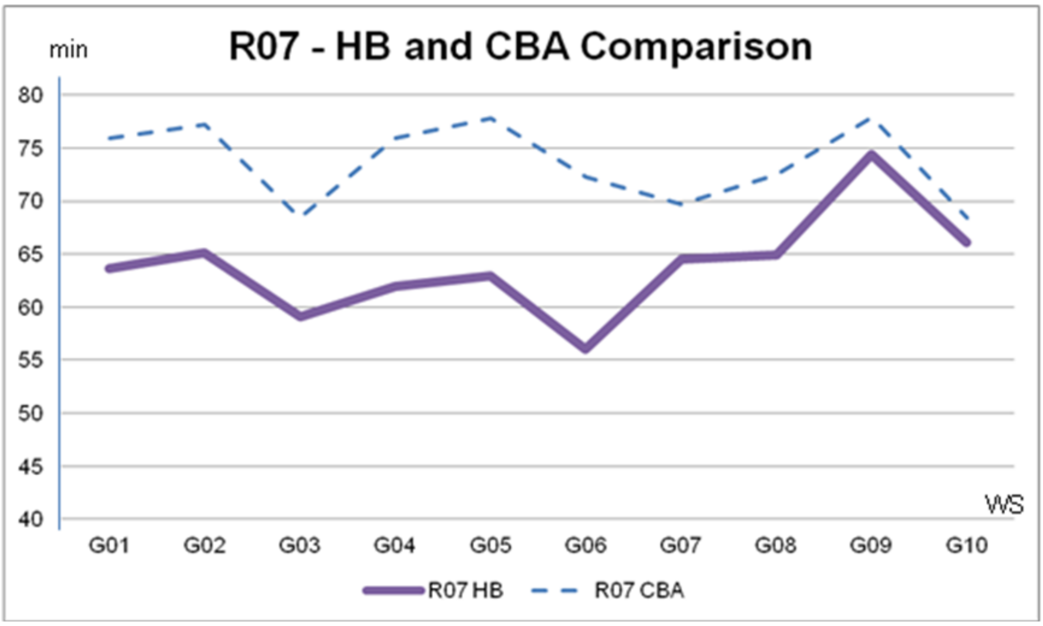


Figure 21: R07 - HB and CBA Comparison

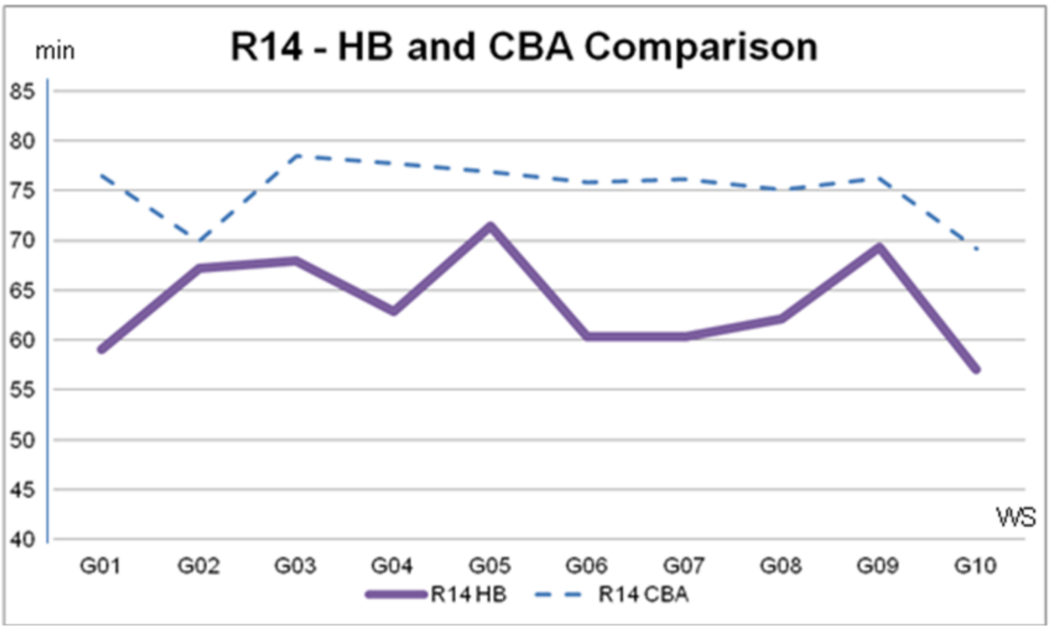


Figure 22: R14 - HB and CBA Comparison

3.2.2 Model Sequencing Approach

Here, as a proposed solution, model sequencing approach is presented for industrial application with different scenarios.

3.2.2.1 Optimal Sequence (OS)

Mixed model sequencing approach is explained in Chapter 2. As explained, two scenarios are determined as synchronous (s) and asynchronous (as). These scenarios are simulated in MS Excel and tardiness and lateness (sum of blocking and starving) times are determined. Screenshots of Excel simulation are given in Appendix C. Also, two sequencing models are solved in GAMS with station times which are taken from current balance (CB), current balance-adjusted (CBA) and heuristic balance (HB) results. Abbreviations of all variations are shown Table 20. Optimal model sequencing results are given in the following sections.

Table 22: Abbreviations of Sequencing Problems

Station times / Line Scenario	*Current Balance (CB)*	*Current Balance-Adjusted (CBA)*	*Heuristic Balance (HB)*
Synchronous Line	CBs	CBAs	HBs
Asynchronous Line	CBas	CBAas	HBas

3.3 RESULTS and COMPARISONS

Before the result part, problem abbreviations are explained. As told before, in this thesis, a real life problem is held and current line balance and sequences are compared with proposed solutions. Results are classified as shown in Figure 23, for being more readily understood.

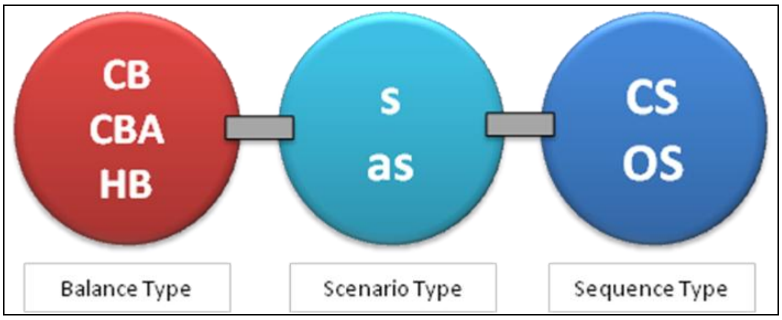


Figure 23: Problem classification

As shown in Figure 23, first part of problem name indicates balance type (CB, CBA or HB), in the middle, there is scenario type for sequencing problem (synchronous - s or asynchronous – a) and the last part of the name indicates sequence type (CS or OS).

3.3.1 Current Sequence (CS) Variations

As told before, May 2011 sequence of the firm is taken as Current Sequence (CS). In this part, the main aim is to show how avg. lateness and makespan values change when sequence is changed. In May 2011, there are 159 vehicles and 112 of them are R07, 32 are R14 and rest is A78. Adhering to these demand amounts, some sequence variations are created and these variations are simulated with station times of CB, CBA and HB. Variations and results are shown in Table 23.

Table 23: Current Balance (CB) - for Synchronous (s) and Asynchronous (as) scenarios

Current Balance (CB)		Synchronous (s)		Asynchronous (as)	
Sequence	Sequence property	Makespan	Avg. Tardiness	Makespan	Avg. Lateness
CS	**May 2011 Seq.**	**15045.68**	**17097.61**	**15045.68**	**1386.33**
OS	**Optimal sequence (OS)**	**14980.15**	**16278.70**	**14980.15**	**1067.80**
B1	112*R07-15*A78-32*R14	15151.27	17177.69	15151.27	1386.64
B2	112*R07-32*R14-15*A78	15118.53	16965.58	15118.53	1301.09
B3	32*R14-15*A78-112*R07	15619.85	18904.57	15619.85	1970.85
B4	32*R14-112*R07-15*A78	15074.72	17722.00	15074.72	1507.33
B5	15*A78-112*R07-32*R14	15104.60	18150.91	15104.60	1878.42
B6	15*A78-32*R14-112R07	15622.57	18459.72	15622.57	2068.77
Sce1	Repeat 7*R07 - 1*A78 - 2*R14	15050.22	17318.83	15050.22	1465.02
Sce2	Repeat 7*R07- 2*R14*- 1*A78	15021.23	17252.19	15021.23	1441.50
Sce3	Repeat 1*A78 - 2*R14-7*R07	15056.75	17176.55	15056.75	1475.41
Sce4	Repeat 1*A78-7*R07 - 2*R14	15024.58	17085.95	15024.58	1453.69
Sce5	Repeat 2*R14-7*R07-1*A78	15067.32	17337.36	15067.32	1462.20
Sce6	Repeat 2*R14-1*A78-7*R07	15072.05	17359.27	15072.05	1484.15

B1 to B6 indicate batch sequences. For example, in B1 sequence, firstly 112 R07 enter the line, then, 15 A78 follow them, and finally 32 R14 are processed. Scenarios (Sce1 to Sce6) are used to generate different type of sequences, like in Sce1; firstly 7 R07 enter, then 1 A78 and 2 R14 come, and this sequence repeats. In these sequences, model amounts are taken from the demand ratios for each model.

As seen from Table 23; when current balances are used for achieving optimal sequence, best average tardiness and lateness values and also makespan are given in synchronous and asynchronous scenarios. On the other hand, B1 sequence also gives better solution than current sequence.

Table 24: Heuristic Balance (HB)- for Synchronous (s) and Asynchronous (as) scenarios

Heuristic Balance (HB)		Synchronous (s)		Asynchronous (as)	
Scenario	Sequence property	Makespan	Avg. Tardiness	Makespan	Avg. Lateness
CS	**May 2011 Seq.**	**12093.10**	**1615.78**	**12093.10**	**883.39**
OS	**Optimal sequence (OS)**	**12032.4**	**1583.32**	**12032.40**	**844.22**
B1	112*R07-15*A78-32*R14	12083.73	1596.68	12083.73	896.46
B2	112*R07-32*R14-15*A78	12087.00	1692.10	12087.00	903.89
B3	32*R14-15*A78-112*R07	12227.11	2102.15	12227.11	951.91
B4	32*R14-112*R07-15*A78	12159.23	2057.60	12159.23	948.04
B5	15*A78-112*R07-32*R14	12142.71	1573.21	12142.71	912.64
B6	15*A78-32*R14-112R07	12201.64	1913.14	12201.64	924.50
Sce1	7*R07 - 1*A78 - 2*R14	12074.35	1527.14	12074.35	862.72
Sce2	7*R07- 2*R14*- 1*A78	12078.73	1567.48	12078.73	863.44
Sce3	1*A78 - 2*R14-7*R07	12096.02	1512.82	12096.02	876.49
Sce4	1*A78-7*R07 - 2*R14	12092.02	1539.66	12092.02	877.30
Sce5	2*R14-7*R07-1*A78	12103.03	1535.91	12103.03	877.16
Sce6	2*R14-1*A78-7*R07	12109.49	1591.45	12109.49	890.41

In Table 24; heuristic balance are used in several sequence types. The best average tardiness and lateness values and also makespan are not taken obtained by optimal sequence in synchronous scenario, because as told before, in synchronous scenario workloads of buses on each workstation should be more closer to pre-determined cycle time, 65 minutes. When heuristic balance results are used to get optimal sequence, average lateness and makespan values become worse than some other manually generated sequences, for example Sce3 and Sce1 in synchronous scenario.

In asynchronous scenario, optimal sequence which is heuristic balance results are used, gives the best average tardiness and lateness values and also makespan, because in this type of scenario, smoothed workloads decrease average lateness.

3.3.2 Finding Best Cycle Time for Synchronous (s) Lines

As told in Chapter 2, in synchronous scenario, tardiness is defined as difference between starting time and cumulative cycle time. So, the main aim is to decrease tardiness in synchronous lines. Up to know, cycle time is taken as 65 minutes for synchronous lines. In this part, a cycle time that decrease average lateness as much as possible is found for each CB, CBA and HB. Sequence is taken from Current sequence (CS) and workloads are taken from current balance (CB). Results are shown in Figure 24.

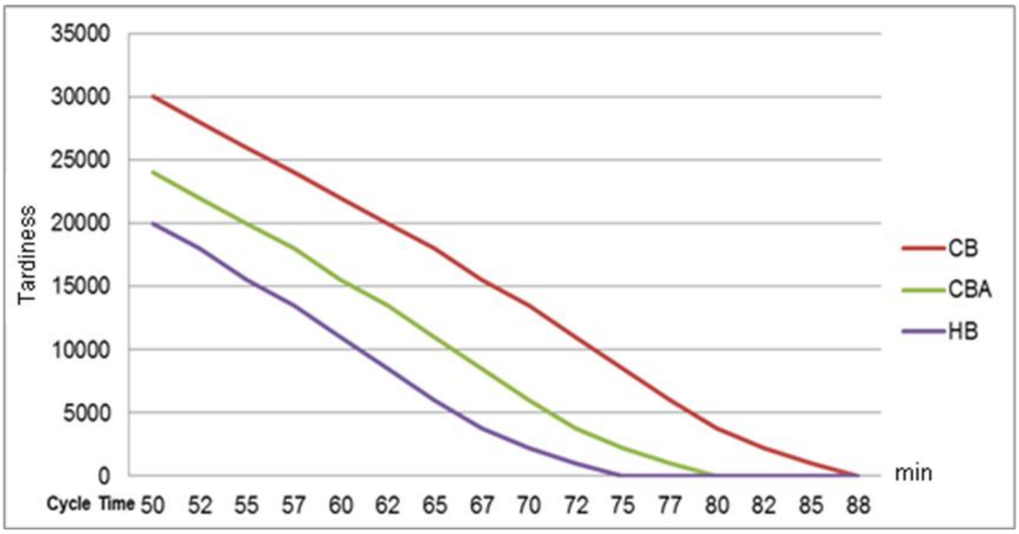


Figure 24: Best Cycle Time for Balance Types

As shown in figure, for current balance (CB), best cycle time (CT) value is 88 minutes, since there is no tardiness above that CT. For current balance-adjusted (CBA) best CT value is 80 minutes and for heuristic balance (HB) best CT value is 74 minutes. By this "Best CT" study, company can determine a realistic cycle time for pushing forward vehicle on the line synchronously. This also helps company to forecast accurate delivery dates.

3.3.3 Penalties

For synchronous and asynchronous scenarios, tardiness and lateness (sum of blocking and starving) times are determined, respectively. In the assembly line, for each scenario, model change in sequence brings additional tardiness or lateness (blocking plus starving) times in each workstation. We define the change in these tardiness and lateness times as **penalty**. As seen from the Table 25, for the scnhronous scenario, if a model A78 enters to G03 and previous model was A78, a tardiness value (7.77 min) is composed. To see how a change in sequence affects, penalties are calculated and after that a choice table is determined. In this choice table, "1" is for best avg. lateness value and "6" is for worst. If a cell contains "1", this means that this vehicle sequence brings best penalty. Penalties and choices are shown in Tables below. Sequence is taken from Current sequence (CS) and workloads are taken from current balance (CB).

Penalties for Synchronous Scneario

Table 25: Penalties for Synchronous Scneario

	G01			*G02*			*G03*			*G04*		
	A78	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*
A78	-26.60	-2.33	-1.90	11.60	21.48	10.73	7.77	7.33	17.33	37.67	34.30	33.78
R07	-2.33	21.93	22.37	21.48	31.37	20.62	7.33	6.90	16.90	34.30	30.93	30.42
R14	-1.90	22.37	22.80	10.73	20.62	9.87	17.33	16.90	26.90	33.78	30.42	29.90
	G05			*G06*			*G07*			*G08*		
	A78	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*
A78	89.97	66.52	72.98	88.63	58.88	76.47	21.40	15.37	21.88	11.27	13.08	15.77
R07	66.52	43.07	49.53	58.88	29.13	46.72	15.37	9.33	15.85	13.08	14.90	17.58
R14	72.98	49.53	56.00	76.47	46.72	64.30	21.88	15.85	22.37	15.77	17.58	20.27
	G09			*G10*								
	A78	*R07*	*R14*	*A78*	*R07*	*R14*						
A78	-13.87	19.10	4.35	17.47	12.12	12.85						
R07	19.10	52.07	37.32	12.12	6.77	7.50						
R14	4.35	37.32	22.57	12.85	7.50	8.23						

Table 26: Sum and Averages of Penalties for Synchronous Scneario

	Tardiness				*Average*		
Sum	*A78*	*R07*	*R14*	*Avg*	*A78*	*R07*	*R14*
A78	245.30	245.85	264.25	A78	24.53	24.59	26.43
R07	245.85	246.40	264.80	R07	24.59	24.64	26.48
R14	264.25	264.80	283.20	R14	26.43	26.48	28.32

Table 27: Choice Table for Synchronous Scneario

Choice	*A78*	*R07*	*R14*
A78	1	2	4
R07	2	3	5
R14	4	5	6

Penalties for Asynchronous Scneario

Table 28: Penalties for Asynchronous Scneario

	G01			*G02*			*G03*			*G04*		
	A78	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*
A78	0.00	-24.27	-24.70	0.00	-9.88	0.87	0.00	0.43	-9.57	0.00	3.37	3.88
R07	24.27	0.00	-0.43	9.88	0.00	10.75	-0.43	0.00	-10.00	-3.37	0.00	0.52
R14	24.70	0.43	0.00	-0.87	-10.75	0.00	9.57	10.00	0.00	-3.88	-0.52	0.00
	G05			*G06*			*G07*			*G08*		
	A78	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*	*A78*	*R07*	*R14*
A78	0.00	23.45	16.98	0.00	29.75	12.17	0.00	6.03	-0.48	0.00	-1.82	-4.50
R07	-23.45	0.00	-6.47	-29.75	0.00	-17.58	-6.03	0.00	-6.52	1.82	0.00	-2.68
R14	-16.98	6.47	0.00	-12.17	17.58	0.00	0.48	6.52	0.00	4.50	2.68	0.00
	G09			*G10*								
	A78	*R07*	*R14*	*A78*	*R07*	*R14*						
A78	0.00	-32.97	-18.22	0.00	5.35	4.62						
R07	32.97	0.00	14.75	-5.35	0.00	-0.73						
R14	18.22	-14.75	0.00	-4.62	0.73	0.00						

Table 29: Sum and Averages of Penalties for Asynchronous Scneario

	Blocking				*Starving*				*Average*		
Sum	*A78*	*R07*	*R14*	*Sum*	*A78*	*R07*	*R14*	*Avg*	*A78*	*R07*	*R14*
A78	0.00	68.38	38.52	*A78*	0.00	-68.93	-57.47	*A78*	0.00	-0.06	-1.90
R07	68.93	0.00	19.03	*R07*	-68.38	0.00	-44.42	*R07*	0.06	0.00	-1.84
R14	57.47	44.42	0.00	*R14*	-38.52	-26.02	0.00	*R14*	1.90	1.84	0.00

Table 30: Choice Table for Asynchronous Scenario

Choice	*A78*	*R07*	*R14*
A78		3	1
R07	4		2
R14	6	5	

As seen from the Table 30, in asynchronous scenario, while the workstations are changed, no blocking and starving times occur for the same workloads for same models.

3.3.4 Comparisons and Comments

As mentioned in Section 4.3, current line balance and sequences are compared with proposed solutions by using MS Excel. Results are classified as shown in Figure 23. Demands are taken from master data of May 2011. Results are compared for synchronous and asynchronous scenarios, respectively. All comparisons are made on average tardiness, lateness and makespan values.

3.3.4.1 Synchronous Scenario Results

Table 31: Line Balancing Approach Results (synchronous scenario)
a) Average Tardiness and Makespan Results Based on CB

	Avg. Tardiness	*makespan*
CB-s-CS	17097,61	15045,68
HB-s-CS	1615,78	12093,10
Improvement	90,55%	19,62%

b) Average Tardiness and Makespan Results Based on CBA

	Avg. Tardiness	*makespan*
CBA-s-CS	10324,65	13125,97
HB-s-CS	1615,78	12093,10
Improvement	84,35%	7,87%

Table 31 includes two sub-tables “a” and “b”, “a” shows that how line balancing heuristic algorithm treats on current sequence and improvement ratio between current balances and heuristic algorithm. The other one shows improvement ratio between current balances adjusted balances and heuristic algorithm. According to the table, it can be said that on synchronous lines, line balancing on current sequence gives better results.

Table 32: Sequencing Approach Results (synchronous scenario)
a) Average Tardiness and Makespan Results Based on CB

	Avg. Tardiness	*makespan*
CB-s-CS	17097,61	15045,68
CB-s-OS	16278,70	14980,15
Improvement	4,79%	0,44%

b) Average Tardiness and Makespan Results Based on CB

	Avg. Tardiness	*makespan*
CBA-s-CS	10324,65	13125,97
CBA-s-OS	10115,61	13076,45
Improvement	2,02%	0,38%

Table 32 presents that improvement ratio between current balance with current sequence and current balance with optimal sequence. The value of ratios shows that only performing sequencing on current balance does not affect as much as only performing heuristic balance.

Table 33: Hierarchical Solution (Line balancing>Sequencing) Results (synchronous scenario)
a) Average Tardiness and Makespan Results Based on CB

	Avg. Tardiness	*makespan*
CB-s-OS	16278,70	14980,15
HB-s-OS	1583,32	12032,40
Improvement	90,27%	19,68%

b) Average Tardiness and Makespan Results Based on CBA

	Avg. Tardiness	*makespan*
CBA-s-OS	10115,61	13076,45
HB-s-OS	1583,32	12032,40
Improvement	84,35%	7,98%

c) Average Tardiness and Makespan Results Based on HB

	Avg. Tardiness	*makespan*
HB-s-CS	1615,78	12093,10
HB-s-OS	1583,32	12032,40
Improvement	2,01%	0,50%

Table 33 includes three sub tables “a”, “b” and “c”. First one shows improvement ratio between current balance with optimal sequence and heuristic balance with optimal sequence. Second one presents improvement ratio between current balance adjusted with optimal sequence and heuristic balance with optimal sequence. Improvement ratio between heuristic balance with current sequence and heuristic balance with optimal sequence can be seen from third one. Also from these tables, it can be seen that, the value of ratios shows that only performing on heuristic algorithm provides more improvement than sequencing approach.

3.3.4.2 Asynchronous Scenario Results

Table 34: Line Balancing Approach Results (asynchronous scenario)

a) Average Tardiness and Makespan Results Based on CB

	Avg. Lateness	*makespan*
CB-as-CS	1386,33	15045,68
HB-as-CS	883,39	12093,10
Improvement	36,28%	19,62%

b) Average Tardiness and Makespan Results Based on CBA

	Avg. Lateness	*makespan*
CBA-as-CS	368,45	13125,97
HB-as-CS	883,39	12093,10
Improvement	-139,76%	7,87%

For asynchronous assembly lines, same comparisons with synchronous assembly lines are performed. Table 34 shows the improvement ratios when just heuristic balance is made. There are two sub tables, “a” and “b”. In second one there is no improvement comes from heuristic balance. Because, the values shown in Table 13 and 14 show that current balance adjusted values are more smoothed than heuristic balance.

Table 35: Sequencing Approach Results (asynchronous scenario)

a) Average Tardiness and Makespan Results Based on CB

	Avg. Lateness	*makespan*
CB-as-CS	1386,33	15045,68
CB-as-OS	1067,80	14980,15
Improvement	22,98%	0,44%

b) Average Tardiness and Makespan Results Based on CBA

	Avg. Lateness	*makespan*
CBA-as-CS	368,45	13125,97
CBA-as-OS	108,65	13076,45
Improvement	70,51%	0,38%

Table 35 shows improvement ratios when only sequencing approach is applied on current balance on asynchronous assembly lines. When compared to the synchronous assembly lines, sequencing approach gives better results.

Table 36: Hierarchical Solution (Line balancing>Sequencing) Results (asynchronous scenario)

a) Average Tardiness and Makespan Results Based on CB

	Avg. Lateness	*makespan*
CB-as-OS	1067,80	14980,15
HB-as-OS	844,22	11975,45
Improvement	20,94%	20,06%

b) Average Tardiness and Makespan Results Based on CBA

	Avg. Lateness	*makespan*
CBA-as-OS	108,65	13076,45
HB-as-OS	844,22	12032,4
Improvement	-677,01%	7,98%

c) Average Tardiness and Makespan Results Based on HB

	Avg. Lateness	*makespan*
HB-as-CS	883,39	12093,10
HB-as-OS	844,22	12032,4
Improvement	4,43%	0,50%

Table 36 shows that heuristic balance on optimal sequence gives better results in asynchronous assembly lines. In this scenario, hierarchical solution (heuristic balance first, than sequencing) gives better improvement (4,43%) than synchronous scenario (2,01%) for avg. lateness.

3.4 SUMMARY

In the scope of industrial application, firstly it is seen that current balance should be re-balanced to get more realistic solutions. Then, in the light of study that held by Cevikcan (2009), adjusted current balance is constructed. For the current sequence, data of 2011 May, which is basis for our study, is investigated.

After that, results of solution approaches for the line balancing and sequencing problems are presented in Section 4.2. In this section, firstly, line balancing approach results are presented. According to this, as reffered in Chapter 3, Construction_3 and improvement heuristic algorithms are used, and for each model of products, balances are taken. Based on HB results, workloads are smoothened and decreased for each product model.

By using model sequencing approaches for synchronous and asynchronous scenarios, optimal sequence results are compared with current sequence and manually generated sequences.

Moreover, in synchronous scenario, by using the "Best CT" study, a tool is presented to company which is developed for determining a realistic cycle time. Also, penalties approach as a performance measure for the model sequencing is presented. In the assembly line, for each scenario, to see how a model change in sequence affects avg. lateness, penalties are calculated and after that a choice table is determined.

At last, comparisons for synchronous and asynchronous assembly lines based on line balancing approach and sequencing approach are performed by using real life data. According to the results in synchronous assembly lines, line balancing is much more important than sequencing, because it is important to decrease workloads to cycle time by using balancing approach.

Line balancing is important in due to the requirement of smoothed workstations, asynchronous assembly lines. Also sequencing bus models has an important role to prevent starving and blocking times.

CONCLUSION

In this thesis study, a single model assembly line balancing problem with multi manned workstations and model sequencing problem are studied. For the line balancing problem, the aim is to find minimum cycle time under given number of workstations. For the model sequencing aim is to find better sequence to overcome tardiness or lateness at a large number of workstations for the given model mix.

To the best of our knowledge, the study is original with the side of problem definitions and industrial application. For the line balancing problem, mixed integer linear mathematical formulations are developed. However, due to the combinatorial nature of the problem, the mathematical formulations do not give even any feasible solution in a reasonable time. Therefore, priority based heuristic approaches are developed. To make performance analysis on our heuristic, new test problems are generated based comparison results for the test problems are presented. Small, medium and large sized instances (in total 60 instances) for optimal seeking and proposed heuristic algorithms are solved for line balancing problem. Results of these instances show that in reasonable time, as the problem size increases, getting optimal solution becomes impossible. But algorithms “Construction 3” and “Construction 3 with improvement” has better effect on the solution of instances than other proposed heuristic algorithms.

For the model sequencing problem two different mixed integer linear programming mathematical formulations are developed for synchronous and asynchronous assembly lines.

In addition to these, industrial application is presented. Industrial application is solved for two different scenarios (synchronous and asynchronous). First, current system is illustrated for line balancing and model sequencing. Then, for the scenarios, solution approaches first solved separately, then together, and lastly, all separately and hierarchal solutions are compared. Moreover, new tools (like penalty and best CT) are presented for the practical usage in the industry.

As further research direction, for line balancing problem, more zoning constraints, assignment constraints can be included. For example, in real life, a task covers more than one zone at a workstation. Also, new objective functions by considering cost can be developed.

As another further research direction is using a new objective function by considering delivery times of product can be developed for model sequencing problem. To get more realistic solutions, instead of deterministic demands, stochastic demands can be included.

Moreover, line balancing and model sequencing problems can be solved simultaneously. For each problem, more construction and improvement heuristics can be developed; also solution approaches with meta-heuristics can create new research areas in literature.

REFERENCES

1. Bard, J.F., Dar-elj, E., Shtub, A., 1992, An analytic framework for sequencing mixed model assembly lines, International Journal of Production Research, Volume 30, Issue 1, pp:35-48.

2. Bartholdi, J.J., 1993, Balancing Two Sided Assembly Lines: A Case Study, International Journal of Production Research, Vol.31, Pages 2447- 2461.

3. Baybars, I., 1986., A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem, Management Science, Vol. 32, Pages 909-932. Becker, C., Scholl, A., 2006. A Survey on Problems and Methods in Generalized Assembly Line Balancing. European Journal of Advanced Manufacturing Technology, Vol. 36, Pages 582-588.

4. Becker, C., Scholl, A., 2009. Balancing Assembly Lines with Variable Parallel Workplaces: Problem Definition and Effective Solution Procedure, European Journal of Operations Research, Vol.199, Pages 359-374.

5. Boysen, N., Fliedner, M., Scholl, A., 2007. A Classification of Assembly Line Balancing Problems, European Journal of Operational Research, Vol. 183/2, Pages 674-693.

6. Boysen, N., Fliedner, M., Scholl, A., 2009. Sequencing Mixed-Model 5. Assembly Lines: Survey, Classification and Model Critique, European Journal of Operations Research, Vol. 192/2 , Pages 349-373.

7. Cevikcan, E., Durmusoglu, M.B., Unal, M.E., 2009, A Team-Oriented Design Methodology for Mixed Model Assembly Systems, Computers and Industrial Engineering, Vol.56, Pages, 576-599.

8. Dar-El, E. M., Cother, R. F., 1975, Assembly line sequencing for model mix. International Journal of Production Research, 13, 463-477.

9. Dar-el, E. M., Nadivi, A., 1981, A mixed model sequencing application, International Journal of Production Research Volume 19, Issue 1, pages 69-84.

10. Dimitriadis, S.G., 2006, Assembly Line Balancing and Group Working: A roups Operating on the Same Product and Workstation, Computers and Operations Research, Vol.33, Pages 2757-2774.

11. Fattahi, P., Salehi, M., Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval ". Int J Adv Manuf Technol, vol.12, 2009.

12. Garey, M.R. and Johnson, D.S., 1979, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freemann, San Francisco.

13. Gurevsky, E., Battaïa, O., Dolgui, A., 2013, "Stability measure for a generalized assembly line balancing problem". Discrete Applied Mathematics Vol.161, pp.377–394.

14. Inman, R. R., Bulfin, R. L., 1991, "Sequencing JIT Mixed-Model Assembly.

15. Karabak, F, 2012. "Workforce Assignment in a Multi-Worker Multi-Sided Mixed Model Assembly Line Balancing Problem", Master Thesis.

16. Karabatı, Sayın, 2003, Assembly Line Balancing in a Mixed Model Sequencing Environment with Synchronous Transfers, European Journal of Operational Research, Vol. 149, Pages 417-429.

17. Kellegöz T., Toklu B., 2012, An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations, Computers & Operations Research, Volume 39, Issue 12, Pages 3344-3360.

18. Kim S., Jeong B., "Product sequencing problem in Mixed-Model Assembly Line to minimize unfinished works ".Computers & Industrial Engineering, vol. 53, 2007, pp. 206–214.

19. 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 Vol.36, pp.853–865.

20. Miltenburg, J., Sinnamon, G., 1989, "Scheduling Mixed-model Multi-level Just-in-time Production Systems", International Journal of Production Research, 27, 1487-1509.

21. Mirzapour Al-e-Hashem, S.M.J ., Aryanezhad, M.B ., "An Efficient Method to Solve a Mixed-model Assembly Line Sequencing Problem Considering a Sub-line . " World Applied Sciences Journal, vol. 6, 2009, pp. 168-181.

22. Özbakir, L., Tapkan, P., (2011) "Bee colony intelligence in zone constrained two-sided assembly line balancing problem".

23. Özcan 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).

24. Özcan U, Gökçen H., Toklu B. (2010). Balancing parallel two-sided assembly lines. International Journal of Production Research, 48(16)(4767- 4784).

25. Özcan U., Toklu B., 2009. Balancing of Mixed Model Two Sided Assembly Lines, Computers and Industrial Engineering, Vol.57, Pages 217-227.

26. Purnomo, H.D., Wee, H., Rau, H., 2013, Two-sided assembly lines balancing with assignment restrictions. Mathematical and Computer Modelling 57(1-2): 189-199.

27. Roshania, A., Roshani, A., Roshani, A., Salehi, M., Esfandyari, A., (2013) “A simulated annealing algorithm for multi-manned assembly line balancing problem”. Journal of Manufacturing Systems Vol.32, pp. 238– 247.

28. Salveson, M.E., 1955, The Assembly Line Balancing Problem. Journal of Industrial Engineering, Vol. 6(3), 18-25.

29. Scholl, A., 1999. Balancing and Sequencing of Assembly Lines, Physica- Verlag.

30. Tapkan, P., Ozbakir, L., Baykasoglu, A., 2012, “Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms”. Applied Soft Computing Vol.12, pp.3343–3355.

31. Thompoulos, N.T., 1967, Line Balancing-Sequencing for Mixed-Model Assembly. Management Science, Vol.41, Pages B59-B75.

32. Thompoulos, N.T., 1970, Mixed Model Line Balancing with Smoothed station Assignments. Management Science, Vol.16, Pages 593-603.

33. Yano C., Rachamadugu R., 1991, Sequencing to minimize work overload in assembly lines with product options, Management Science, Volume 37 Issue 5, Pages 572-586.

APPENDICES

APPENDIX A.

EXAMPLE GAMS CODE FOR LINE BALANCING PROBLEM

```
set
i/a1*a30/
s/1*4/
j/s1*s3/

alias(i,ii);
alias(j,jj) ;
alias(s,ss)  ;

Scalar
V/9999/;


parameter
TIME(i) /
a1      11
a2      7
a3      6
a4      19
a5      15
a6      13
a7      11
a8      7
a9      20
a10      17
a11      6
```

a12 19

a13 18

a14 15

a15 5

a16 15

a17 18

a18 16

a19 17

a20 12

a21 10

a22 11

a23 20

a24 18

a25 19

a26 16

a27 20

a28 8

a29 5

a30 15

/ ;

Table wQUAL(j,s,i)

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28	a29	a30
s1.1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
s1.2	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0
s1.3	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1

s1.4	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0
s2.1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
s2.2	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0
s2.3	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1
s2.4	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0
s3.1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
s3.2	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0
s3.3	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1
s3.4	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0

Table PRE(i,ii)

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28	a29	a30
a1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a6	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
a17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

18

a19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
a20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
a22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
a23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
a24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
a25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
a27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
a28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
a29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

;

binary variables

WASG(i,s,j)

Z(i,ii,s,j)

```
;

positive variable

B(i)
CT
;

variable
zz
;

equations

obj
c1(i)
c11
c2(i,ii)
c3(i,ii,s,ss,j,jj)
c4(i,ii,s,j)
c5(s,j)
c51
c6(ii,s,j)
c7(i,s,j)
c8(i,ii)
c9(i)
;

obj..zz=e=CT;

c1(i)..sum((s,j),wQUAL(j,s,i)*WASG(i,s,j))=e=1;

c11(i)..sum((s,j),WASG(i,s,j))=e=1;
```

```
c2(i,ii)$(PRE(i,ii) eq 1)..sum((s,jj),ord(jj)*(WASG(ii,s,jj)))-
sum((s,j),ord(j)*(WASG(i,s,j)))=l=0;

c3(i,ii,s,ss,j,jj)$(PRE(i,ii) eq 0)..B(ii)+V*(3-WASG(i,s,j)-WASG(ii,ss,jj)-
Z(i,ii,s,j))=g=B(i)+TIME(i);

c4(i,ii,s,j)..Z(i,ii,s,j)+Z(ii,i,s,j)=l=1;

c5(s,j)..sum(i,WASG(i,s,j))-1=e=sum(i,sum(ii,Z(i,ii,s,j)));

c51(i,s,j)..2*WASG(i,s,j)=g=sum(ii,Z(i,ii,s,j))+sum(ii,Z(ii,i,s,j));

c6(ii,s,j)..sum(i,Z(i,ii,s,j))=l=1;

c7(i,s,j)..sum(ii,Z(i,ii,s,j))=l=1;

c8(i,ii)$(PRE(i,ii) eq 1)..B(ii)=g=B(i)+TIME(i);

c9(i)..CT=g=B(i)+TIME(i);


Model PR1_NDG_130310 /all/;
option reslim=7200;
option optca=0;
option optcr=0;
Solve PR1_NDG_130310 minimizing zz using MIP;
display wasg.l, z.l;
```

APPENDIX B.

EXAMPLE GAMS CODE FOR SEQUENCING PROBLEM

Synchronous Scenario:

```
set
k/G01,G02,G03/
i/m1,m2,m3,m4,m5/
t/1*50/

scalar
CT/65/;

parameters

d(i)/
m1    10
m2    10
m3    10
m4    10
m5    10
/



table P(i,k)

      G01    G02    G03
m1     52     71     69
m2     76     81     68
m3     76     70     78
```

m4
m5
;

binary variables

X(i,t)
**i isi t pozisyonuna atandıysa 1
;

positive variable

SS(k,t)
**k istasyonunda t pozisyonunun başlama zamanı
W(k,t)
**k istasyonunda t pozisyonundaki gecikme
PP(k,t)
**k istasyonunda t pozisyonunun iş süresi
;

variable
z
;

equations

obj
c1
c2
c3
c4

```
c6
c61
c62
c63
*c64
*c7
*c71
;

obj..z=e=sum(k,sum(t,W(k,t)));

c1(i)..sum(t,X(i,t))=g=d(i);

c2(t)..sum(i,X(i,t))=e=1;

c3(k,t)..PP(k,t)=e=sum(i,P(i,k)*X(i,t));

c4(k,t)..SS(k,t)-CT*(ord(t)-1)=l=W(k,t);

c6(k,t)$(ord(t) > 1).. SS(k,t)=g=PP(k,t-1)+SS(k,t-1) ;
c61(k,t)$(ord(k) > 1).. SS(k,t)=g=PP(k-1,t)+SS(k-1,t) ;
c62(k,t)..SS("G01","1")=e=0;
c63(k,t)$(ord(k) > 1).. SS(k,"1")=g=SS(k-1,"1")+PP(k-1,"1");
*c64(k,t)$(ord(t) > 1 and ord(k) > 1)..SS("G01",t)=g=SS(k,t-1);


Model Seq_NDG_120429 /all/;
option optca=0 ;
option optcr=0  ;

Solve Seq_NDG_120429 minimizing z using MIP;
Display X.l,W.l,SS.l,PP.l
```

Asynchronous Scenario:

```
set
k/G01,G02,G03/
i/m1,m2,m3,m4,m5/
t/1*50/

scalar
CT/65/;

parameters

d(i)/
m1     10
m2     10
m3     10
m4     10
m5     10
/



table P(i,k)

        G01    G02    G03
m1       52     71     69
m2       76     81     68
m3       76     70     78
m4
m5
;

binary variables
```

X(i,t)
**i isi t pozisyonuna atandıysa 1
;

positive variable

SS(k,t)
**k istasyonunda t pozisyonunun başlama zamanı
B(k,t)
**k istasyonunda t pozisyonundaki block
ST(k,t)
**k istasyonunda t pozisyonundaki starve
PP(k,t)
**k istasyonunda t pozisyonunun iş süresi
;

variable
z
;

equations

obj
c1
c2
c3
c4
c41
c42
c5
c51
c52

```
c6
c61
c62
c63
*c64
*c7
*c71
;

obj..z=e=sum(k,sum(t,B(k,t)))+sum(k,sum(t,ST(k,t)));

c1(i)..sum(t,X(i,t))=g=d(i);

c2(t)..sum(i,X(i,t))=e=1;

c3(k,t)..PP(k,t)=e=sum(i,P(i,k)*X(i,t));



c4(k,t)$(ord(k) > 1)..SS(k,t)-SS(k-1,t)-PP(k-1,t)=l=B(k,t);
c41(k,t)..B(k,"1")=e=0;
c42(k,t)..B("G01",t)=e=0;



c5(k,t)$(ord(t) > 1)..SS(k,t)-PP(k,t-1)-SS(k,t-1)=l=ST(k,t);
c51(k,t)..ST(k,"1")=e=0;
c52(k,t)..ST("G01",t)=e=0;



c6(k,t)$(ord(t) > 1).. SS(k,t)=g=PP(k,t-1)+SS(k,t-1) ;
c61(k,t)$(ord(k) > 1).. SS(k,t)=g=PP(k-1,t)+SS(k-1,t) ;
c62(k,t)..SS("G01","1")=e=0;
c63(k,t)$(ord(k) > 1).. SS(k,"1")=g=SS(k-1,"1")+PP(k-1,"1");
*c64(k,t)$(ord(t) > 1 and ord(k) < 2)..SS(k,t)=e=SS(k+1,t-1);
```

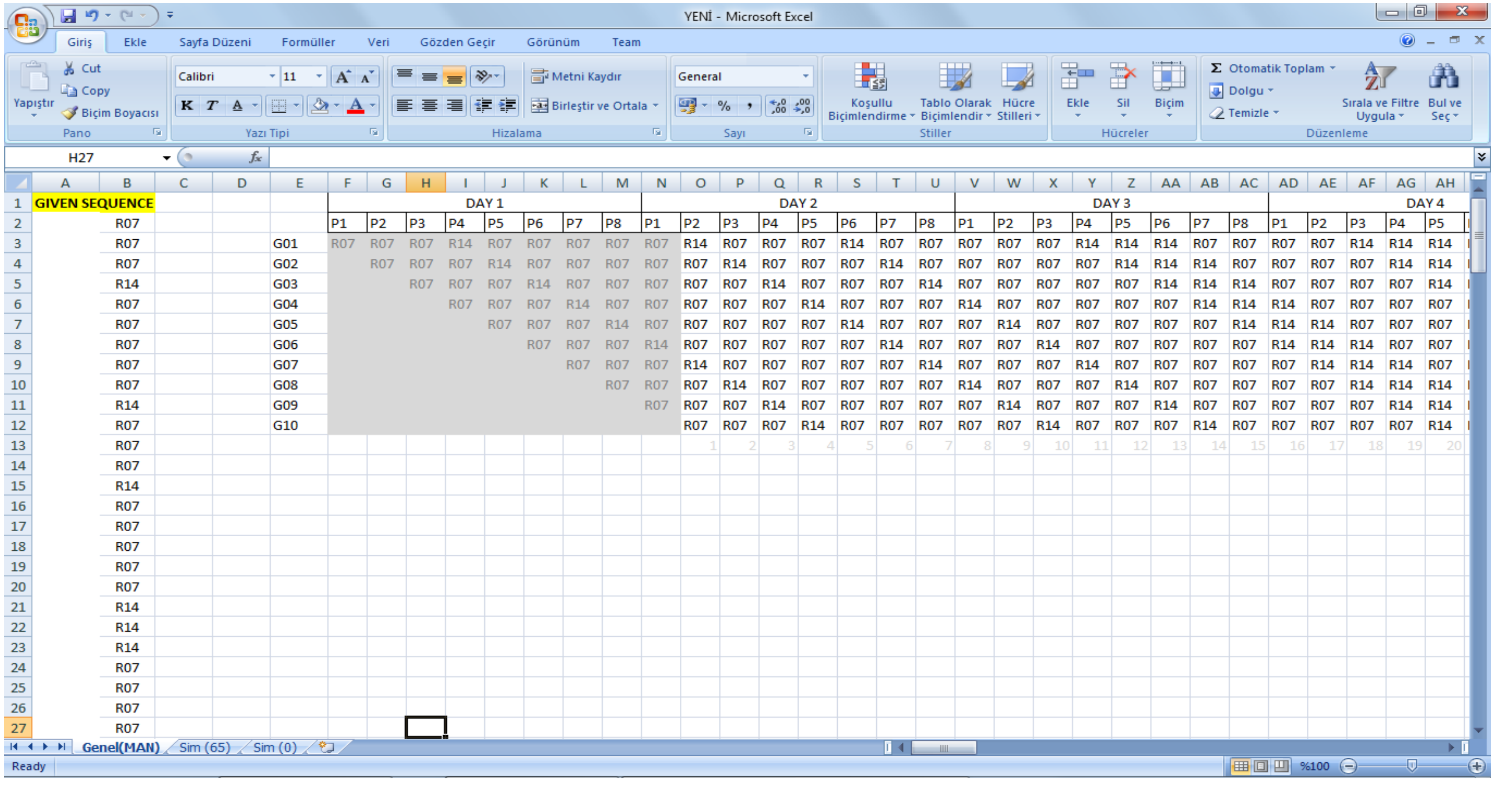
```
Model Seq_NDG_120429 /all/;
option optca=0 ;
option optcr=0  ;


Solve Seq_NDG_120429 minimizing z using MIP;
Display X.l,B.l,ST.l,SS.l,PP.l
```

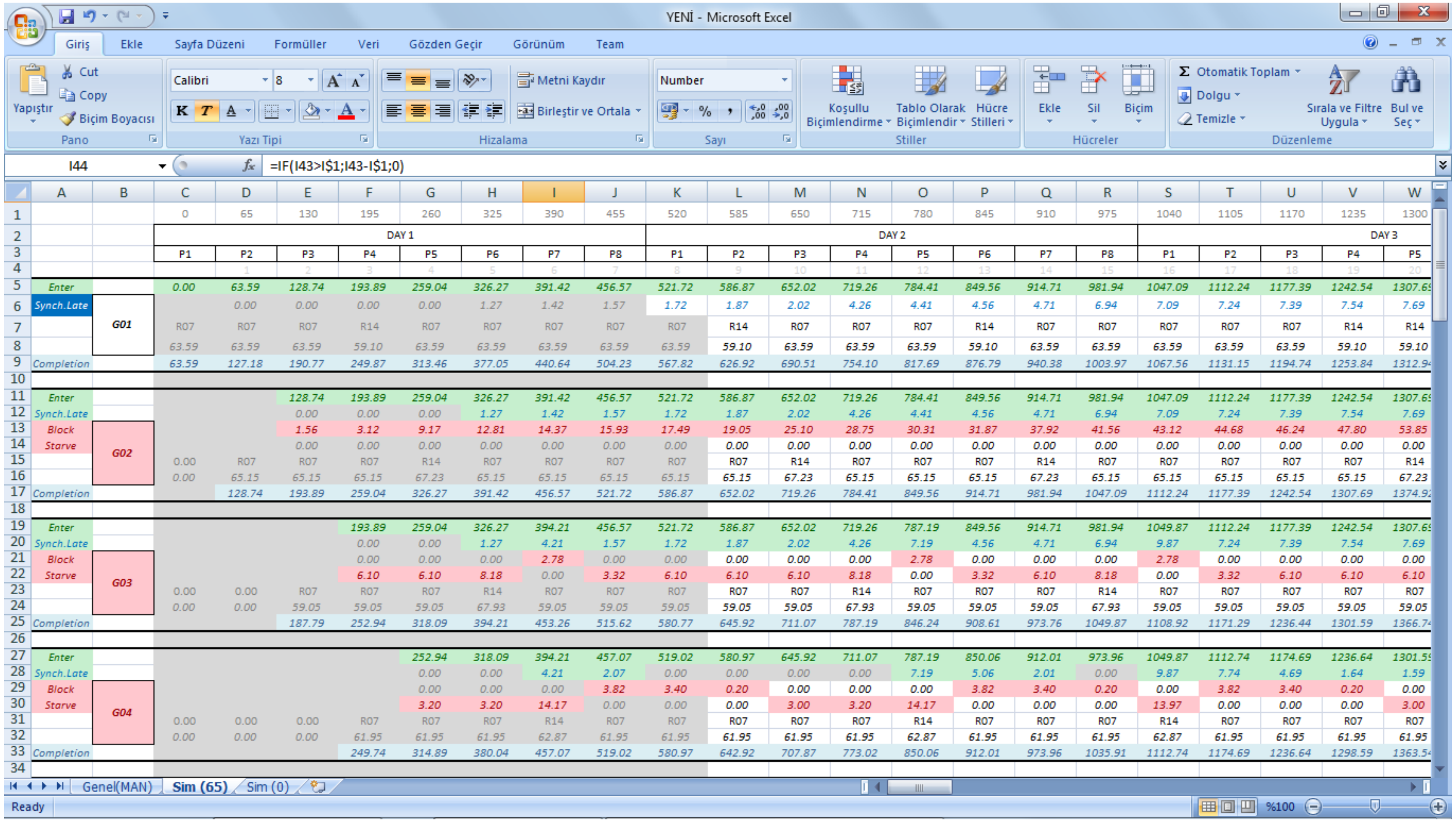
APPENDIX C.

SCREENSHOTS OF EXCEL SIMULATION

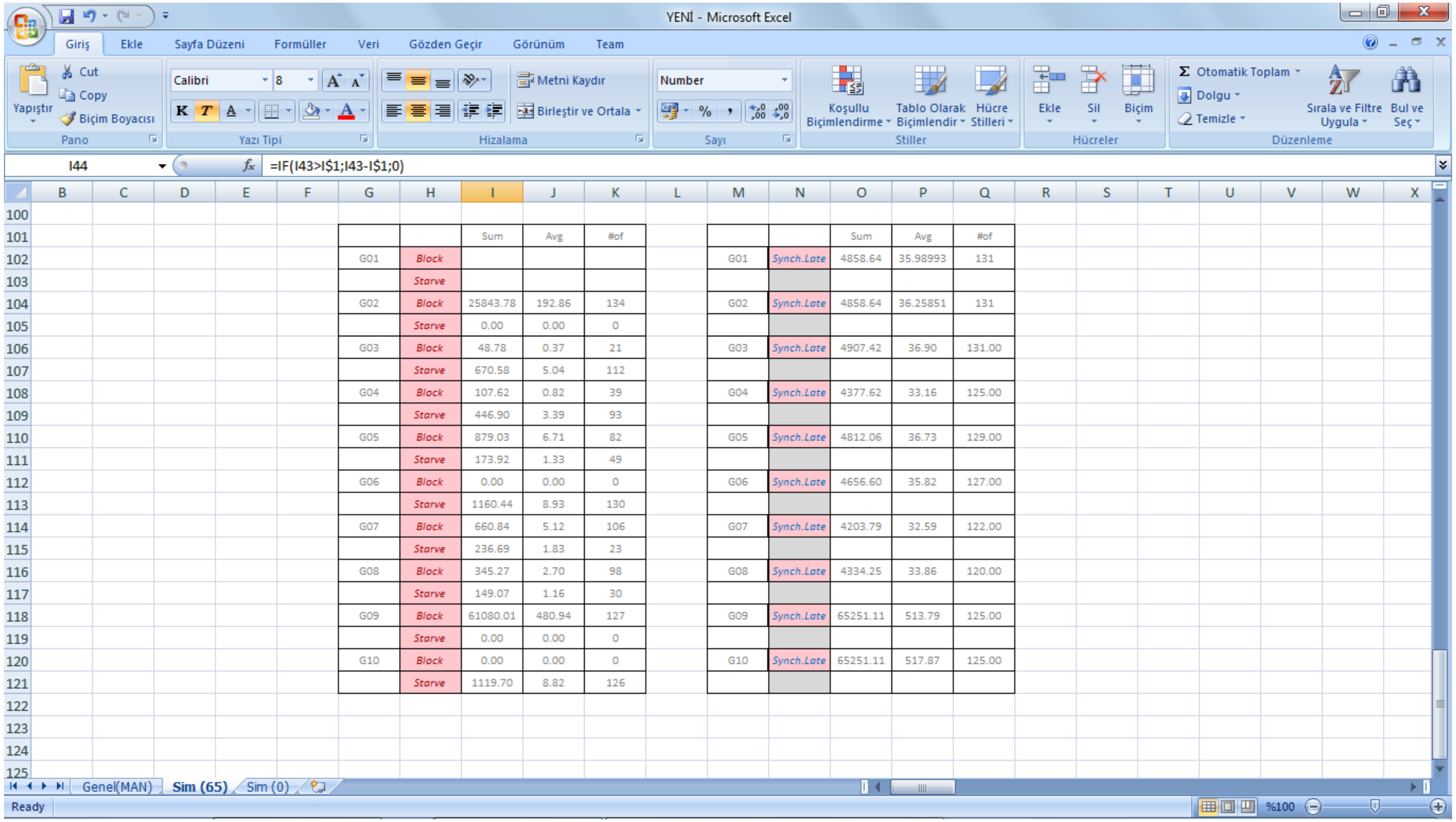
1. Enter the Given Sequence Screen



2. Simulation Screen



3. Simulation Results Screen



		Sum	Avg	#of
G01	Block			
	Starve			
G02	Block	25843.78	192.86	134
	Starve	0.00	0.00	0
G03	Block	48.78	0.37	21
	Starve	670.58	5.04	112
G04	Block	107.62	0.82	39
	Starve	446.90	3.39	93
G05	Block	879.03	6.71	82
	Starve	173.92	1.33	49
G06	Block	0.00	0.00	0
	Starve	1160.44	8.93	130
G07	Block	660.84	5.12	106
	Starve	236.69	1.83	23
G08	Block	345.27	2.70	98
	Starve	149.07	1.16	30
G09	Block	61080.01	480.94	127
	Starve	0.00	0.00	0
G10	Block	0.00	0.00	0
	Starve	1119.70	8.82	126

		Sum	Avg	#of
G01	Synch.Late	4858.64	35.98993	131
G02	Synch.Late	4858.64	36.25851	131
G03	Synch.Late	4907.42	36.90	131.00
G04	Synch.Late	4377.62	33.16	125.00
G05	Synch.Late	4812.06	36.73	129.00
G06	Synch.Late	4656.60	35.82	127.00
G07	Synch.Late	4203.79	32.59	122.00
G08	Synch.Late	4334.25	33.86	120.00
G09	Synch.Late	65251.11	513.79	125.00
G10	Synch.Late	65251.11	517.87	125.00

APPENDIX D.
CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: GÜNER, Nizamettin Doğan
Nationality: Turkish (TC)
Date and Place of Birth: 24 October 1988, Gümüşhane/Şiran
Marital Status: Married
Phone: +90 537 633 26 76
email: gunerd@ekonomi.gov.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Çankaya Univ. Industrial Engineering Program	2013
BS	Çankaya Univ. Industrial Engineering	2010
High School	Ankara Kurtuluş High School	2005

WORK EXPERIENCE

Year	Place	Enrollment
2013 Jan.- Present	Republic of Turkey, Ministry of Economy	Assistant Foreign Trade Expert

FOREIGN LANGUAGES

Advanced English

HOBBIES

Reading book and traveling. Interests: Political History, Islamic Architecture & Urban Design, Digital Design