

**DOKUZ EYLUL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

**BALANCING MIXED MODEL ASSEMBLY
LINES IN AN AUTOMOTIVE SUPPLIER BY
RECONFIGURING LAYOUT**

by

F. Sümeyra MERCAN

January, 2012

İZMİR

**BALANCING MIXED MODEL ASSEMBLY
LINES IN AN AUTOMOTIVE SUPPLIER BY
RECONFIGURING LAYOUT**

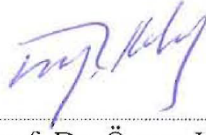
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylul University
In Partial Fulfillment of the Requirements for the Degree of Master of
Science in Industrial Engineering, Industrial Engineering Program**

**by
F. Sümeyra MERCAN**

**January, 2012
İZMİR**

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**BALANCING MIXED MODEL ASSEMBLY LINES IN AN AUTOMOTIVE SUPPLIER BY RECONFIGURING LAYOUT**” completed by **F. SÜMEYRA MERCAN** under supervision of **ASSIST. PROF. DR. ÖZCAN KILINÇCI** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



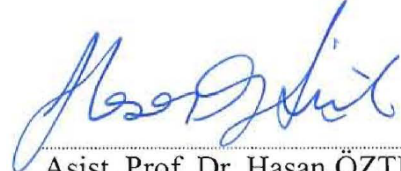
Asist. Prof. Dr. Özcan KILINÇCI

Supervisor



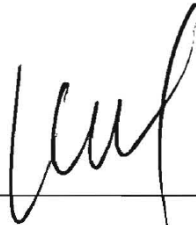
Asist. Prof. Dr. Gökalg YILDIZ

(Jury Member)



Asist. Prof. Dr. Hasan ÖZTÜRK

(Jury Member)



Prof. Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank my respectable supervisor assistant professor doctor Özcan Kılınçcı, who does not withhold his paramount patient and support in implementation and evaluation of outcomes of my thesis subjected as **“Balancing Mixed Model Assembly Lines in an Automotive Supplier by Reconfiguring Layout”**.

I would like to thank my precious friend Nimet T.KAHRAMAN who does not avoid her labor and support.

I would like to thank all MARTUR AŞ. Managers and Employees for their contribution.

Finally, I would like to express my deepest thanks to my extended family. Thanks my father İsmail MERCAN and my mother Fatma MERCAN who give me material and moral support even the most difficult times without once complaining and support me in this difficult process as in all my life.

F. Sümeyra MERCAN

BALANCING MIXED MODEL ASSEMBLY LINES IN AN AUTOMOTIVE SUPPLIER BY RECONFIGURING LAYOUT

ABSTRACT

In globalizing world, enterprises must have the power to respond rapidly changing and rising customer demands. In order to have this power, enterprises generally produce specific characteristics or high volumes of standard products. In high-volume production of standard products, assembly lines are generally used.

Lines which in materials are processed by being transferred automatically or with help of labor are called assembly lines. The number of problems arises that ruin the ideal of the proper production in assembly lines. Assembly line balancing studies have been developed to overcome these problems. When the work required for assembly operations, time taken by these works and precedence relations between them are given; creating cycle time and station number to minimize idle time on the line and assignment of works to these stations orderly are called assembly line balancing.

Common purpose of hundreds of studies in Industrial Engineering and Operations Research literature is the effort of the creation of methods for elimination of line imbalance problems.

In this thesis, primarily (firstly), general information presented related to assembly lines and assembly line balancing problems, and then contributing to the private sector is aimed by running the application in an automotive company. In the last part of this thesis the balancing was compared before and after situation.

Keywords: assembly lines, line balancing, mixed-model assembly line, parallel assembly line.

BİR OTOMOTİV TEDARİKÇİSİNDEKİ KARIŞIK MODELLİ MONTAJ HATLARININ YERLEŞİMİN YENİDEN DÜZENLENMESİ İLE DENGELENMESİ

ÖZ

Globalleşen dünyada işletmeler, büyük bir hızla değişen ve gelişen müşteri isteklerine cevap verebilme gücüne sahip olmalıdır. Bu güce sahip olabilmek için işletmeler genelde, spesifik özellikte ürünler ve ya yüksek hacimlerde standart ürünler üretmektedirler. Yüksek hacimde standart ürünlerin üretilmesinde, genellikle montaj hatları kullanılmaktadır.

Malzemelerin işgücü yardımıyla ya da otomatik olarak transfer edilerek işlendikleri hatlara montaj hatları denir. Montaj hatlarında üretimin düzgün bir şekilde yapılabilmesi idealini bozan bir takım problemler ortaya çıkmaktadır. Bu problemlerin giderilmesi için montaj hattı dengeleme çalışmaları geliştirilmiştir. Montaj işlemlerinin yapılabilmesi için gerekli işler, bu işlerin aldıkları süreler, aralarındaki öncelik ilişkileri verildiğinde, hattaki boş zaman miktarını minimize edecek şekilde çevrim süresi, istasyon sayısı ve işlerin bu istasyonlara sıralı şekilde atanması çalışmalarına montaj hattı dengeleme denir.

Montaj hattı dengeleme ile ilgili Endüstri Mühendisliği ve Yöneylem Araştırması literatüründe karşılaşılan yüzlerce araştırmanın ortak amacı, hat dengesizliği problemlerinin ortadan kaldırılmasına yönelik metotların oluşturulma çabasıdır.

Bu tez çalışmasında öncelikle montaj hatları ve montaj hattı dengeleme problemleri ile ilgili genel bilgiler, sonrasında ise bir otomotiv firmasında uygulama çalışması yapılarak özel sektöre katkı sağlanması amaçlanmıştır. Tezin son bölümünde dengeleme öncesindeki durum ve sonrasındaki durum karşılaştırılmıştır.

Anahtar Sözcükler: montaj hatları, hat dengeleme, karışık modelli montaj hattı, paralel montaj hattı.

CONTENTS

	Page
M.SC THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGMENTS	iii
ABSTRACT.....	iv
ÖZ	v
CHAPTER ONE - INTRODUCTION.....	1
CHAPTER TWO - ASSEMBLY LINE BALANCING.....	4
2.1 General Information	4
2.2 Basic Terms of Assembly Line Balancing.....	5
2.3 Classification and Description of Assembly Line Balancing Problem	7
2.3.1 Number of Products	8
2.3.1.1 Single-Model Lines	8
2.3.1.2 Mixed-Model Lines.....	8
2.3.1.3 Multi-Model Lines	8
2.3.2 Paced and Unpaced Lines	9
2.3.2.1 Paced Lines	9
2.3.2.2 Unpaced Lines.....	9
2.3.3 Variability of Operation Times.....	9
2.3.3.1 Deterministic Operation Times	10
2.3.3.2 Stochastic Operation Times	10
2.3.3.3 Dynamic Operation Times	10
2.3.4 Assignment Restrictions	11
2.3.4.1 Task Related Constraint	11
2.3.4.2 Station Related Constraint.....	11
2.3.4.3 Position Related Constraint.....	11
2.3.4.4 Operator Related Constraint.....	12
2.3.5 Layout of the Production System.....	12

2.3.5.1 Serial Lines.....	12
2.3.5.2 U-Shaped Lines	13
2.3.5.3 Parallel Lines.....	14
2.3.5.4 Two-sided Lines.....	14
2.3.5.5 Feeder Lines and Supplementary Units	15
2.3.6 Type of Station.....	15
2.3.6.1 Degree of Automation and Flexibility	15
2.3.6.2 Closed Station – Open Stations.....	16
2.3.6.3 Inspection Station.....	16
2.3.7 Launching Discipline.....	16
2.3.8 Equipment Selection and Processing Alternatives	16
2.3.9 Volume of Production.....	17
2.3.10 Objectives	17
2.4 Mixed-Model Assembly Line Balancing	18
2.4.1 Constraint of Mixed-Model Assembly Line Balancing Problem	20
2.4.1.1 Basic Constraints.....	20
2.4.1.2 Sub-Constraints.....	20
2.4.2 Advantages and Disadvantages of Mixed-Model Assembly Line Balancing Problem	21
2.4.3 Literature Review of Mixed-Model Assembly Line Balancing Problem..	22
CHAPTER THREE - THE METHODOLOGY FOR SOLVING MMALBP ...	37
3.1 Basic Information about the Mixed-Model Line Balancing Problem.....	37
3.2 The Problem Statement	38
3.3 The Methods Used to Solving the Problem.....	40
3.3.1 The Algorithm Used to Solving the Problem	40
3.3.2 Ranked Positional Weighted Method	49
CHAPTER FOUR - APPLICATION	52
4.1 Factory Information.....	52
4.1.1 General Information about the Main Section	53
4.1.2 Introduction of the Front Seat Production Line.....	55

4.2 Description of Problem	58
4.2.1 Data Collection	60
4.2.1.1 Computing of the Planning Period	60
4.2.1.2 Information about Model	60
4.2.1.3 The Amount of Product-based Customer Demand	62
4.2.1.4 Definition of the Tasks in Assembly	62
4.2.1.5 Time Study about Processing Times	67
4.2.2 Information about the Current Line	67
4.2.2.1 The Joint Precedence Diagram for Current Line	68
4.2.2.2 Computing Cycle Time of Current Line.	70
4.2.2.3 Assignment Tasks to Workstations and Bottleneck Stations	72
4.2.3 Information about the Proposed Line	74
4.2.3.1 Separating into 2 Parallel Lines	75
4.2.3.2 The Joint Precedence Diagrams for Each Parallel Line	77
4.2.3.3 Computing Cycle Time of Parallel Lines.....	80
4.2.3.4 Initial Solution with Positional Weight Method	81
4.2.3.5 Improving Initial Solution with the Study Method	85
4.2.4 Simulation in ARENA and Comparison Results	86
4.2.4.1 Simulation of Current Model with ARENA	86
4.2.4.2 Simulation of Proposed Model with ARENA.....	94
4.2.4.3 Comparison of the Current and Proposed Situation.....	97
CHAPTER FIVE - CONCLUSION	99
REFERENCES.....	101
APPENDICES A	107
APPENDICES B	129

CHAPTER ONE

INTRODUCTION

In today's world, enterprises should prioritize some main purposes in order to be successful. These main purposes are; raising the level of productivity and efficiency, increasing capacity, improving quality, reducing costs, providing customer requests and satisfaction, using labor, machine and equipment effectively and providing ergonomic work environment.

In continuous production systems, the situations where the production is carried out in units and there is mass demand; the most reasonable way of meeting the demand with high production rate is configuration of assembly lines (Anonym, 2011).

Combining components of a system in a specific order and sequence is called assembly. Assembly starts with parts that completely independent of each other and ends with combination of these parts to form a whole system (Keskindürk & Küçük, 2006).

The system, which requires that stations, that are formed by transferring with the advantage of hardware or labor force through material flow line and unifying operations on component with taking into consideration of constraints such as cycle time and primary relations between them, is arranged in a line, is named as "assembly line". For the efficiency of assembly line, just because of distribution of work to stations in such a way that leaving little time or no time in assembly line to each assembler during production period, i.e. under the existing constraints, a very high number of processes and production rate, assembly line balancing is; smallest sum of the processing time differences between work stations (Kahraman & İspir, 2004).

One of the main purposes in assembly line balancing is distribution equal amount of work load to each work station. During the assignment of these works to stations,

various constraints are discussed. Main constraints are; cycle time constraint and precedence constraint of work items. And side constraints are; stable equipment constraint, station load constraint, work-elements to be grouped together at one station and works to be separated from each other. These constraints make the complicate of a complete line balancing and prevent the provision of work load balance. There will be more workload on some stations than the others; reductions in efficiency and emergence of some losses are inevitable. But the purpose is, to find line balancing solutions that will minimize the loss.

Many manufacturers are switching their production lines from single product or batch production to mixed-model production, often as a consequence of implementing just-in-time (JIT) principles into their operations. In mixed-model production, different products or models are produced on the same line with the models interspersed throughout a production sequence. This helps manufacturers provide their customers with a variety of products in a timely and cost effective manner (Sparling & Miltenburg, 1998).

Enterprises trying to keep up with today's conditions; as well as adopting a mixed-model production, aimed to minimize idle times in lines and elimination bottleneck. This research deals with mixed-model assembly line balancing and parallel assembly line balancing problems.

Considering the contents of the study, in the second chapter called assembly line balancing, general information that tells what the assembly line balancing is, basic concepts used in assembly line balancing, classification and identification of assembly line balancing problems are investigated. In addition to this, a detailed study related to mixed-model assembly line balancing problems is included in chapter 2. At the end of the second chapter, general literature search is done related to assembly line balancing.

In the third chapter, problem description is done. Basic information about the problem application, purposes and solution methods are detailed. The algorithm detail which is used in application is given.

The fourth chapter of the study is the application part. The context of the application, the presentation of the enterprise where the application will be performed at, practice area and detailed descriptions about study methodology are explained with support of visuals. Planning and data collection phase is described in detail. A model is developed related to the current situation. The new simulation model is improved after line balancing and compared with existing model. Finally, the comparison results are evaluated.

CHAPTER TWO

ASSEMBLY LINE BALANCING

2.1 General Information

The process in which the parts of a product are combined in accordance with a predetermined sequence is named as assembly. The production line in which assembly operations that carried out on sequential stations is called assembly line. Competition motive between enterprises has created a need to ensure mass production and flexibility in product range. Also assembly lines reached various shapes and capacities by growing and developing over time in order to meet demands. Due to this development in assembly lines, assembly line balancing problem has occurred. In cases, when all work required for the realization of the assembly, time taken by these works and priority relations between works are known; assignment of any performance scale of works (e.g. idle time, cycle time, number of stations) to sequential workstations in order to optimize, is called assembly line balancing problem.

According to Erel, Sabuncuoğlu & Aksu (2001), assembly line balancing can be defined as; “Line balancing is the process of allocating a set of tasks (the smallest indivisible portions of the assembly operation) to an ordered sequence of stations in such a way that some performance measures (e.g. cycle time, number of stations) are optimized subject to the precedence relations among the tasks.”.

In order to get effective results in assembly line balancing problems, there are various purposes. These are (Erkut & Baskak, 1997);

- ❖ Ensuring a regular material flow,
- ❖ Using manpower and machine capacity at the highest level,
- ❖ Completing operations as soon as possible,
- ❖ Minimizing the number of workstations on the assembly line,
- ❖ Minimizing idle times,

- ❖ Distributing idle times properly between workstations,
- ❖ Minimizing production costs,
- ❖ Minimizing line cycle time.

But the implementation of above-mentioned objectives all at once is impossible. In this direction, our purpose in the line balancing is balancing the line considering constraints and criteria above in the most appropriate way (Tanyaş & Baskak, 2003).

The idea of line balancing was first introduced by Bryton (1954) in his graduate thesis. The first published scientific study belonged to Salveson (1955). For more than 45 years, many studies have been made on this subject. During this period various new balancing problem concepts such as U-type, two-sided, parallel, flexible assembly line, etc., and solution algorithms for those problems have been produced. The common thing for all these problems is using both the operator and the machine in the most efficient way, at the same time providing flexibility in production (Ağpak & Gökçen, 2004).

2.2 Basic Terms of Assembly Line Balancing

An Operation (Task) is the smallest part split logically of the all work content that carried out during the production process of the finished product (Çakır, 2006).

Station is the space used by workers where the defined work is completed by using such tools on the assembly line. For an assembly line; there are constraints such as the smallest station number is one and the biggest station number determined during the station number balancing operation should not be exceeded (Çakır, 2006).

Cycle Time can be defined as; the longest period of a product at a station on the assembly line or the necessary period of time for a worker at a workstation in order to complete the work to be done. The total time period of work items assigned to a station, cannot exceed the cycle time (Çakır, 2006).

Processing (Task) Time is the required time for the realization of the smallest part split logically of the all work content that carried out during the production process of the finished product.

Idle Time is a positive difference between the cycle time and the task time. The sum of idle times for all stations of the line is called balance delay time (Çakır, 2006).

Makespan is the time required for the assembly of a product to be produced on the assembly line or sum of standard durations of all work items at all workstations of the product.

Precedence Constraint is predetermining in what order operations will be made. This partial ordering of tasks can be illustrated by means of a *precedence diagram* (See Figure 2.1) which contains nodes for all operations and arcs (i,j) if an operation i must precede an operation j. The combined precedence diagram of products more than one kind that will be used on mixed-model assembly lines is called *joint precedence diagram* (See Figure 2.2).

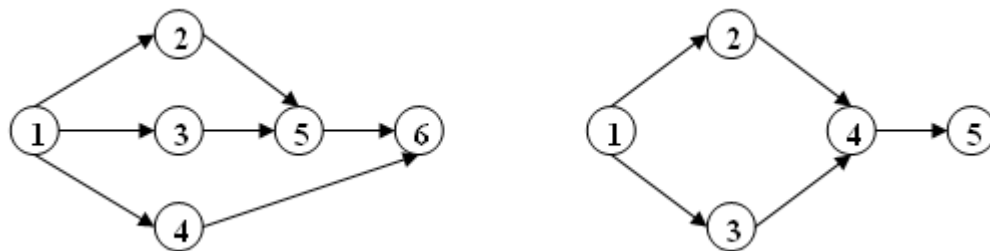


Figure 2.1 First one is the precedence diagram for A product and second one is the precedence diagram for B product.

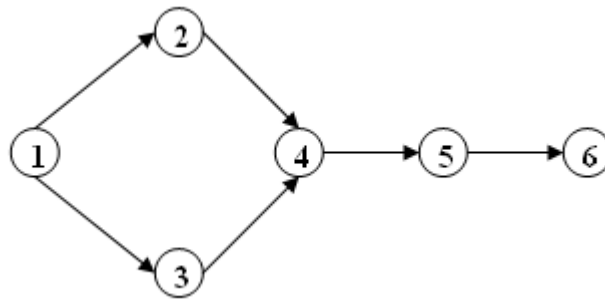


Figure 2.2 Joint precedence diagram for A and B product.

2.3 Classification and Description of Assembly Line Balancing Problem

Assembly line production systems are present in different industrial environments and are utilized to manufacture a large variety of products. Especially, they are used to produce consumer goods such as cars, engines, domestic appliances, television sets, computers and other electrical appliances. These products are rather different, and it is necessary to implement different production systems (Scholl, 1999). Figure 2.3 illustrate the main characteristics of assembly line balancing problems relationships.

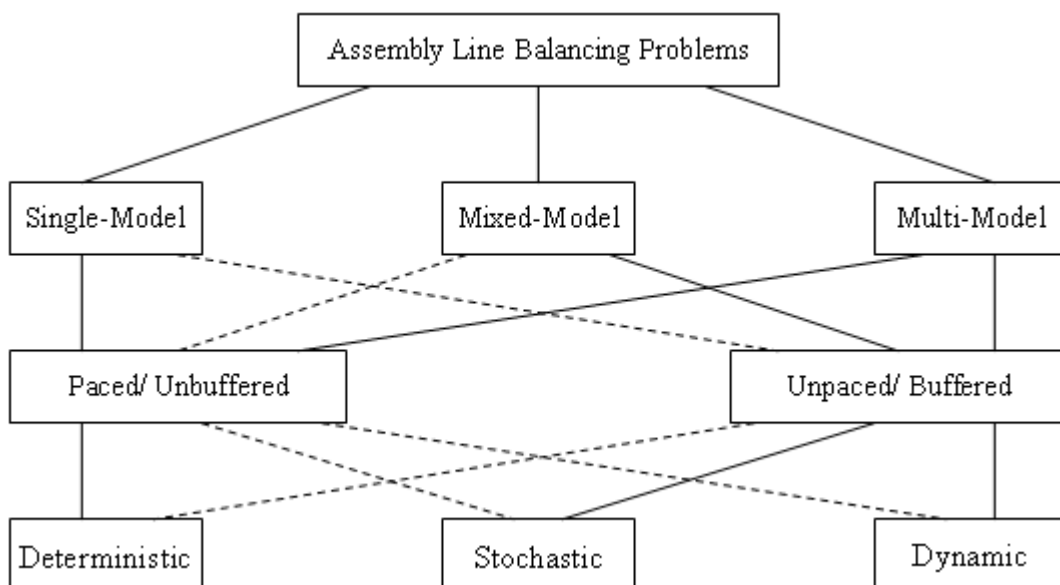


Figure 2.3 Classification of assembly line balancing problems.

2.3.1 Number of Products

One of the assembly line balancing classification criteria's is number of different products which can be produced on the same line.

2.3.1.1 Single-Model Lines

One homogeneous product is continuously manufactured in large quantities. According to Merengo, Nava & Pozzetti (1999), single model lines are “suitable for large-scale production, since they ensure quite low production costs.” (s. 2836). No operation changes are being made at any stations on this kind of lines and all stations repeat the same work. Thus, does not change in workloads of stations.

2.3.1.2 Mixed-Model Lines

It is the line system that provides sequential production by mixing more than one product on the same line. Product ranges produces on the same line are quiet similar to the main product. According to Merengo, Nava & Pozzetti (1999), “it is possible to produce very small batches (even one – unit batches.)” (s. 2836). Also when there is model change on the line, set-up is carried out quite fast and cheap. For example, if option differences of main product are produced sequentially mixed on the same line according to customer demand, this belongs to mixed-model assembly lines class.

2.3.1.3 Multi-Model Lines

Similar products with differences in production processes are produced on these lines. Due to differences in production processes, because of situations like operation processing times, ergonomic need of work space and so on, products are produced in batches. Even a lengthy set-up study is needed during product change. These changes cause an increase in costs and a decrease in productivity.

2.3.2 *Paced and Unpaced Lines*

Due to product range produced on assembly lines, material handling equipments which products are moved by also show a change.

2.3.2.1 Paced Lines

Systems, that provide the realization of operations in continuous flow by linking material handling equipment and stations, are called paced lines. Operators move within the boundaries of workstation in order to make the process on the working flowing line, and when the work is finished operators return to the starting point of the station. According to line regulation, line allows the operation to be done by stopping at stations during the processing time and when the time is over, it moves to the next station.

2.3.2.2 Unpaced Lines

In unpaced lines, the stations are decoupled by buffer stocks which hold workpieces when the succeeding station is still working on previous items. Since buffer capacities are normally restricted, a station may be blocked when the subsequent buffer is full. Then the considered station is idle until the succeeding station requires an item stored in the buffer. Another inefficiency, called starvation, occurs when the input buffer of a station is empty after terminating the current job. Then the station is idle until a workpiece enters the buffer. Starvation may be caused by a lower production rate or a break-down of the preceding station. Note that a starvation cannot be avoided in paced systems whenever the total work content is not equally distributed among the stations (Scholl, 1999).

2.3.3 *Variability of Operation Times*

Depending on structure of tasks and operators, even some changes can be observed in operation times. While it is known that the variability of operation times

in simple tasks is less, operation times in complex structured works show a change due to employee's physical structure, psychology and social environmental factors.

2.3.3.1 Deterministic Operation Times

When there are very small changes in process times, most of the assembly lines' operation time is accepted as constant deterministic. Highly automated production systems reduce changes in process times to minimum. However, in order to prevent changes in process times on assembly lines that the manpower is used, skilled workers with high capacity and highly motivated are needed to be run.

2.3.3.2 Stochastic Operation Times

Changes in process times on assembly lines that the manpower is used, cause an emergence of stochastic operation times.

Excessive operation times result in inefficiencies such as starving of succeeding stations or blocking of preceding ones in unpaced systems. In case of paced assembly lines, workpieces cannot be processed completely whenever the station time exceeds the cycle time. Some of the possible consequences are that the complete line must stop until the work is completed, that additional (utility) workers have to be employed, or that in complete units have to be reworked at additional off-line stations. The same problems arise when defective parts are produced (Scholl, 1999).

2.3.3.3 Dynamic Operation Times

The operation time in newly formed assembly lines or required by an operator during learning a new task process is much more than the required time for employee's learning period as a result of its ability to learn the job and successful results at the end of the process of getting used to running in production process.

2.3.4 Assignment Restrictions

While the assembly line balancing is being applied, during the assignment of works to stations, there can be many constraints, notably the priority constraint.

2.3.4.1 Task Related Constraint

In some situations, two tasks are desired at or must be assigned to the same situation or section of the line. This can be modeled by maximum distance. In some cases, tasks are incompatible, i.e., they must not be performed at the same station or segments of the line. This can be expressed by Minimum distance between the tasks.

These restriction are frequently called zoning constraints.

2.3.4.2 Station Related Constraint

If special machines or tools which are needed to execute a certain task are only available in one or a few stations and cannot be moved without causing prohibitive costs, the task has to be assigned there. A similar restriction occurs if material needed for some task is only available in a particular section of the line. This may be due to limited space for material stocks.

2.3.4.3 Position Related Constraint

Especially in the case of large and heavy workpieces, tasks may need certain position of the workpieces. Since it maybe neither possible nor economical to turn the workpieces too often, tasks which need the same position have to be grouped together in a segment of the line. Workpieces which cannot be turned into another position at all are called fixed items, while removable items may be turned or removed from the conveyor.

2.3.4.4 Operator Related Constraint

Depending on their complexity, tasks require different levels of skill. The requisite qualification of a worker is determined by the most complex operation assigned to the respective station. Therefore, it may be necessary to concentrate complicated tasks in a few stations (Scholl, 1999).

2.3.5 Layout of the Production System

Lay-out of production systems in flow lines are partially determined by the material flow. In addition, some changes can be made in the system in order to use the line more efficiently.

2.3.5.1 Serial Lines

A traditional line organizes stations and the tasks that comprise them sequentially along a straight line (Ajenblit & Wainwright, 1998).

Due to reasons such as being simple and systematic, placement is easy, conveyor system provides the applicability, cost reduction, and it does not contain transition difficulties that may occur in the angular lines; straight lines are preferred in placement of lines. A serial assembly line is illustrated in Figure 2.4.

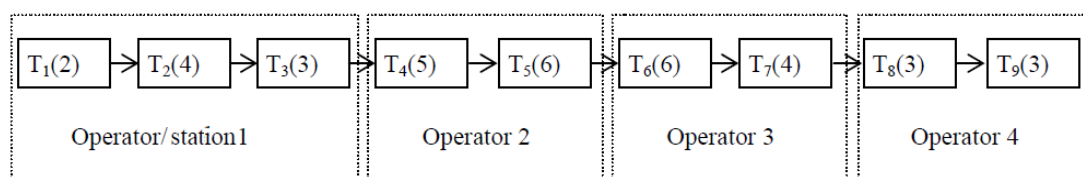


Figure 2.4 Serial lines.

2.3.5.2 U-Shaped Lines

In a U-shaped line, tasks are arranged around a U shape line and are organized into stations that can cross from one side of the line to the other. The assignment of the tasks to the stations on a U-line exploits the geometry of the line to keep the return and crossover distances as small as possible (Baykasoğlu & Dereli, 2009).

The number of stations needed for a U-shaped line layout is never more than the number of stations needed for the traditional straight line (Ajenblit & Wainwright, 1998). A U-shaped assembly line is illustrated in Figure 2.5.

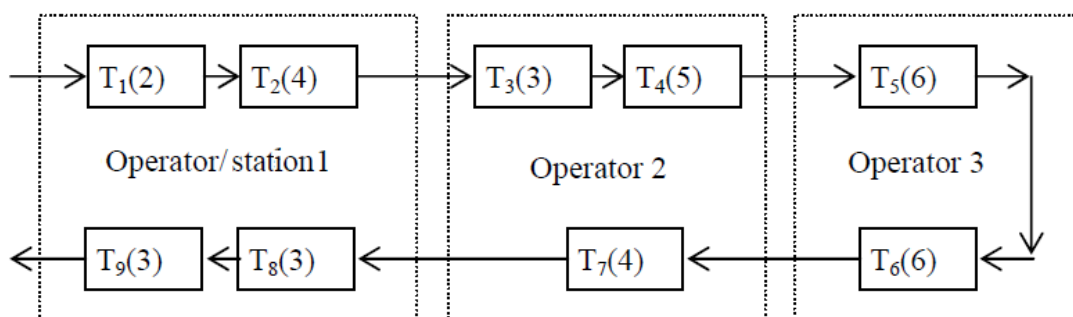


Figure 2.5 U-shaped lines.

The most important advantage of the U-Shaped line placement is providing flexibility in number of employees in order to adapt to optional and capacity changes in customer demands.

There are also many reasons for the current popularity of U-lines as an alternative to traditional batch production in shops with functional lay-outs. These include lower inventories, simpler material handling, easier production planning and control, opportunities for teamwork and problem solving, better control of quality, and so on (Miltenburg & Wijngaard, 1994).

2.3.5.3 Parallel Lines

In modern production environment, number of developing and flexible enterprises is rapidly increasing and these enterprises adopt JIT technique. Therefore, many traditional structures are unable to meet customer demands. The system in which more than one parallel and similar lines meeting customer demands oriented work synchronized, is called parallel lines.

In practically, most production systems consist of one or more assembly lines. There are two cases in producing products on one or more assembly lines. In the first case, the demand is high enough and a single line is insufficient to meet it and a second line is needed to be formed. In other words, the same products are produced on multiple identical lines. In the second one, if each demand is large enough to form a line, similar products more than one are produced on separate assembly lines (Gökçen & Ağpak, 2004). A parallel assembly line is illustrated in Figure 2.6.

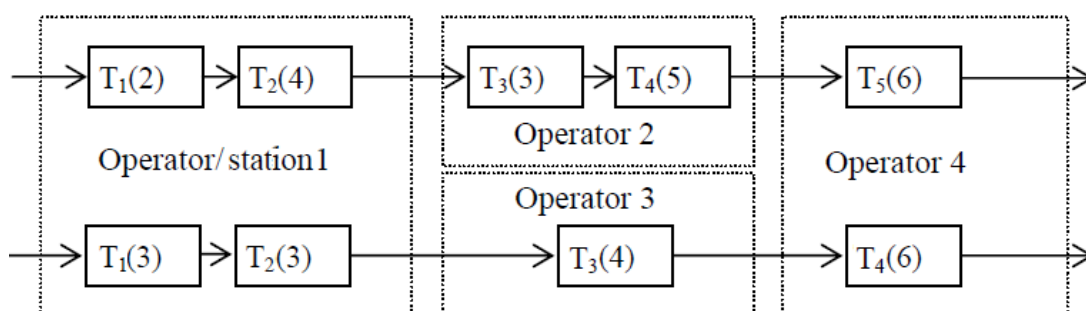


Figure 2.6 Parallel lines.

2.3.5.4 Two-sided Lines

Two-sided assembly lines are typically found in assembling large-sized high-volume products, such as buses and trucks. In a two-sided assembly line, both left and right sides of the line are used and different assembly tasks are carried out on the same product in parallel at both sides (Wu, Jin, Bao, & Hu, 2007). A two-sided assembly line is illustrated in Figure 2.7.

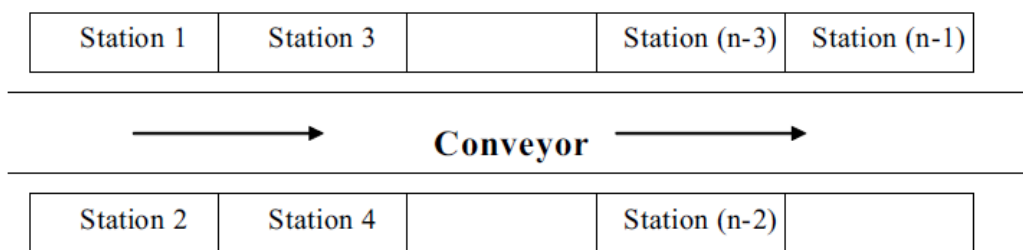


Figure 2.7 Two-sided lines.

The consideration of the preferred operation directions is important since it can greatly influence the productivity of the line, in particular when assigning tasks, laying out facilities, and placing tools and fixtures in a two-sided assembly line (Lee, Kim & Kim, 2001).

2.3.5.5 Feeder Lines and Supplementary Units

In flexible manufacturing systems, assembly lines are often used for the final assembly of products. Components (subassemblies) are produced by different supplementary units which may be organized as job shops, flexible manufacturing cells, or feeder lines. Then the balancing problem is connected with the problem of synchronizing the different production processes, i.e., the production rates of the supplementary units have to be determined (Scholl, 1999).

2.3.6 Type of Station

The stations may have different structures depending on the production line layout, style of production control and mechanical structure.

2.3.6.1 Degree of Automation and Flexibility

Stations on assembly lines are divided into 3 groups depending on their mechanical structures. Stations at which worker or workers make operation with simple tools are called manual stations. Stations where material procurement and

operations are performed by the employee and the line system is automation; are called semi-automated stations. And at automated stations, all operations are carried out automatically.

2.3.6.2 Closed Station – Open Stations

These stations determine the employee's working limits at the station. At closed stations, workers cannot switch station limits. U type stations are used when required by process. At open stations, employees can work out of the station until a certain distance.

2.3.6.3 Inspection Station

Inspection stations are created for quality control. These stations should be added to production processes.

2.3.7 *Launching Discipline*

It states the frequency of loading workpieces to production line. This frequency is the factor of running in production system, formation of bottleneck and starvation situation. This rule is divided into 2 as constant loading rate and variable loading rate. Constant loading rate are usually applied in paced systems. It provides a loading of workpieces to production with constant intervals. Variable loading rate is applied in more flexible work systems. Since the loading is accepted at the first station, what is going to be discussed is; starvation situation if the processing time of the first station is long and bottleneck situation if the processing time is short.

2.3.8 *Equipment Selection and Processing Alternatives*

Considering the flexibility of the chosen equipment, the tasks have to be assigned to the stations. While *general-purpose machines* are able to perform many different tasks, *special-purpose machines* are restricted to a small number of operations. The

latter ones lead to station related assignment restrictions. Because the selected equipment influences the task assignment, and vice versa, the balancing problem and the equipment selection problem have to be considered simultaneously. The combined problem is called an assembly system design problem (Scholl, 1999).

2.3.9 Volume of Production

It can be defined as “small-lot-assembly” due to low production numbers and production of large and expensive products, “mass production” due to high number of production and production of standard and cheap products.

On small-lot-assembly lines that cycle time is long and production rate is low, variable processing time is too much. Therefore, learning is slowly progressing and consequently the balance is becoming hard. In addition; just because parts in production of these products are expensive, supply and loading time must be planned carefully.

Production of standard and cheap products in serial production provides a detailed workload distribution and close repetition frequency of operations. This also speeds employee’s learning process up.

2.3.10 Objectives

Newly formed systems should be balanced in the design phase, and already installed systems in particular periods or including changes in production processes. Purpose of the enterprise in balancing should be made considering investment targets. Relevant purposes can be collected in main groups as; purposes related to capacity, purposes related to time, cost-related purposes, social and organizational purposes.

2.4 Mixed-Model Assembly Line Balancing

The current market is intensively competitive and consumer-centric. For example, in the automobile industry, most of the models have a number of features, and the customer can choose a model based on their desires and financial capability. Different features mean that different, additional parts must be added on the basic model. Due to high cost to build and maintain an assembly line, the manufacturers produce one model with different features or several models on a single assembly line. Under these circumstances, the mixed model assembly line balancing problem arises to smooth the production and decrease the cost (Xu & Xiao, 2008).

Productions of products in which customer demands are high, using single-model lines are not appropriate. In addition; since mixed-model lines are able to provide product range, the market can meet the demand and therefore it is widely used in the market.

Many manufacturers are switching their production lines from single product or batch production to mixed-model production, often as a consequence of implementing just-in-time (JIT) principles into their operations. In mixed-model production, different products or models are produced on the same line with the models interspersed throughout a production sequence. This helps manufacturers provide their customers with a variety of products in a timely and cost-effective manner (Sparling & Miltenburg, 1998).

Line balancing purposes for mixed-model assembly line are as follows (Scoll, 1999);

MALBP-F (Feasibility of mixed-model assembly line balancing problem), is the form of single-model assembly line balancing problem based average model suitability converted into mixed-model problem. If it is considered that each model has its own process priority diagram, these diagrams can be grouped in a single joint precedence diagram. Therefore, mixed-model line balancing problems that have

more than one model can be formulated as single model line balancing problems. Under these conditions, the most efficient assignment is provided with given cycle time and number of stations.

MALBP-1, is ensuring minimization of the station's number when the cycle time is given.

MALBP-2, is ensuring minimization of the cycle time when the number of stations is given.

MALBP-E, is ensuring maximization of the line efficiency due to number of stations and cycle time.

There are some assumptions and line features determined for mixed-model assembly line balancing problems. Assumptions used and line features can be defined as follows;

- ⊗ It is assumed that customer demand is known in advance.
- ⊗ Operation times of all operations are considered as deterministic, unless otherwise stated.
- ⊗ Precedence relations of operations belong to each model is definite and consistent on the basis of the model, in other words precedence of operations do not replace according to the model.
- ⊗ Cycle time is determined in advance.
- ⊗ Operation times cannot be more than cycle time.
- ⊗ Each operation must be assigned to any station and each operation must be assigned to one station.
- ⊗ Since some operations may be required to be assigned to the same station, also may be required to be assigned to a different station.

In accordance with above-itemized assumptions, problem definition, content and solution are formed step by step.

2.4.1 Constraint of Mixed-Model Assembly Line Balancing Problem

2.4.1.1 Basic Constraints

- ❖ *Cycle Time Restriction:* The total duration of operations assigned to a station (i.e. task times, sum of lost times due to uncontrollable periods and pre-designed downtimes), cannot exceed the cycle time. When the sum of task durations in a work centre exceeds the specified cycle time, either one or more tasks must be removed from the work centre, or else duplicate workstations (and workers) can be included in the work centre (Yılmaz & Erol, 2005).
- ❖ *Priority Relations:* It is the constraint which sets the condition such as; other priority processing has to be done while any of assembly operations are being made. In other words, the priority processing must be completed before starting a process (Yılmaz & Erol, 2005).

2.4.1.2 Sub-Constraints

- ❖ *Constant Equipment Constraint:* It is the constraint that provides assignment of operations to stations without changing station locations on assembly lines which have constant equipments such as machines and testing tools. Constant equipment constraint reduces the modifiability of work items (Özkıran & Düşünür, 2011).
- ❖ *Station Load:* This constraint provides the situation in which operation times of some stations on assembly line is quite lower than the cycle time. Especially at first station or stations, it can be applied in order to reduce the effect of probable disruptions in the beginning of the line to entire line. This constraint is not mandatory (Özkıran & Düşünür, 2011).

- ❖ *Works Desired To Be Assigned To the Same Station:* This constraint allows assignment of works to the consecutively or same station which require the same equipment to be used or the same operator to run due to works require special qualifications (Özkıran & Düşünür, 2011).
- ❖ *Works Not Desired To Be Assigned To The Same Activities:* This constraint prevents assignment of works to the same station which requires advanced or extreme physical force or using big equipments that cannot be placed to the same station (Özkıran & Düşünür, 2011).

2.4.2 Advantages and Disadvantages of Mixed-Model Assembly Line Balancing Problem

Mixed-model production systems are mainly used due to the following advantages (Cao & Ma, 2008);

- They provide a continuous flow of materials,
- They reduce the inventory levels of final items,
- They are very flexible with respect to model changes.
- They keep up with customer demands.

Mixed-sided assembly lines in practice can provide disadvantages over a single-model assembly line.

- One of the most important disadvantages is, it has more constraints than single model assembly line balancing problems due to much product range.
- The flexibility of the mixed model line requires expensive equipment which reduces or even eliminates delays due to set-up activities.

2.4.3 Literature Review of Mixed-Model Assembly Line Balancing Problem

Many studies on the mixed-model assembly line balancing problems including exact solution methods, heuristics and meta-heuristic approaches have been reported in the literature. Summaries of some of them are presented in this section.

Aşkin and Zhou (1997) developed heuristic that assign tasks with creating parallel stations for balancing in mixed-model production lines.

This study deals with the assignment of tasks to workstation in serial production systems and designed a model accordingly.

According to this study, objective function is; balancing the idle time of station with increase of task-dependent equipment.

In this study, integer and dynamic programming are used to solve the problem. In order to increase the use of workstation or optimize a task that has a greater time than cycle time, some methods of task-parallelizing or station-parallelizing are considered.

A mathematical programming is used to allow parallel stations. A heuristic is developed for assignment processes. In heuristic's content, firstly task sequence is determined. Then, weighted average task set is defined. This weighted average task time is used when the appropriate time for workstation is determined. The necessary lower bound should be determined for the number of stations. Detailed information is given about the formation of parallel stations. There are two separate situations for the formation of more than one workstation. One of them is the product operation time is more than the cycle time. In this case, the task cannot be completed and therefore a station in parallel may be needed. Another one, if there is no appropriate task for the current workstation, this station can be closed and continued assignment to a new station. Or, cost of equipment increase is considered by opening a parallel workstation. Because of this case, the utilization factor strategy belong to the current

station is applied. At the final stage, the heuristic task assignment is carried out. An algorithm is created for this and given in the content of the study.

Station utilization is considered by using a threshold variable. Computational experiments are also provided to analyze the performance of the heuristic. The total cost calculation has been realized with the heuristic solution.

McMullen and Frazier (1997) have presented an approach to solve mixed assembly line balancing problems with stochastic task times of tasks becoming parallel at work centers. In this approach, task selection rule is established for tasks to be assigned to work centers.

Companies producing high volume and mixed model products can duplicate some of their equipment in order to increase production flexibility and the amount of output. The reason is the longest task time exceeds the cycle time. And in this approach, equipment duplication is implemented by creating parallel workstations for tasks exceeding the cycle time. The Just in Time production system is applied within the study and it is purposed to minimize the number of employees without changing the cycle time (type 1). In addition, task times are set to be stochastic, considering that the performance of employees can be affected by environmental effects.

While performances of different task selection rules are evaluated, the average stock level in the process, the average flow time, the unit number in the system in the given period, the average unit labor cost, the average system utilization and the percentage of completed tasks at each work center are considered. Here, the numbers of worker to keep the performance criteria high, and the amount of required equipment for the assembly line to run, are important.

The methodology developed in this study, is the modification for mixed model production systems with stochastic task time of the heuristic developed by Gaither (1996) for single model production systems with deterministic task time. In this heuristic, when the cell usage increases, new work centers are created by closing the

related cell. Thus, tasks are paralleling with workers who can perform these tasks. Due to paralleling, task times exceeding the cycle time are eliminated.

The algorithm providing the distribution of tasks is created and described with its steps within the study. Seven task selection rules are created in the algorithm and results are compared by simulating in a computer program. In the first rule, task with the highest cell utilization according to the expected task time and in the second rule, the list of tasks assigned to the current cell are determined. The third rule is, selecting the task that has the longest task time from the list of tasks to be assigned to the current cell. The fourth rule is, selecting the task that has the shortest task time from the list of tasks to be assigned to the current cell. The fifth rule is, selecting the task with highest cell usage from the list of tasks providing the least cell usage. The sixth rule is, selecting task from all tasks in the cell from the cell with the highest possibility to be completed in time, in order to enter the current cell. The seventh rule provides selecting the product with highest cell utilization.

Performance measurements are used in order to compare the data obtained as a result of rules. These performance measurements are also obtained by computer simulation.

Sparling and Miltenburg (1998) did a study for balancing mixed-model U-line running with the JIT (just-in-time) concept. They aimed the assignment of tasks required for the production of all models, to minimum number of stations.

In mixed model lines, different products or models are produced along the same line. Due to the adaptation of the line to product range, provides meeting the customer demand by minimizing time and cost. On the other hand, balancing mixed-model assembly lines are more difficult than balancing single-model assembly lines. Because, completion times of tasks and priority relations vary from model to model. Therefore, mixed-model assembly line balancing for straight lines is examined first. Thomopolous (1967-1970)'s 4-step heuristic procedure is discussed for this. Model sequencing can be applied in the final balancing of 4.step to reduce impacts of imbalances.

All tasks are completed by an operator along the cycle of only a single model in a station on the straight line. In a station on U-line, for different units, tasks performed in the front side of the line and tasks performed in the back side of the line may differ from each other. According to this, a model is designed for mixed-model U line balancing problem and the solution algorithm of this model is established. Smoothing algorithm arranges the initial balance in order to reduce imbalance level of the model. This algorithm is given in the final balancing. After that, model sequencing is used to reduce the impact of rest unbalanced models. Model imbalance is measured by comparing targeted times and required times for stations.

With smoothing procedure, reducing imbalance measure value is provided by displacement of tasks between stations. Three observations are used for this. First of these, tasks with a high variability of processing time, create more imbalance than tasks with low variability. Second, task pairs that positively-correlated in processing times create more imbalance than task pairs that negatively-correlated in operation times. Third, a task that has positive correlation between its own processing time and total time of a station, create more imbalance than a task that has negative correlation between its own processing time and total time of a station. Smoothing algorithm is based on these observations.

Traditionally assembly line balancing problem is known as NP-hard problem. The U-line balancing problem in this study is generated by inspired the traditional assembly lines, so it is NP-hard problem. In NP-hard problems, approximate solution algorithms are required for the solution of realistic size samples.

Erel and Gökçen (1999) are studied the balancing problem with the shortest-route formulation by turning mixed-model assembly lines into single model assembly lines.

In order to meet the customer satisfaction, and also to get high volume and variety of products, mixed-model assembly lines are examined within the scope of even this study. Creation of task sets of each model, performance time measurement of tasks, considering precedence relations are quite difficult. It is assumed that each model has common tasks to avoid this situation in this study. In the other words, even if

performance times of tasks, belong to different models which are considered common is different each other, will be processed in the same station. In this study, sum of idle times for each model generates the performance criteria.

In the shortest path formulation, there are arcs representing possible assignments of tasks to stations. Every path from source to sink refers to the design of the line. And nodes occur in a similar way for each model. Thus, the tree composed of nodes and arcs grow at a rate depending on the size of each model and each task.

The combined precedence diagram is created by combining priority relations of models. Thus, the mixed-model line balancing study turns to a single model balancing study. The precedence matrix which is generated as upper-triangular shape is obtained from the priority relations diagram. Many assumptions and various constraints are adopted with the assumption that the required to assign tasks which are accepted common for different model to same station.

To achieve the aim of minimizing the total value of idle times in stations, first it is needed to minimize the number of stations.

Nodes in the model's structure show the set of tasks that have no tasks to be completed before it, in the precedence diagram. Here the path consists of nodes, means the sum of idle times of all stations for all models. The node generation process described in the study shows all possible assignments.

This study can set a framework for developing heuristic procedures in mixed-model assembly line solutions.

Merengo, Nava, and Pozzetti (1999) studied for sorting and balancing in manual mixed model assembly lines. They worked with three assembly line types in this study. First of these is 'moving line', composing transport system in which parts are carried along the line by being distributed smoothly. Second one is 'paced line' that provides the transportation with regular intervals. In these lines, parts stay in the station during the cycle time and are moved to the other station when the time is

completed. In these lines, the part stays in the station during the required time for completion of the work and takes its place in the buffer when the work is completed.

The objectives under this studies are minimizing the rate of incomplete jobs (in paced lines and in moving lines) or the probability of blocking/ starvation events (in unpaced lines), reducing WIP (work in progress). In addition, minimizing the number of stations is also aimed.

Firstly, balancing is discussed within the article. Balancing is, the distribution of basic assembly tasks to different work stations, under given constraints. In other words, it is determining the number of stations will be used and distribution of tasks to be assigned to each station. In the system discussed, an operator is appointed for distribution of all assembly tasks to given stations. Here, various constraints are considered. Situations in which cycle time is shorter than the processing time may arise. In this case, the operator works peak and the station is extended enough for the operator to complete his work. This kind of situation may be ignored if only there is an effective sequencing. Two balancing types as horizontal and vertical are given under the study. Even each balancing type is divided into two groups according to the constraint it includes.

The balancing algorithm given in the study includes 4 different balancing types and consists of 3 steps for each version. In the first step, it creates initial solution, in the second step, tries to reduce the number of stations without deterioration in horizontal balancing and by developing the initial solution, and in the third step, develops the vertical balancing by correcting the solution formed in the second step and without deterioration in horizontal balancing again. Later, these studies are extended to other line types.

Then, sequencing is discussed within the article. It is focused on the production system that had been focused in balancing. Here, incomplete parts possibility is minimized in order to get the best sequencing methodology. Minimum part set (MPS) principle has been mentioned in sequencing. This procedure cannot provide the best solution every time. In a situation facing a problem, all sets of units to be produced cause a very high computation time. Basic principles which will be valid

for the sequencing are determined and the sequencing algorithm is organized according to these principles. Then, studies obtained extend to even other line types.

Balancing and sequencing methodologies are tested. The test is discussed in 2 parts. In the first one, a comparison is done between four versions of the balancing algorithm. And in the second one, a comparison is done between sequencing algorithms described previously. FORTRAN programming language is used in order to test all algorithms.

Methodologies recommended here are designed specifically for the transport lines. But they can also be applied to paced and unpaced lines with extension.

Matanachai and Yano (2001) have balanced mixed-model assembly lines, in order to reduce workload of work stations. Therefore a heuristic solution procedure based on filtered beam search is developed. Their focus is an assigning task to stations so that workloads are reasonably well balanced and it is relatively easy to construct daily sequences of jobs that provide stable workloads (in a minute to minute sense) on the assembly line. Stability provides to contribute to the quality of the product by the fact that employees working without having to rush. For it, they focused on closed-station, paced lines with Fixed-Rate Launching (FRL) on structure of the line. Works on the line are transported with a constant-speed conveyor at equal intervals. If task times exceed conveyor cycle time, work overload occurs. In this case, either the operator will be worked quickly by pass over the quality, or uncompleted tasks will be completed in repair station, or line efficiency will be reduced by stopping the line for the operator to complete this task, or cost will be increased by adding operators who know the job well to complete tasks quickly. The quality of produced products, line efficiency and cost caused by the line are important for the line balancing. The objective of this article is developing a line that better meets daily model changes as well as developing a line that works with higher value than average performance. Therefore, balancing study is done considering the workload.

In the article, various terms are available in the objective function. The various terms are available in the objective function of the article. The first is that, to minimize sum of the absolute deviation of the actual usage rate of each station from

the average usage rate. Secondly, to minimize sum of the absolute deviation of the actual usage rate of each station created for each job type, from the average usage rate overall. And the last one is, to minimize sum of absolute deviation of the processing time of each station's job type for across all job types from the average processing time of this job type's each station. It is to minimize sum of the absolute deviation of the actual usage rate created from average usage rate overall.

Due to difficulties in reaching the optimal solution of the presented model, a new heuristic method is developed to reduce the number of decision variables. In this heuristic, the processing time of each task is changed by workload of that task.

The proposed heuristic procedure is similar to a filtered beam search. According to this approach, starting with the first station, one station is constituted at a time. It is tried to create various potential task assignments for each station. For a station, a subset is considered that consists of tasks which have even suitable priority relations. Tasks which have to be assigned to the next station of set are allowed to be assigned to that station. In the beginning, branches are created with possible subsets and the best objective value. The remains are stored for backtracking. After the solution is completely created for all stations, the improvement procedure including the transfer of tasks from station to station is applied.

The study is discussed in two parts. In the first part, small problems which are reached the optimal results including both purposes are solved. And results are compared with results of the proposed heuristic solution approach. Since the study is limited with small problems, the aim is to measure benefits of using a new objective function and losses of using heuristic. In the second part, solutions obtained by the proposed heuristic are compared with solutions obtained by adapting Rachamadugu and Talbot (1991)'s heuristic.

Line performance is not depending on only the line balance, but also depends on the sequencing and station length. As a result, in order to compare different line balancing approach performances both of the two decision variables must be controlled.

Jin and Wu (2002) tried to balance mixed-model assembly lines by taking advantage of goal chasing method and using good parts in early sequence. A heuristic method called 'variance algorithm' is used for this.

The objective of the problem is to minimize the variation in rate of consuming the parts of the sequence. The objective function and constraints are given to get solution. Since the problem is an integer non-linear problem, it is a NP-hard problem. However, some optimization software can solve the problem with quadratic objective function.

In just in time systems, a simple heuristic method called goal chasing method can be used in problem solving. Since the objective function is different within the scope of this problem, the algorithm has been revised without changing the impact of basic point. The goal chasing method is very simple and large scale problems can be solved with a small amount of time, regardless of the number of parts, models or demand. The biggest disadvantage of this method is myopia and being in tendency of using good units in early recurrence. If there are units meeting the required speed very good, the case that units are in early position, is possible. In a case like this, some models with special options will leave for later; in this case, a large variance will occur in units in the end of the sequence. There are studies for development of this method and three methods are mentioned. These are, symmetry, horizon, rate-preserving methods.

The goal chasing method is symmetric. Starting to application from the beginning or the end does not change anything. While creating order, units can be added forwards from first or backwards from last. In this method, good or poor units do not create conglomeration in the beginning or end of the sequence. The disadvantage of this method is, poor units are in the middle of the sequence. Therefore, symmetry method could not provide the adequate development.

In order to reduce the 'short-sightedness' in horizon method, it can be accepted that more than one position in the sequence are in each iteration. 2 feasible units with minimum cost are being selected and only of these is placed in the sequence.

Also, the algorithm is re-arranged according to this. The implement of this method is easy.

In rate-preserving method, poor units are deposited after a few iterations; so that when the structure of the remaining units is compared with the original structure, there will be more poor and less good. This method tries to minimize the distance between the initial composition and sequenced units and protect the structure of original units. Also this method has 'look-ahead' property and its computation load is not very large compared with horizon improvement but larger than symmetry improvement.

The most basic problem of goal chasing method is, good units are being used quickly in early iterations and bad units remain to the end of iterations. There is not a quantitative measurement to determine which units are good or bad. Therefore, the variance method is developed to determine good or bad units. This development is used everywhere where goal chasing method is used, as well as mixed-model assembly lines.

Vilarinho and Simaria (2002) focused on balancing mixed-model assembly lines with parallel workstations by using the two-stage heuristic method.

In this study, it is mentioned that production rate may decrease because of the constraint that the cycle time is not shorter than the longest task time. It is explained that production rate will increase, manufacturing flexibility will be provided and cycle time will be shorter than the longest task time with paralleling workstations.

In this two stage procedure, simulated annealing approach is used. The procedure is trying to find solutions to the minimizing number of stations purpose according to the cycle time given in the first stage. In the second stage, looking for a solution to the providing a workload balance purpose between workstations.

In order to ensure the first purpose, i.e. minimize workstations, mathematical programming model has been created. This model also meets the second purpose with ensuring a balanced workload.

Just because the mathematical model given in the study cannot solve the optimal result, simulated annealing approach is used. First, simulated annealing approach was described in the general structure. After that, the simulated annealing approach including two-stage procedure is given. In both stages a simulated annealing approach was used.

In the first stage of the approach, the initial solution is determined. So then, the rank positional weight heuristic that can be adapted to the mixed model assembly line is used. In solution evaluation criterion, when the balance delay time is minimized, it is defended that purpose of minimizing the number of stations in the first stage is provided. In neighboring solutions, transferring of task from one station to another and swapping of tasks between stations are performed. One of these movements, the transferring due to reduces workstation number, is more effective than swapping in minimizing balance delay time.

In the second stage, for the number of workstations determined in the first stage, it is mentioned that balancing of workload between-stations and within-stations simultaneously is aimed. The initial solution of the second stage is the final solution of the first stage. At this stage, as in the first one, swapping and transferring movements work. But unlike the first stage, swapping movement is more efficient in this stage.

Vilarinho and Simaria (2006) focused on balancing mixed-model assembly lines with parallel workstations by using the ant colony optimization algorithm.

According to the contents of the study, while meeting the demand considering constraints during the assembly line balancing, minimizing the cost is also important.

In the study, balancing problem type 1 is fictionalized. In other words, minimizing the number of workstations that tasks will be assigned to in the given cycle time is aimed. Ant colony optimization is used for this. Meanwhile, zoning constraints and parallel workstations are considered.

The author defended that ant colony optimization has more effective results referring to the success of the simulated annealing study in year 2002.

In the study, ant colony optimization approach is described in detail. In this approach, problem is solved based on insect societies' behavior. The behavior of real ant colonies has the instinct of finding the shortest path between nests and food sources. This instinct is triggered with the pheromone traces that the places are passed through by other ants. Ants are more likely to follow the trails where they are heavy. Ants are more likely to follow the trails where they are intensive. On continuation of the study, ant colony optimization algorithm is described for the type 1 of mixed model assembly line balancing problems. It has been mentioned the applicable constraints to this type. The feasible balancing solution algorithms are outlined. Here, solution quality was evaluated according to the operator location and line efficiency. According to the study, in each sub-colony iteration, pheromone amount should be updated. Algorithm has been tested and the results were evaluated according to the specified parameters.

Xu and Xiao (2008) have balanced mixed-model assembly lines according to fuzzy operation times and drifting operations. The aim during this balancing is, minimizing the total work overload time.

In assembly lines, parts are processed in stations by moving with a sort of transport system on the line. Since the current market is intensely competitive and consumer-centric, tendency to the production of mixed-model products is quite high. Thus mixed-model assembly line balancing problem is to assign the operations to an ordered sequence of stations such that precedence relations of each model are satisfied and some performance measures are optimized. In addition, balancing is extremely getting hard in the case of uncertain operation times.

There are uneven operation times within the study. In addition, the operator drifts with the operation. When the operation is completed, operator leaves the part and walks through the previous station. If the operator cannot complete his work in required time, he walks through the next station with the part and work overload occurs.

In the problem definition, assumptions for mixed-model assembly lines have been mentioned. Ones which the problem influenced from these assumptions are described. Iterative relations have been given and proved. Here it is concluded that total work overload occurs in the planning period process. Since the model is too complicated and includes fuzzy uncertainties, total work overload has been resolved with computer simulation method.

In addition to these, fuzzy simulation and genetic algorithm are integrated, so hybrid intelligent algorithm is designed and its algorithm is created for the resolving model. Results are evaluated and compared by entering various samples to the software.

Esmailian, Ismail, Sulaiman, Ahmad and Hamedi (2009) focus on assigning and balancing of tasks to workstations as long as target purposes are provided. Mixed model production balancing problem usually is transformed into a single model line-balancing problem to solve. But in this study, mixed-model problem has not been turned into single-model problem, and the settlement has been done by arranging it as mixed products on the parallel assembly lines.

This paper gives a heuristic method for finding out an initial assignment of mixed-model tasks to workstations in the parallel assembly lines in order to minimize cycle time for each model as an initial solution for meta-heuristic mixed model parallel assembly line. This research shows the modification of the mixed model production into parallel assembly line and an algorithm can be used for more than two products together with different cycle times.

A successful mixed-model parallel assembly line includes the solution of how many lines and work-stations are needed. At the same time, it also has the solution about how to assign tasks in models to work-stations on the parallel assembly line. The goal of this paper is to present a heuristic procedure to assign mixed-model assembly lines tasks to parallel assembly lines and create an initial balancing of the mixed-model parallel assembly lines for using as the initial solution of meta-heuristic method.

One of the objectives of this paper is to assign i -th task of m -th model for each model to k -th workstation of h -th line of parallel assembly lines and create initial solution for using meta-heuristic methods in order to minimize cycle time for each product and balance the parallel lines.

In the beginning of the flow that will be used for line balancing, firstly, the product that will be produced on one of the lines is chosen. Then, minimum number of stations related to chosen product and tasks to be possibly assigned to these stations according to precedence are determined. Then, tasks are assigned to workstations in sequence. If total time of the workstation does not exceed the cycle time, assignment is done, if exceeds, assignment is done by opening a new station. When all assignments are done, improvement quantity is calculated. These output performance measures were obtained by using MATLAB® software. The Arcsusi1's test problem was selected to check the performance of proposed algorithm and illustrate how the proposed heuristic works.

Yağmahan (2011) focused on balancing mixed-model assembly lines by using multi-objective ant colony optimization approach.

This study considers the aim to minimize smoothness index and the balance delay for the cycle time given in mixed-model assembly lines. The multi-objective ant colony optimization algorithm is used in the solution of this problem.

Ant colony optimization is explained in detail like a new meta-heuristic approach. It is mentioned that they find the shortest path between food sources and nests by using the pheromone without a visual cue.

The other hands, mixed-model assembly line balancing problems are processed as single model assembly line balancing problems by combining precedence diagrams of models mostly.

In this work, a multi-objective ant colony optimization algorithm was proposed to solve mixed-model assembly line balancing problem type 1. Applicable constraints and algorithms of this method are issued.

CHAPTER THREE

THE METHODOLOGY FOR SOLVING MMALBP

3.1 Basic Information about the Mixed-Model Line Balancing Problem

Various objectives can be mentioned in assembly line balancing like as minimizing station number, minimizing cycle time, reducing equipment cost. But, as mentioned before, it is not possible to perform all objectives at once. The main objective of our problem is, minimizing the number of workstations on the assembly line. In addition to this, while focusing on to achieve the main goal, using manpower and machine capacity at the highest level, minimizing idle times, distributing idle times properly between workstations are considered.

For the product is produced on the line that is addressed in the problem, the customer demand is quite high. It is not suitable to use single-model lines in production of products with high customer demand. Thus, these products are produced in mixed-model lines. Therefore, the main purpose of our problem is MALBP-1, ie. in assignment of mixed-models to stations problem, can be determined as minimizing the number of stations when the cycle time is given.

In assembly lines which mixed model products are produced, balancing of assembly lines can be difficult due to high product range. In other words, products on assembly lines have different features and these features are produced with different operations.

In the assembly line considered within the problem, the biggest reason for bottleneck and starvation is the wide range of products manufactured in the same line. Operation times applied to these products are different from each other. Cycle time of the line is calculated by correlating the planning period and customer demand amount in this period. When the assignment within the problem is considered, bottleneck does not occur when there is product order with no-option in the line. But if there is a product order that has one or a few of those options, the bottleneck is

inevitable. In this case, providing the order to the customer in time is done by changing line speed. As a result, inefficient and poor quality production, low line efficiency and delivery delay occurs.

3.2 The Problem Statement

In the assembly line that has been located as U-shaped and works as straight-line under the evaluation, flow is provided continuously with conveyor. Therefore the line is paced line. Here the line provides an operation stopping at stations along the operation time that is determined according to arrangement, and moves to the next station when the time is completed. It will be assumed that feeder lines and supplementary units do not create a problem in providing products. Feeder lines and supplementary units are not included in production processes, but inspection station is included in process where the quality control happens.

The constant loading rate that is generally applied in paced lines is even applied in our problem. This feature lets us know whether the system functioning creates a bottleneck or starvation or not. Thus, the compliance in workload distribution between stations can also be seen.

In order to overcome bottlenecks occurred and starvation, U-shaped line can be converted into 2 parallel lines. Products with very different characteristics can be produced in different lines. Thus, line imbalances can be avoided in the line. New lines that will be created have same characteristics. The cycle time of the line and equipment to be used in the line may differ from each other according to products that will be produced in the line. In this study, removing bottleneck stations with minimum number of stations is planned by converting the straight line working as U-shaped, into the parallel line.

In the assembly line considered, there are some constraints which may affect workload distribution. These are priority relations, cycle time restriction, works desired to be assigned to the same station and works not desired to be assigned to the same

activities. Our problem runs as “task related constraint” according to these constraints.

A lot of assumptions are considered in assembly line balancing. One of these is, accepting operation times deterministic unless otherwise stated. Operation times in the content of our problem are evaluated as deterministic.

Just in Time (JIT) has revolutionized the manufacturing world. Enterprises focus on implementing JIT and gaining profit from loss of stock. JIT means is producing the required product, in required time, with required amount. JIT implementation is generally used in mixed-model assembly lines.

In Martur, JIT implementation is used. Thus, the product that customer wants is produced at any time, any model with any amount. End product stock losses and semi-finished stock losses are not in question. There should not be line balancing losses in production lines. However, in order to gain profit fully with JIT concept, there should not be balancing losses in production lines.

In order to reduce line balancing losses in production lines, it is needed to eliminate bottleneck and starvation cases.

In order to overcome problems in the assembly line, the line is re-designed and balanced. Due to many differences between models, operation times and operations are different from each other. Considering these differences, in the new design, the existing single-line is converted into 2 parallel lines. In this case, similar models are assigned to each line. After determining models predicted for the assignment to lines, neighborhood search method is used in order to balance lines that will also be discussed under the “the methods used to solve the problem” subject heading in detail. The results obtained here and current situation simulation study are compared and interpreted.

3.3 The Methods Used to Solving the Problem

The problem within the scope of the study corresponds to type 1 (MALB1) of the mixed model assembly line balancing studies. So, minimizing the number of stations will be aimed when the cycle time is given. In order to achieve this aim, problem is solved with the help of an algorithm. The simulation method may be used in order to control suitability to the system of the problem solved.

3.3.1 The Algorithm Used to Solving the Problem

Because of assembly line balancing problems' design, seeking a solution to the problem with the help of optimal methods will bring some difficulties along, especially when the problem size extended. For this reason, using heuristic methods for solving these kinds of problems may be more meaningful and realistic. Heuristic methods do not guarantee the optimal solution, but under certain constraints, can relatively supply good and valid solutions with a less calculation.

In this section, a heuristic method is presented for mixed-model assembly line balancing problems. This offered method is same as the neighborhood search technique that developed by Bukchin, Dar-El, Rubinovitz (2002) for mixed-model assembly lines.

It is aimed to minimize the number of stations by holding the cycle time constant including the study. Besides, it is focused on eliminating the bottleneck. It is aimed to achieve more effective results for this purpose and assignment process continues until reaching the most efficient result. The moment that a better solution cannot be obtained, it is assumed that the best solution has been obtained. Bukchin, Dar-El, Rubinovitz (2002)'s study was consulted since it has the content to meet the purpose of our problem.

Bukchin, Dar-El and Rubinovitz (2002) developed a heuristic that targets to minimize the number of stations according to determined cycle time for modelling mixed-model assembly lines according to the order.

In this study it is mentioned that the developed approach requires much more versatile employees. It may even be assumed that employees should have the ability to do any work on the line. Thus, having different employees do the same work for different models may be obtained. According to this, it is stated that two ways can be followed in assignment of tasks to lines. The first of these; a specific task is required to be assigned to the same station for all models, and the second one; a specific task may be assigned to different stations for different models.

The solution procedure in the article consists of three main stages. The first of these; is obtaining the initial solution in order to determine positions (E set) of tasks to be assigned to the same station for different models. In the second one, considering positions determined at the first stage, the balancing (L set) is made separately for every model type. At this stage; the assignment is performed considering tasks to be assigned to different stations for different models. At the last stage; neighborhood search method is used. And at this stage, the result is obtained by increasing the variability positions of tasks that required to be assigned to the same station for different models. In development of the solution, “bottleneck” measure is evaluated. When it is reached to the optimum positioning, an average cycle time is obtained by running the simulation. This cycle time is compared with the required cycle time and it is decided to whether accept the solution or not.

In this article, the goal of the study is firstly mentioned at the model identification stage. The goal of the study is; minimizing the number of stations for the given cycle time, in assignment of different models’ tasks to stations. In addition, differences between traditional models and the model within the scope of the study are explained with various assumptions. These are;

1. Precedence diagrams of all model types can be accumulated into a single combined precedence diagram.
2. Each task of the combined precedence diagram is performed for at least one model.
3. Task duration is known and depends on the model type.
4. Asynchronous line pace, as well as blockage and starvation are possible.
5. The first station is never starved and the last station is never blocked.
6. The line production policy is 'make-to-order'.
7. A task that is common to several models can be restricted in certain cases to the same station for all models (Bukchin, Dar-El & Rubinovitz, 2002).

The seventh of assumptions above, indicates the main difference between traditional mixed-model assembly line balancing methods and the proposed method. Here it is defended that, same tasks with different models can be assigned to different stations. So, high idle times that will be probably occurred may be avoided. But since this situation requires duplicated machines, it may cause high equipment costs.

Formulation of the problem is given within the scope of the study. According to this formulation, objective function is maximizing line throughput. Here the objective function is the implicit objective of every balancing problem. In addition, there are constraints within the scope of the study as; each task is assigned to only one station, all predecessors are assigned to either this task's station or a station earlier, each model time at each station is calculated, a set of tasks required to be assigned to the same station for different models, is created. The objective function in problem formulation composes of two elements as; assignment of tasks to stations and sequence of model arrivals to line. Model arrivals are according to given demand proportions. In other words, probability of each model to be the next model that enters to the line depends on the average periodic demand. On the other hand, in order to evaluate the objective function, it is stated that a good performance measure may result more quickly and easily than simulation.

In the contents of the article, five performance measures are mentioned as; smoothed station measure, minimum idle time measure, station coefficient of variation measure, model variability measure and bottleneck measure. Among these, bottleneck measure creates the superior performance measure for short assembly lines. Therefore, in order to development of maximizing throughput aim, bottleneck measure is examined in detail. Bottleneck measure is reversible with throughput. According to this measure, maximum station time cannot be less than the assembly time of the station that has the maximum assembly time. Bottleneck measure is combined with the balancing algorithm in order to compare it to the algorithm that will be presented.

The purpose of the algorithm to be presented is; minimizing the number of stations that will meet production capacity. But a cycle time is given for this. Firstly, balancing is tried with lowest required number of stations by using balancing procedure and minimizing the cycle time is tried meanwhile. In the end of the balancing procedure, an average cycle time is obtained by running the simulation. If this cycle time is smaller than or equal to required cycle time, the procedure is completed, if bigger than, it is continued to iteration and increasing the number of stations by one, until the feasible solution is found. Figure 3.1 shows the flowchart of the algorithm. In this figure, N is stated to number of workstation (lowest required), C is stated to required cycle time, C_s is stated to resultant cycle time.

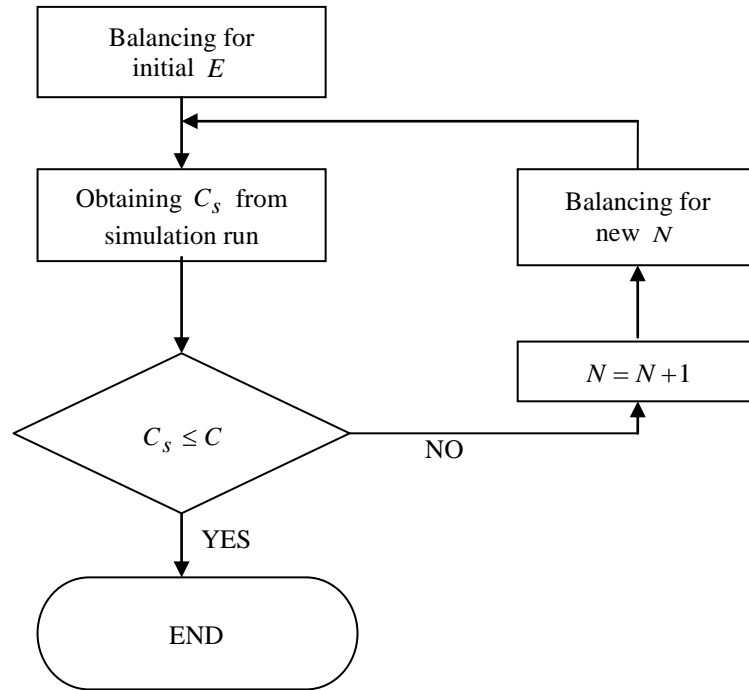


Figure 3.1 General flowchart of the algorithm.

It is previously stated that, solution procedure composes of three main stages. Steps of balancing procedure including these three stages are given in figure 3.2.

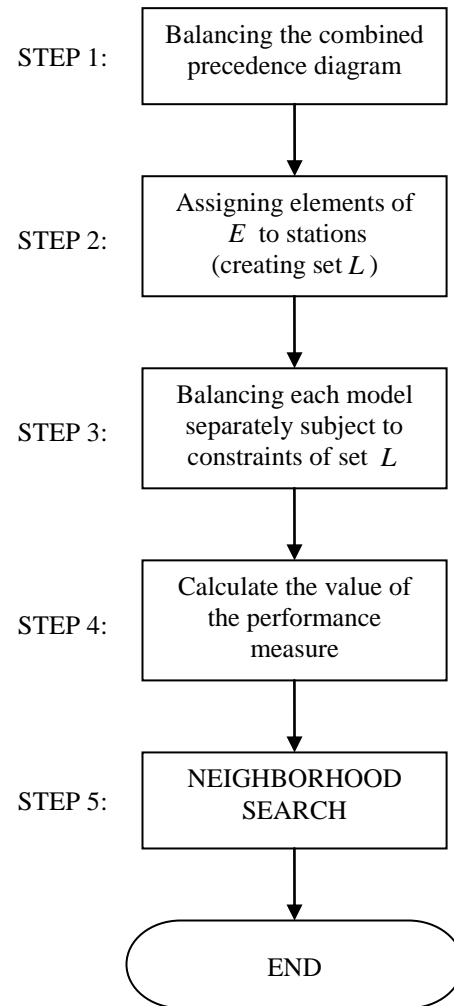


Figure 3.2 The balancing procedure.

At the first stage, combined precedence diagram is formed and balanced in order to determine E set. At this stage, it is tried to minimize the cycle time by holding the number of stations constant. In combined diagram, task times are calculated according to the weighted average considering the demand size. At the second stage, the formation of L set is provided by assignment of tasks to stations in E set. At the third stage, balancing is made separately for each model under L set constraints. It is tried to minimize the cycle time by holding the number of stations constant even at this stage. At the fourth stage, the performance measure value is calculated. At the fifth and last stage, neighborhood search is developed.

Neighborhood search find alternative solutions by shifting tasks required to be assigned to the same station for different models, between neighbor stations. For comparison between these alternative solutions, value of bottleneck measure is used.

According to the article content, calculated cycle time and value of bottleneck measure as a result of assignment obtained at the first stage, will be a decision measure for next calculations. Assignment result obtained from the third stage, composes initial solution for next assignments. New assignments are carried out by using shifting principle that is shifted task between neighbor station and the value of bottleneck measure is calculated. If the new performance measurement value is better than the value determined at the first stage, assignment rankings that this performance measure is obtained from, becomes initial solution for the new assignment, if not better, these calculations are eliminated. If there is no assignments having better performance measurement value than assignments obtained, the algorithm is terminated and the best solution obtained is determined as the final solution.

Consider the neighborhood search algorithm. Let e_1, e_2, \dots, e_n denote the tasks of set E . Let L_s denote the seed, which holds the locations of the tasks of set E ($l_{s_1}, l_{s_2}, \dots, l_{s_n} \in L_s$), and L_j denote solution j , received during the algorithm execution. Let O_s denote the seek performance measure as bottleneck. The neighborhood procedure operates as follows (see the neighborhood search flow chart in Figure 3.3):

Step 1: Define the initial seed: $L_s = L$.

Step 2: Set initial values to variable: $i = 1, j = 0$.

Step 3: If task e_i is assigned to the last station, or, there is k , in which task e_k succeeds task e_i , and, $l_{s_k} = l_{s_i}$, then go to Step 8.

Step 4: Create a new feasible solution: $j = j + 1, L_j = L_s$.

Step 5: Shift task e_i in solution j forward by one station: $l_{j_i} = l_{j_i} + 1$.

Step 6: Balance each model separately, subject to L_j constraints.

Step 7: Calculate the value of the ‘bottleneck’ measure associated with the obtained solution: $O_j = f(L_j)$.

Step 8: If task e_i is in the first station, or, there is k , in which task e_k precedes task e_i , and, $l_{s_k} = l_{s_1}$, then go to Step 13.

Step 9: Create a new feasible solution: $j = j + 1, L_j = L_s$.

Step 10: Shift task e_i in solution j backward one station: $l_{j_i} = l_{j_i} - 1$.

Step 11: Balance each model separately, subject to L_j constraints.

Step 12: Calculate the value of the ‘bottleneck’ measure associated with the obtained solution: $O_j = f(L_j)$.

Step 13: If $i \leq n : i = i + 1$, go to step 3.

Step 14: If $\min_j \{O_j\} \geq O_s$: End procedure with L_s and O_s .

Step 15: Define the best neighbor as the new seed: $L_s = \{L_k | O_k = \min_j \{O_j\}\}$.

Step 16: Define a new value for the performance measure: $O_s = O_k$, go to step 2 (Bukchin, Dar-El & Rubiovitz, 2002).

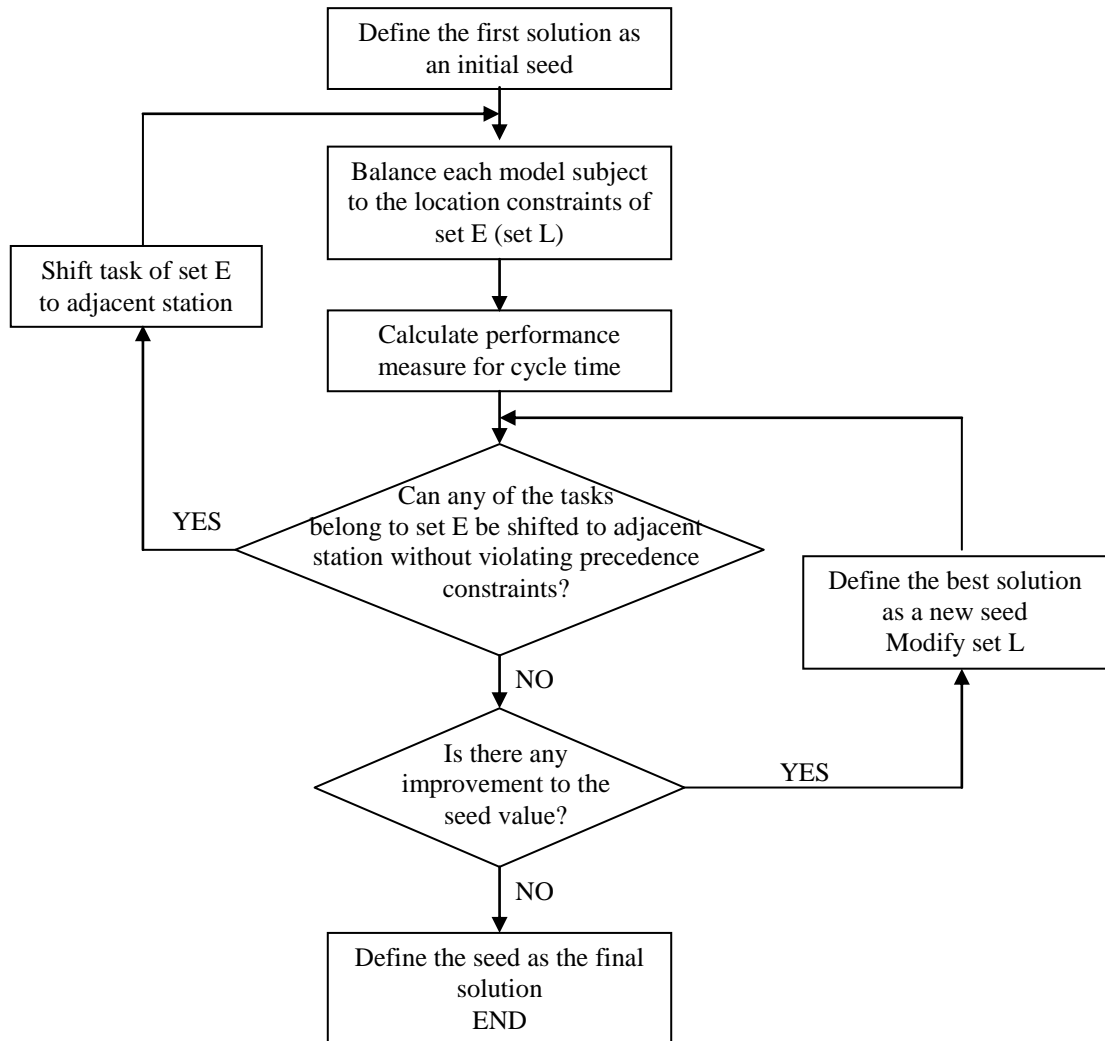


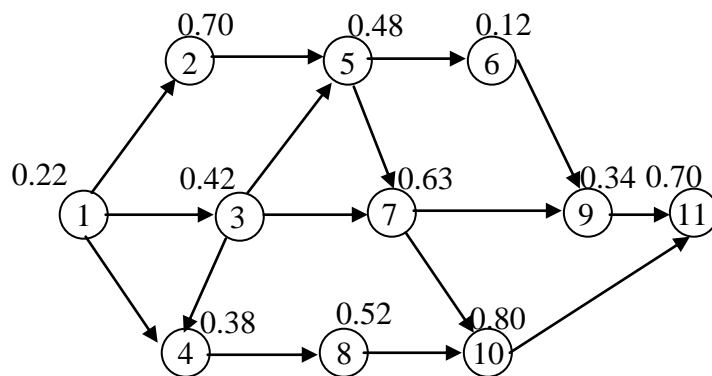
Figure 3.3 The neighborhood search.

The neighborhood search will be used even within the scope of the thesis. First three stages in the article will not be used for the initial solution, positional weight method that is going to be mentioned in further chapters will be used. Bottleneck performance measure is chosen as evaluation measure even within the scope of thesis. Therefore, bottleneck measure calculated at the fourth stage in the article, will also be calculated within the scope of the thesis. Also, the algorithm of the article will be used within the scope of application.

3.3.2 Ranked Positional Weighted Method

This method is also known as Helgeson – Birnie Method. In this method, each work item is weighted over time value. For weighting, sum of the work item's operation time and duration of works after that on the line and required works to pass through this work item are calculated (UZUN, 2008).

Table 3.1 The precedence diagram sample



For the line given in Table 3.1, the position weight of work item no.1 is the sum of its time and times of other work items it affects. Work item no. 1 affects work items no. 2, 3 and 4 and creates the formation of work items no. 5, 6, 7, 8, 9, 10 and 11 respectively. In this case, the sum of all work items' time values creates the position weight of the work item no. 1.

When it comes to work item no. 5, the position weight value is, the sum of its time and times of other work items no. 5, 6, 7, 9, 10 and 11 after it, which it also affects.

Table 3.1 shows the time value of each work item and which work items it is connected to. From this point of view, we should calculate the position weight of a work item. More details are as follows:

For 1 : (1) + (2) + (3) + (4) + (5) + (6) + (7) + (8) + (9) + (10) + (11) = 5,31

For 2 : (2) + (5) + (6) + (7) + (9) + (10) + (11) = 3,77

For 3 : (3) + (4) + (5) + (6) + (7) + (8) + (9) + (10) + (11) = 4,39

For 4 : (4) + (8) + (10) + (11) = 2,40

...

For 10 : (10) + (11) = 1,50

For 11 : (11) = 0,70

After calculating these values, create the following table.

Table 3.2 The ranking by positional weight

Tasks	Positional Weight	Time	Relation
1	5.31	0.22	0
3	4.39	0.42	1
2	3.77	0.70	1
5	3.07	0.48	2-3
7	2.47	0.63	3-5
4	2.40	0.38	1-3
8	2.02	0.52	4
10	1.50	0.80	7-8
6	1.16	0.12	5
9	1.04	0.34	6-7
11	0.70	0.70	9-10

On this table, position weights of each work item are created and listed in descending order. Position weights according to the value of line's cycle time are considered with the following algorithm and work item assignment is performed to required stations.

1. Find the work item that has the biggest position weight and not loaded yet, and assign it to the new station. Equalize the station time with this work item time.
2. For the idle time at the work station, find a next work item that has not been selected from the list yet.

3. If all connection values of this work item (on the table above) have been previously placed on any station AND the value of this work item does not exceed the idle time at the selected station, go through step no.4, IF NOT, go through step no.2.
4. Add the selected work item time to the station time. Assign this work to this station. Go through step no.2.
5. If there is not any other station to be added, it means the workstation is completed. Go back to step no.1 in order to go through the next station.
6. If all work items are assigned, END.

Efficiency measurement of all these assignments is as follows.

$$=100 * (1 - (\text{cycle time} * \text{number of stations} - \text{total production time}) / (\text{cycle time} * \text{number of stations}))$$

CHAPTER FOUR

APPLICATION

4.1 Factory Information

The application study under the thesis is implemented at Turkey's independent automotive supply industry company MARTUR Automotive Seating and Interiors. MARTUR Automotive Seating and Interiors produces commercial vehicle (automobile) seats.

Martur, founded in 1983, serves with 2200 employees with eight factories at Bursa and Kütahya. Martur has a large and reliable customer list, including FIAT, Peugeot, Citroen, Renault, Ford, Toyota, Hyundai and MAN. Martur also provides competitive, qualified products and service as soon as possible with a definitive customer satisfaction, with its vertical integrated enterprise structure dominating the entire value chain.

There are three main supply sources including Martur as; ready-made parts obtained from sub-industry, sub-industries under the Üstünberk holding and in-plant production. Production in the main factory consists of frame welding, frame painting and assembling departments.

Most of semi-finished products needed for seat production within the under the Üstünberk holding are produced in companies shaped as sub-industry.

Wires and pipes of the frame used on the seat are formed in İndesta Aş. Belonging to holding and shipped to Bestal Aş. and Martur frame department that are also belonging to holding. Metal sheet formed in Bestal Aş. and shipped to Martur frame department. Products coming to Martur frame department are welded and shipped to Martur painting department. Painted products at painting department provide input as semi-finished products to Martur assembly lines.

Foams used on the seat provide semi-finished product input to Martur assembly lines by being produced in Fompak Aş. under the Üstünberk holding.

Slipcover packing the upper surface of the seat are produced from fabrics which are manufactured in Martur Textile Aş. belonging to holding. Produced fabrics are subjected to fabric dyeing process in Berk Aş. that is also belonging to holding. Finishing process is carried out on dyed fabrics. Then this fabric is passed through the lamination. Fabrics are cut in accordance with templates of the relevant seat in the cutting section at Kütahya Slipcover Manufacturing Factory and sewing operation is carried out. Sewed slipcovers provide input as semi-finished product to Martur assembly department.

All these products produced within the Üstünberk holding and products supplied from independent sub-industries are shipped to the customer with Just in Time (JIT) operating system by producing seats in Martur assembly department.

4.1.1 General Information about the Main Section

Martur carries out MCV (Mini Cargo Vehicle) and D200 (Linea) seat production in two separate places as front seat line and rear seat line. The place where rear seats are produced in is cell type production place and conveyor system is used in order to carry semi-finished parts between stations. In the place where front seats are produced, an assembly line is used which has 34 assembly platforms in total on the conveyor and is formally U-type but functionally straight.



Figure 4.1 Production place of the rear seat.



Figure 4.2 Production place of the front seat.

Products manufactured are shipped to the customer with Just in Time (JIT) operating system. The customer informs seat product information of the automobile planned to be produced, 6 hours before the automobile arrives to the station where seat assembly is done on the customer assembly line. When the product information goes to Martur through the system, product starts to prepare in semi-finished product lines. The seat from the assembly line is taken to the finished product standby place,

called magazine. When the shipment time comes for finished products, loading to trailer trucks with automatic belts and intelligent loading systems is provided. Finished product delivery to the customer is carried out with trailer trucks with a capacity of 48 sets. One set has 2 front seats and 1 rear seat. The rear seat is optional, only front seats (without rear seats) may be ordered for some automobiles.

4.1.2 Introduction of the Front Seat Production Line

Martur, the place where MCV (Mini Cargo Vehicle) and D200 (Linea) front seats are produced is an assembly line which has 34 assembly platforms in total on the conveyor and is formally U-type but functionally straight. Workers on this line, carry out operations on one side of the line (See figure 4.3).



Figure 4.3 The product considered within the thesis.

Martur adopts qualified production as a principle and checks all products 100%. For this reason, there is quality control station covering 3 assembly platforms where aesthetic controls are done before packing station. Before this, there is also a test unit station that function tests are performed (See figure 4.4).



Figure 4.4 Final control station.

In order to feed front seat assembly line, there is one feeder line and two supplementary units where semi-finished products are created. Backrest of front seat is produced on the feeder line. Required option information comes to the computer in the beginning of the feeder line. Backrest is produced in accordance with this option information and provides input as semi-finished backrest product to the main assembly line (See figure 4.5).



Figure 4.5 The feeder line that is produced backrest of front seat.

For both supplementary units, information is obtained from separate computer systems. In the first one of units, foam and slipcover are assembled with hook and this operation is called hook (See figure 4.6).



Figure 4.6 The supplementary units that is hooked slipcover to foam.

Semi-finished products from the cell that hook operation is carried out are transported to the main assembly lines by transportation racks. The transport operation is done with the transportation vehicle which is adjusted automatically. This vehicle moves on magnetic tapes laid on the ground (See figure 4.7).



Figure 4.7 The transportation vehicle moves on magnetic tapes.

In the second one of supplementary units, slide assembly that allows the seat move back and forth is done (See figure 4.8).



Figure 4.8 The supplementary units that is assembled slide to frame.

Semi-finished products are transported to the main assembly line in the same way. This means that semi-finished products are hanged to transporting racks according to the order row and transported with the transport vehicle moving on magnetic tapes.

According to this thesis, line balancing study will be applied on the main assembly line by assuming that feeder lines and supplementary units send product to production line just in time.

4.2 Description of Problem

The front line which is formally U-type but functionally straight and included to the operation extend of the enterprise is evaluated and mixed-model products are produced in this line. Balancing that is applied to the line, does not have constant equipment constraint. The changes cost that may occur, is not much to prevent carried out the assignment.

Line balancing study is intended to be done in mixed-model parallel assembly lines for production of products on this line. Because of the product range, bottlenecks and idle times may occur in order to catch cycle time. Production of similar products in the same line with new parallel lines that will be applied, thus, ensuring efficiency increase is aimed by providing variability in line speeds.

Type 1 of mixed-model assembly line balancing studies (MALB1) will be implemented to the application. So, minimizing the number of stations will be purposed when the cycle time is given. This is because cycle time cannot be changed in order to meet amount of the customer product demand. In the case it is changed, problems and bottlenecks will occur during shipment.

The methodology required to be applied in solution of the problem is as follows;

1. Creation of detailed task definitions of tasks applied on the line under the study,
2. Determining precedence relations of tasks on the basis of the model,
3. Performing a time study by making at least 10 observations for each task definition on the basis of the model,
4. Performing the modelling study of front line system with the obtained data, on the Arena Simulation program,
5. Creating initial solution of the proposed model with the positional weight method,
6. Matlab coding using the algorithm given by using initial solution and obtaining assignments on the new model proposed,
7. Modelling the proposed new front line model with assignments obtained on the Arena simulation program and evaluation,
8. Observing differences of new model results in comparison with the previous model,
9. Final assessment.

4.2.1 Data Collection

Data were collected from the front line of MCV-D200 seat models in MARTUR Automotive Seating and Interiors

4.2.1.1 Computing of the Planning Period

In MARTUR Automotive Seating and Interiors, there are 2 shifts as between 06:30-16:00 and 16:00-01:30, and each shift works 9.5 hours. In 9.5 hours, there is a half-hourly lunch break and 2 quarter tea breaks. In other words, since the working time of each shift is 8,5 hours, 17 hours (1020 minutes) is Martur' s daily work time. In addition, the factory works every day except Sundays.

Since there is minimum 26 days per month and 17 hours working per day, planning period can be calculated as $PT=1591200$ sec.

4.2.1.2 Information about Model

Products manufactured in Martur front seat line consists of 20 seat models. 10 of these models are MCV seats and 10 models are D200 seats. Options composing seat models are; tilt mechanism feature, side bag (airbag), heater, porthole, armrest feature for MCV seats, tilt mechanism feature, side bag, heater, electrical porthole for D200 seats. On seats, while there can be none of these features, there also can be more than one feature. These features create the seat models by coming together in various combinations. These combinations are given in Table 4.1.

Table 4.1 MCV- D200 types of options

MCV SEAT OPTIONS					
TYPE	TILT MECHANISM	SIDEBAG	HEATER	ARMREST	PORTHOLE
Aa	X	X	X	X	X
Ab	X	X	√	X	X
A1a	X	√	X	X	X
A1b	X	√	√	√	X
A2a	√	X	X	√	√
A2b	√	X	√	√	√
A2c	√	X	√	X	√
A3a	√	√	X	√	√
A3b	√	√	√	√	√
A3c	√	√	√	X	√

D200 SEAT OPTIONS				
TYPE	TILT MECHANISM	SIDEBAG	HEATER	ELECTRICAL PORTHOLE
Aa	X	X	X	X
Ab	X	X	√	X
A1a	X	√	X	X
A1b	X	√	√	X
A2a	√	X	X	X
A2b	√	X	√	X
A3a	√	√	√	√
A3b	√	√	X	X
A3c	√	√	√	X
A3d	√	√	X	√

4.2.1.3 The Amount of Product-based Customer Demand

The amount of product-based customer demand during the planning period (PT=1591200 sec.) is given in Table 4.2.

Table 4.2 The amount of product-based customer demand

TYPE	MODEL	DESCRIPTION CODE OF PRODUCT	TOTAL	
MCV	Aa	WITHOUT OPTION	5397	26368
	Ab	HEATER	810	
	A1a	SIDE BAG	6246	
	A1b	SIDE BAG+HEATER+ARMREST	416	
	A2a	TILT MECHANISM+ARMREST+PORTHOLE	5131	
	A2b	TILT MECHANISM+HEATER+ARMREST+PORTHOLE	166	
	A2c	TILT MECHANISM+HEATER+PORTHOLE	132	
	A3a	TILT MECHANISM+SIDE BAG+ARMREST+PORTHOLE	7950	
	A3b	TILT MECHANISM+SIDE BAG+HEATER+ARMREST+PORTHOLE	118	
	A3c	TILT MECHANISM+SIDE BAG+HEATER+PORTHOLE	2	
D200	Aa	WITHOUT OPTION	3114	6059
	Ab	HEATER	60	
	A1a	SIDE BAG	2102	
	A1b	SIDE BAG+HEATER	28	
	A2a	TILT MECHANISM	191	
	A2b	TILT MECHANISM+HEATER	109	
	A3a	TILT MECHANISM+SIDE BAG+HEATER+ELECTRICAL PORTHOLE	263	
	A3b	TILT MECHANISM+SIDE BAG	13	
	A3c	TILT MECHANISM+SIDE BAG+HEATER	1	
	A3d	TILT MECHANISM+SIDE BAG+ELECTRICAL PORTHOLE	178	
TOTAL PRODUCTION NUMBER FOR A MOUNTH			32427	

Product-based customer demands show variety. When the amount of customer demands is considered, it is remarked out that models with heater option are demanded less. In seats with heater option, 7 extra operations are operated.

4.2.1.4 Definition of the Tasks in Assembly

In the application, product models that will be produced in the front-line include different operations (tasks). There are totally 61 tasks that are provided different product types. Although these tasks are basically the same, according to model they

can show variety in terms of structural as well as in terms of time. These tasks can be explained on the basis of the model as follows.

Task 1: Taking seat frame from rack and placing frame to the platform; frames prepared as semi-finished product and carried on racks. The frame that is taken from rack is placed to the platform on the line. This task is applied to all models.

Task 2: Swiping label; the product model is introduced to the line. This task is applied to all models.

Task 3: Inserting slide stopper; is the stopper assembly. This task is applied to all models.

Task 4: Turning slipcover that is hooked to foam and leaving it on the frame; is the preparation operation for assembly. This task is applied to all models.

Task 5: Dressing seat foam to the frame; is the positioning operation of foam and frame. This task is applied to all models.

Task 6: Inserting dowels into holes of slipcover side plastics; is the preparation operation for assembly. This task is applied to all models.

Task 7: Inserting dowels into the frame; is the dowel assembly. This task is applied to all models.

Task 8: Rotating the frame (180°); is the preparation for the next operation. This task is applied to all models.

Task 9: Placement of dressed backrest on the seat; is the preparation operation for assembly. This task is applied to all models.

Task 10: Backrest-seat label verification; is the control whether the backrest to be assembled to seat comes as the right product or not from feeder lines to the main assembly line. For this, both seat label and backrest label should be matched by swiping them. This task is applied to all models.

Task 11: Running cables coming from backrest through the seat frame; is the assembly of heater cable, side bag cable, heater & side bag cable and electrical porthole cable. One of the cables is selected according to the option and assembled. Operation times of cables assembly show variety from each other depending on the model. Models without cable are also available.

Task 12: Fixing screws of backrest by hand; is preparation operation for assembly. This task is applied to all models.

Task 13: Inserting porthole bottom cover and porthole cover; is the porthole plastics assembly. This task is applied to only MCV type with porthole option.

Task 14: Torching backrest-seat merging screws; is the fixing operation of seat and backrest. This task is applied to all models.

Task 15: Lifting platform; is the preparation for the next operation. This task is applied to models with the heater option.

Task 16: Fixing heater cable to pullmaflex with dowels; is the cable fixing operation. This task is applied to models with heater option. If the model has side bag optionally, heater-side bag should be fixed, but in this case operation times also change.

Task 17: Landing platform; is the preparation for the next operation. This task is applied to models with heater option.

Task 18: Rotating platform (90°); is the preparation for the next operation. This task is applied to all models.

Task 19: Fixing heater cable to the frame with dowels; is the operation that provides come of cable close to the button. This task is applied to models with heater option.

Task 20: Inserting heater clip to the plastic casing; is the assembly of heater cable to the button in casing. This task is applied to models with heater option.

Task 21: Inserting outer plastic casing to the frame; is the outer plastic casing assembly. This task is applied to all models.

Task 22: Inserting keiper plastic; is the keiper plastic assembly. This task is applied to all models.

Task 23: Fixing the screw of tilt adjustment lever by hand; is the preparation operation for assembly. This task is applied to models with tilt mechanism.

Task 24: Rotating platform (180°); is the preparation for the next operation. This task is applied to all models.

Task 25: Inserting inner plastic casing to the frame; is the inner plastic casing assembly. This task is applied to all models.

Task 26: Rotating platform (90°); is the preparation for the next operation. This task is applied to all models.

Task 27: Lifting platform; is the preparation for the next operation. This task is applied to all models.

Task 28: Screwing backrest screws of inner and outer casing; is the screw assembly. This task is applied to all models.

Task 29: Fixing side bag cable to pullmaflex with dowels; is the cable fixing operation. This task is applied to models with side bag option but without heater option.

Task 30: Fixing the screw of safety belt buckle by hand; is the preparation operation for assembly. This task is applied to all models.

Task 31: Fixing the safety belt cable to pullmaflex by dowels; is the cable fixing operation. This task is applied to all models.

Task 32: Binding with zip tie; is the operation of rounding up all cables in model's contents with 2 zip ties. This task is applied to models except the model without option and operation times do not change when the model changes.

Task 33: Binding cables to slide handle with yellow zip tie; is the operation performed in order to block the movement of all cables. This task is applied to all models.

Task 34: Fixing safety belt cable with clip-on zip ties; is the cable fixing operation. This task is applied to all models.

Task 35: Cutting surpluses of zip ties; is the zip tie cleaning operation. This task is applied to all models.

Task 36: Landing platform; is the preparation for the next operation. This task is applied to all models.

Task 37: Screwing porthole cover screw; is the cover fixing operation. This task is applied to only MCV type with porthole option.

Task 38: Inserting porthole cover cork; is applied to only MCV type with porthole option.

Task 39: Rotating platform (180°); is the preparation for the next operation. This task is applied to all models.

Task 40: Screwing front screw of outer casing; is the screw assembly. This task is applied to all models.

Task 41: Torching safety belt screw; is the screw torque operation. This task is applied to all models.

Task 42: Torching tilt adjustment lever screw; is the screw torque operation. This task is applied to models with tilt mechanism.

Task 43: Marking safety belt screw with torque pen after torching; is an application for controlling. This task is applied to all models.

Task 44: Marking tilt adjustment lever screw with torque pen after torching; is an application for controlling. This task is applied to models with tilt mechanism.

Task 45: Rotating platform (90°); is the preparation for the next operation. This task is applied to models with armrest.

Task 46: Assemble armrest; is the armrest assembly. This task is applied to models with armrest.

Task 47: Inserting armrest pin; is the armrest fixing operation. This task is applied to models with armrest.

Task 48: Rotating platform (90°); is the preparation for the next operation. This task is applied to models with armrest.

Task 49: Iron; is the steaming operation in order to remove wrinkle. This task is applied to all models.

Task 50: Safety belt buckle test; is the current control test. This task is applied to all models.

Task 51: Side bag test; is the current control test. This task is applied to models with side bag option.

Task 52: Heater test; is the current control test. This task is applied to models with heater option.

Task 53: Electrical porthole test; is the current control test. This task is applied to models with electrical porthole option.

Task 54: Sticking the test label; is the operation of labeling that indicates statuses of all tests except safety belt test. This task is applied to all models except the model without option.

Task 55: Final control; is aesthetic and functional control operation. This task is applied to all models.

Task 56: Packaging; is the product packaging operation. This task is applied to all models.

Task 57: Headrest shafts lubrication; is the lubrication operation in order to ease the movement of headrest. This task is applied to all models.

Task 58: Assemble headrest; is the headrest assembly. This task is applied to all models.

Task 59: Swiping label; the product model is introduced to the shipping department. This task is applied to all models.

Task 60: Sticking label; is the product description operation. This task is applied to all models.

Task 61: Rotating platform (90°); is the preparation of product for shipment. This task is applied to all models.

Tasks that applied to all models constitute the general structure of the product. Remaining tasks provide the creation of variable product models by changing product option. This means that some tasks may not apply to the model while applying to other models. This situation provides the diversification of product and consisting of 20 new type models. Model-based task precedence relations diagrams are given between Table 4.3 and Table 4.22 in the Appendices A.

4.2.1.5 Time Study about Processing Times

A time study is performed with at least 10 observations for each task description on the basis of model. Average operation times of all models about MCV and D200 is given in Table 4.23 in the Appendices A.

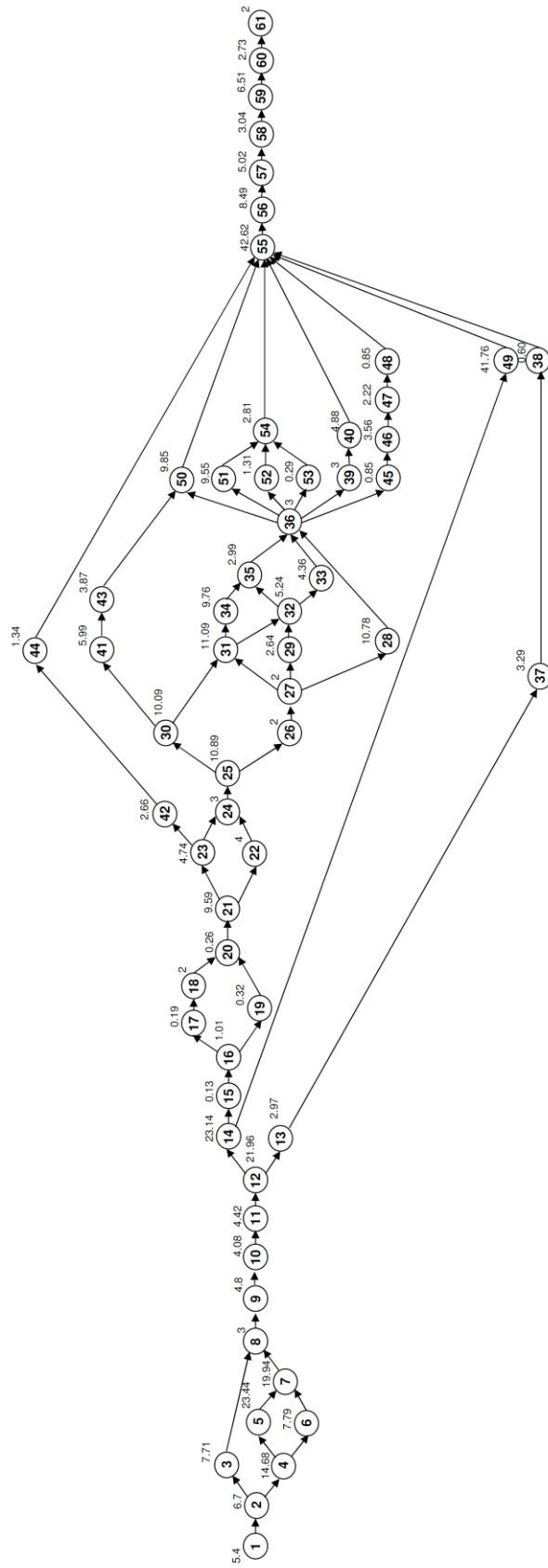
4.2.2 Information about the Current Line

Arrangement of the current situation was examined under the following headings.

4.2.2.1 The Joint Precedence Diagram for Current Line

When evaluating the current assembly line production, all models that produced in this assembly line, are considered and joint precedence graph is created. Joint precedence graph consisting of these models is given in Table 4.24.

Table 4.24 Joint precedence diagram for current line



It is stated before in the “description of problem” section, that the front line which is formally U-type but functionally straight is evaluated in the current situation. Manufactured products in this line are mixed model and have a fixed cycle time. This cycle time is calculated in a way to meet customer demand and not to cause idle time or bottleneck during shipment. According to identified criteria, assignments of tasks to stations are made.

4.2.2.2 Computing Cycle Time of Current Line

As mentioned in “data collection” section, planning period is calculated as $PT=1591200$ second. Planning period should be associated with the amount of products produced during this period, in order to calculate current situation cycle time. The amount of products produced during planning period and their distribution according to models are given in Table 4.25.

Table 4.25 The amount of products produced during planning period and their distribution according to models

Type	Model	Number of products	Total	Time	Number of products*time	Total durations of the current line (h)	Total number of products of the current lines (pcs)
MCV	Aa	5397	8511	369	1991493	3775.91	32427
D200	Aa	3114		366.9	1142526.6		
MCV	A1a	6246	8348	411.1	2567730.6		
D200	A1a	2102		412.9	867915.8		
MCV	Ab	810	870	437.5	354375		
D200	Ab	60		438.4	26304		
MCV	A1b	416	444	478.4	199014.4		
D200	A1b	28		458.8	12846.4		
MCV	A2a	5131	5322	420.2	2156046.2		
D200	A2a	191		385.7	73668.7		
MCV	A2b	166	275	493.7	81954.2		
D200	A2b	109		457.4	49856.6		
MCV	A2c	132	132	475.5	62766		
MCV	A3a	7950	8213	468.7	3726165		
D200	A3a	263		500.8	131710.4		
MCV	A3b	118	131	515	60770		
D200	A3b	13		429.8	5587.4		
MCV	A3c	2	3	497	994		
D200	A3c	1		479.9	479.9		
D200	A3d	178	178	455.4	81061.2		

CYCLE TIME OF CURRENT LINE

= planning period / the total number of product in planning period

$$\begin{aligned} \text{CYCLE TIME OF CURRENT LINE} &= 1591200 \text{ (sec)} / 32427 \text{ (pcs)} \\ &= 49.07 \text{ (sec / pcs)} \end{aligned}$$

When it is evaluated in terms of shipping; there is not any of bottleneck or case that will cause missing shipment due to 49-second cycle time. This case is also evaluated and supported in the following way:

It is described in the “general information about the main section” part that trailer trucks carry 48 set of seats. In other words, one trailer truck can carry 96 front seats and maximum 48 rear seats (Rear seats may not be in every automobile.)

Distribution of the production amount per month on the basis of model and shift are given in Table 4.26 in the Appendices A.

AVERAGE AMOUNT OF PRODUCTION PER DAY

= average amount of production per month / number of shifts per month * number of shifts per day

AVERAGE AMOUNT OF PRODUCTION PER DAY

$$\begin{aligned} &= 32427 \text{ (pcs)} / 52 \text{ (shift)} * 2 \text{ (shift/day)} \\ &= 1247,192308 \text{ (pcs / day)} \end{aligned}$$

$1248 \text{ (pcs / day)} / 96 \text{ (trailer truck / pcs)} = 13 \text{ (trailer truck / day)}$ times, 1 truck must move from Martur for shipment.

Daily working times calculated as 17 hours in “data collection” section corresponds to 1020 minutes.

$1020 \text{ (min)} / 13 \text{ (trailer truck)} = 78.46 \text{ (min / trailer truck)}$ Trailer trucks must move from Martur every 78.46 minutes for shipment. In other words, in 78.46 minutes (4707.69 sec) 48 set, which is 96 front seats must produce.

According to this support, the cycle time that will meet customer demand and not create a problem in shipment is determined as 49 seconds.

4.2.2.3 Assignment Tasks to Workstations and Bottleneck Stations

In the current situation, 61 tasks are assigned to 12 stations in total. Assignment of tasks to stations is as follows;

Station1={1, 2, 3, 4,}

Station2={5, 6}

Station3={7, 8}

Station4={9, 10, 11, 12, 13}

Station5={14}

Station6={15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27}

Station7={28, 29, 30, 31, 32, 33, 34, 35, 36}

Station8={37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48}

Station9={49}

Station10={50, 51, 52, 53, 54}

Station11={55}

Station12={56, 57, 58, 59, 60, 61}

Assignment of tasks to stations according to the current assignment and operation times of stations and tasks are given in Table 4.27.

Table 4.27 Operation times of stations and tasks in the current line

DESCRIPTION OF TASKS	CURRENT CONDITION				
	task	time	precessor	station time	station number
TAKING SEAT FRAME FROM RACK AND PLACING FRAME TO THE PLATFORM	1	5,40	0	34,49	Station1
SWIPING LABEL	2	6,70	1		
INSERTING SLIDE STOPPER	3	7,71	2		
TURNING SLIPCOVER THAT IS HOOKED TO FOAM AND LEAVING IT ON THE FRAME	4	14,68	2		
DRESSING SEAT FOAM TO THE FRAME	5	23,44	4	31,23	Station2
INSERTING DOWELS INTO HOLES OF SLIPCOVER SIDE PLASTICS	6	7,79	4		
INSERTING DOWELS INTO THE FRAME	7	19,94	5-6	22,94	Station3
ROTATING THE FRAME (180°)	8	3,00	3-7		
PLACEMENT OF DRESSED BACKREST ON THE SEAT	9	4,80	8	38,24	Station4
BACKREST-SEAT LABEL VERIFICATION	10	4,08	9		
RUNNING CABLES COMING FROM BACKREST THROUGH THE SEAT FRAME	11	4,42	10		
FIXING SCREWS OF BACKREST BY HAND	12	21,96	11		
INSERTING PORTHOLE BOTTOM COVER AND PORTHOLE COVER	13	2,97	12	23,14	Station5
TORCHING BACKREST-SEAT MERGING SCREWS	14	23,14	12		
LIFTING PLATFORM	15	0,13	14	40,13	Station6
FIXING HEATER CABLE TO PULLMAFLEX WITH DOWELS	16	1,01	15		
LANDING PLATFORM	17	0,19	16		
ROTATING PLATFORM (90°)	18	2,00	17		
FIXING HEATER CABLE TO THE FRAME WITH DOWELS	19	0,32	16		
INSERTING HEATER CLIP TO THE PLASTIC CASING	20	0,26	19		
INSERTING OUTER PLASTIC CASING TO THE FRAME	21	9,59	20		
INSERTING KEIPER PLASTIC	22	4,00	21		
FIXING THE SCREW OF TILT ADJUSTMENT LEVER BY HAND	23	4,74	21		
ROTATING PLATFORM (180°)	24	3,00	22-23		
INSERTING INNER PLASTIC CASING TO THE FRAME	25	10,89	24	59,95	Station7
ROTATING PLATFORM (90°)	26	2,00	25		
LIFTING PLATFORM	27	2,00	26		
SCREWING BACKREST SCREWS OF INNER AND OUTER CASING	28	10,78	27		
FIXING SIDEBAG CABLE TO PULLMAFLEX WITH DOWELS	29	2,64	27		
FIXING THE SCREW OF SAFETY BELT BUCKLE BY HAND	30	10,09	25		
FIXING THE SAFETY BELT CABLE TO PULLMAFLEX BY DOWELS	31	11,09	27-30		
BINDING WITH ZIP TIE	32	5,24	29-31		
BINDING CABLES TO SLIDE HANDLE WITH YELLOW ZIP TIE	33	4,36	32		
FIXING SAFETY BELT CABLE WITH CLIP-ON ZIP TIES	34	9,76	31		
CUTTING SURPLUSES OF ZIP TIES	35	2,99	32-34	33,11	Station8
LANDING PLATFORM	36	3,00	28-33-35		
SCREWING PORTHOLE COVER SCREW	37	3,29	13		
INSERTING PORTHOLE COVER CORK	38	0,60	37		
ROTATING PLATFORM (180°)	39	3,00	36		
SCREWING FRONT SCREW OF OUTER CASING	40	4,88	39		
TORCHING SAFETY BELT SCREW	41	5,99	30		
TORCHING TILT ADJUSTMENT LEVER SCREW	42	2,66	23		
MARKING SAFETY BELT SCREW WITH TORQUE PEN AFTER TORCHING	43	3,87	41		
MARKING TILT ADJUSTMENT LEVER SCREW WITH TORQUE PEN AFTER TORCHING	44	1,34	42		
ROTATING PLATFORM (90°)	45	0,85	36	23,81	Station10
ASSEMBLE ARMREST	46	3,56	45		
INSERTING ARMREST PIN	47	2,22	46		
ROTATING PLATFORM (90°)	48	0,85	47		
IRON	49	41,76	14	41,76	Station9
SAFETY BELT BUCKLE TEST	50	9,85	36-43	23,81	Station10
SIDEBAG TEST	51	9,55	36		
HEATER TEST	52	1,31	36		
ELECTRICAL PORTHOLE TEST	53	0,29	36		
STICKING THE TEST LABEL	54	2,81	51-52-53	42,62	Station11
FINAL CONTROL	55	42,62	38-40-44-48-49-50-54		
PACKAGING	56	8,49	55	27,78	Station12
HEADREST SHAFTS LUBRICATION	57	5,02	56		
ASSEMBLE HEADREST	58	3,04	57		
SWIPING LABEL	59	6,51	58		
STICKING LABEL	60	2,73	59		
ROTATING PLATFORM (90°)	61	2,00	60		
		419,20			

As can be seen from the table; a bottleneck is observed at station number 7. According to this, the line does not move in 49 seconds, it moves in 59.95 seconds. When the line efficiency is calculated according to determined cycle time as 49-second, the value % 71.29 is obtained.

LINE EFFICIENCY

$$= 100 * (1 - (\text{cycle time} * \text{number of stations} - \text{total production time}) / (\text{cycle time} * \text{number of stations}))$$

EFFICIENCY OF CURRENT LINE

$$= 100 * (1 - (49 * 12 - 419.2) / (49 * 12))$$

$$= \%71.29$$

When the line efficiency is calculated according to real cycle time as 59.95-second, the value % 58.27 is obtained.

EFFICIENCY OF CURRENT LINE

$$= 100 * (1 - (59.95 * 12 - 419.2) / (59.95 * 12))$$

$$= \%58.27$$

The efficiency calculated is very low value. In addition, customer demand cannot be met with the current production capacity and problems in shipment occur.

4.2.3 Information about the Proposed Line

The operating line that U-type placed and works as straight line in the current situation, is proposed to separate into 2 parallel lines in the new situation. According to the study will be evaluated through two parallel lines.

4.2.3.1 Separating into 2 Parallel Lines

Product models are grouped in itself to be assigned to parallel lines. This grouping is done by being separated into 4 groups as;

- 1- Option less, total production time low
- 2- Option less, total production time high
- 3- Option more, total production time low
- 4- Option more, total production time high

Features used in assignment of these groups to 2 separate lines are as follows;

- 1- Since more operations are applied to seats with heater option, models with heater option are not assigned to one of the lines.
- 2- Total production numbers of models that assigned to 2 separate lines by considering production quantities, are considered to be close each other.
- 3- It is considered that sum of the multiplication of total production times and production quantities of models are required to be close to each other, on both two lines.

In accordance with specifications above, models are assigned for 2 lines parallel to each other. Models assigned to lines, their production times, total production amount in the planned period, total times of groups and total production times of lines are given in Table 4.28.

Table 4.28 The assignment models to parallel lines

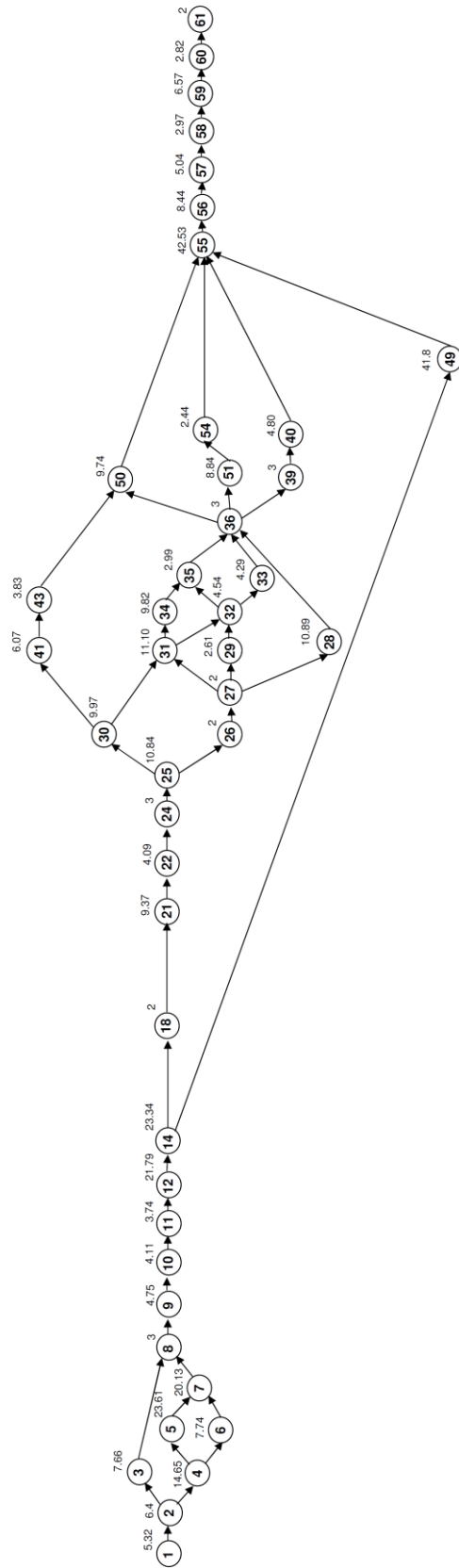
Product group	Type	Model	Number of products	Total	Total product	Time	Number of products*time	Total time(sec)	Total time (min)	Total time (h)	Total durations of 2 parallel lines (h)	Total number of products of 2 parallel lines
Group 1 (item)	MCV	Aa	5397	8511	16859	369	1991493	6569666	109494.4	1824.907	1824.907222	16859
	D200	Aa	3114			366.9	1142526.6					
	MCV	A1a	6246	8348		411.1	2567730.6					
	D200	A1a	2102			412.9	867915.8					
Group 2 (item)	MCV	Ab	810	870	1314	437.5	354375	592539.8	9875.663	164.5944		
	D200	Ab	60			438.4	26304					
	MCV	A1b	416	444		478.4	199014.4					
	D200	A1b	28			458.8	12846.4					
Group 3 (item)	MCV	A2b	166	275	791	493.7	81954.2	388531.1	6475.518	107.9253	1950.999833	15568
	D200	A2b	109			457.4	49856.6					
	MCV	A3c	2	3		497	994					
	D200	A3c	1			479.9	479.9					
	MCV	A2c	132	132		475.5	62766					
	MCV	A3b	118	118		515	60770					
	D200	A3a	263	263		500.8	131710.4					
Group 4 (item)	MCV	A3a	7950	7950	13463	468.7	3726165	6042528.5	100708.8	1678.48		
	MCV	A2a	5131	5131		420.2	2156046.2					
	D200	A2a	191	191		385.7	73668.7					
	D200	A3b	13	13		429.8	5587.4					
	D200	A3d	178	178		455.4	81061.2					

As can be seen from the above table, group 1 is assigned to one of the lines (the line without heater option), 2., 3. and 4. group are assigned to the other line. While the total amount of production of line 1 is 16859 (pcs/month), the total amount of production of line 2 is 15568 (pcs/month). While the total production time of line 1 is 1824.91 (hour), the total production time of line 2 is 1951 (hour).

4.2.3.2 The Joint Precedence Diagrams for Each Parallel Line

Assignment to the parallel line, is required the creation of two separate precedence diagram for two line. The precedence diagram of Line 1 is given in Table 4.29 and the precedence diagram of Line 2 is given in Table 4.30.

Table 4.29 The joint precedence diagram for line 1



4.2.3.3 Computing Cycle Time of Parallel Lines

In order to compute cycle time of lines; line 1 and line 2, planning period should be associated with the total amount of products that is produced line-based during this period. The amount of products produced during planning period and its distribution according to models are given in Table 4.28. According to this, cycle times of lines can be computed as follows.

$$\begin{aligned}\text{CYCLE TIME OF LINE 1} &= 1591200 \text{ (sec)} / 16859 \text{ (pcs)} \\ &= 94.38 \text{ (sec / pcs)}\end{aligned}$$

$$\begin{aligned}\text{CYCLE TIME OF LINE 2} &= 1591200 \text{ (sec)} / 15568 \text{ (pcs)} \\ &= 102.21 \text{ (sec / pcs)}\end{aligned}$$

When it is considered in terms of shipping; the case in which the time period between 2 seats produced is 94 seconds in line 1 and 102 seconds in line 2, does not cause any bottleneck according to joint graph and average task times or a case that will cause missing shipment. When model times are assigned to stations, bottleneck can occur. This case is also evaluated and supported in the following way:

In order to meet the customer demand, the need of producing 1 seat in every 49 seconds is calculated in the “computing cycle time of current line” section. But each produced seat cannot be shipped to the customer right out of the production line and be held in the inventory called magazine until the product accumulates for 1 trail truck. 196 seconds are needed for 4 seats to be produced. The sum of the cycle time of both two lines is 196 seconds. Thus, 4 seats are sent to magazine in shipment area in 146 seconds.

According to this support, the cycle time that will meet customer demand and not create a problem in shipment is determined as 94 (sec), for line 1 and 102 (sec) for line 2.

4.2.3.4 Initial Solution with Positional Weight Method

Initial solution for both two lines separately must be created in order to use in the algorithm that will be adapted to matlab program. The positional weight method is used in creation of this solution.

15 tasks of 61 tasks that applied in total are not applied in line 1. These tasks are excluded from the application and positional weighting method is applied. The positional weight of each task is calculated for this. While calculating positional weight for each task, all operation times of tasks that are operated after relevant task and own time of relevant task are summed up. This value creates positional weight of relevant tasks. At the same time, weighted average of all tasks of models assigned to line 1 is calculated in itself. Thus, task times are created belong to the line. Tasks listed in descending order according to their positional weights, are assigned to stations in turn. One of the issues to be considered while assigning, is that the total time of tasks assigned to the station cannot exceed cycle time of the line. Another one is; all predecessors are required to be assigned to stations before the task that will be assigned to the station. The assignment for line 1 that provides requirements above is given in Table 4.31.

Table 4.31 Assignment task to workstation for line 1 with positional weight method

PARALLEL LINE 1					
task	time	precessor	positional weight	station time	station number
1	5.32	0	389.68	93.26	Station1
2	6.40	1	384.36		
4	14.65	2	370.30		
5	23.61	4	347.91		
6	7.74	4	332.04		
7	20.13	5-6	324.30		
3	7.66	2	311.84		
8	3.00	3-7	304.18		
9	4.75	8	301.18		
10	4.11	9	296.42		
11	3.74	10	292.31		
12	21.79	11	288.57		
14	23.37	12	266.78		
18	2.00	14	201.61		
21	9.37	18	199.61		
22	4.09	21	190.24		
24	3.00	22	186.14		
25	10.84	24	183.14		
30	9.97	25	154.81	92.04	Station3
26	2.00	25	152.44		
27	2.00	26	150.44		
31	11.10	27-30	134.94		
29	2.61	27	116.63		
34	9.82	31	115.00		
32	4.54	29-31	114.01		
28	10.89	27	113.08		
49	41.80	14	112.18		
33	4.29	32	106.48		
35	2.99	32-34	105.18		
36	3.00	28-33-35	102.19		
39	3.00	36	99.19		
40	4.80	39	96.19		
41	6.07	30	90.02		
43	3.83	41	83.95		
51	8.84	36	81.66		
50	9.74	36-43	80.12		
54	2.44	51	72.82	70.38	Station5
55	42.53	40-49-50-54	70.38		
56	8.44	55	27.85		
57	5.04	56	19.41		
58	2.97	57	14.37		
59	6.57	58	11.39		
60	2.82	59	4.82		
61	2.00	60	2.00		
	389.68				

As can be seen from the table above, 5 stations are created. These stations are as follows.

$$S_{11} = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$$

$$S_{12} = \{ 10, 11, 12, 14, 18, 21, 22, 24, 25, 30 \}$$

$$S_{13} = \{ 26, 27, 28, 29, 31, 32, 33, 34, 35, 49 \}$$

$$S_{14} = \{ 36, 39, 40, 41, 43, 50, 51, 54 \}$$

$$S_{15} = \{ 55, 56, 57, 58, 59, 60, 61 \}$$

Station times are calculated as, $t_{s11} = 93.26$ sec, $t_{s12} = 92.29$ sec, $t_{s13} = 92.04$ sec, $t_{s14} = 41.17$ sec, $t_{s15} = 70.38$ sec. After this assignment, the assignment status is controlled with actual values of models and no bottleneck case has been found. Accordingly, the efficiency of this line is calculated as follows.

EFFICIENCY OF LINE 1

$$= 100 * (1 - (94 * 5 - 389.68) / (94 * 5))$$

$$= \%82.91$$

The assignment for line 2 that provides requirements mentioned above is given in Table 4.32.

Table 4.32 Assignment task to workstation for line 2 with positional weight method

PARALLEL LINE 2					
task	time	precessor	positional weight	station time	station number
1	5.48	0	451.16	97.73	Station1
2	7.03	1	445.68		
4	14.71	2	430.89		
5	23.26	4	408.34		
6	7.84	4	392.93		
7	19.74	5-6	385.08		
3	7.77	2	373.11		
8	3.00	3-7	365.34		
9	4.86	8	362.34		
10	4.05	9	357.49		
11	5.15	10	353.43	100.37	Station2
12	22.15	11	348.28		
14	22.88	12	311.84		
15	0.27	14	247.24		
16	2.11	15	246.97		
17	0.41	16	244.19		
18	2.00	17	243.79		
19	0.67	16	242.46		
20	0.55	19	241.79		
21	9.82	20	241.24		
23	9.87	21	227.53		
22	3.89	21	213.23		
24	3.00	22-23	209.34		
25	10.94	24	206.34		
30	10.21	25	178.06	Station3	
26	2.00	25	175.37	Station2	
27	2.00	26	173.37	Station2	
31	11.08	27-30	158.03	Station3	
29	2.67	27	139.93	Station2	
32	5.99	29-31	137.26	99.39	Station3
34	9.70	31	136.53		
28	10.67	27	134.51		
33	4.44	32	128.28		
35	3.00	32-34	126.84		
36	3.00	28-33-35	123.84		
49	41.72	14	112.15	82.84	Station4
39	3.00	36	105.24		
40	4.96	39	102.24		
41	5.91	30	90.23		
45	1.77	36	86.03		
13	6.19	12	84.73		
43	3.91	41	84.32		
46	7.43	45	84.26		
51	10.32	36	83.96		
50	9.97	36-43	80.41		
42	5.54	23	78.76		
37	6.85	13	78.53		
47	4.63	46	76.84		
52	2.74	36	76.38		
53	0.61	36	74.25		
54	3.20	51-52-53	73.64		
44	2.78	42	73.22		
48	1.77	47	72.21		
38	1.25	37	71.69		
55	42.73	38-40-44-48-49-50-54	70.44	70.44	Station5
56	8.55	55	27.71		
57	4.99	56	19.16		
58	3.11	57	14.18		
59	6.44	58	11.07		
60	2.63	59	4.63		
61	2.00	60	2.00		
	451.16				

As can be seen from the table above, 5 stations are created. These stations are as follows.

$S_{21} = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

$S_{22} = \{ 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 \}$

$S_{23} = \{ 28, 30, 31, 32, 33, 34, 35, 36, 49 \}$

$S_{24} = \{ 13, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 50, 51, 52, 53, 54 \}$

$S_{25} = \{ 55, 56, 57, 58, 59, 60, 61 \}$

Station times are calculated as, $t_{s21} = 97.73$ sec, $t_{s22} = 100.37$ sec, $t_{s23} = 99.39$ sec, $t_{s24} = 82.84$ sec, $t_{s25} = 70.44$ sec. After this assignment, the assignment status is controlled with actual values of models and no bottleneck case has been found. Accordingly, the efficiency of this line is calculated as follows.

EFFICIENCY OF LINE 2

$$= 100 * (1 - (102 * 5 - 451.16) / (102 * 5))$$

$$= \%88.46$$

As can be seen, the efficiencies both of two lines are very close to each other.

4.2.3.5 Improving Initial Solution with the Study Method

The initial solution is developed by using the neighborhood search method mentioned in part 3. For this, MATLAB (Matrix Laboratory) programming language is used. MATLAB software created under this study consists of one main coding and four function coding. In the main coding page, model information can be changed for each line. Thus, data of models belong to lines are filed separately. When the command of main coding page is entered to program, the program can provide the result with calculation in the function page to achieve necessary information. Main coding page and function coding are given in Appendices B. The program is run for separately for each line. The software did not find a better result than the initial solution for line 1 and line 2.

4.2.4 Simulation in ARENA and Comparison Results

The current situation and the new situation are modeled in ARENA simulation software. ARENA 10.0 is used during modeling. When models run, results obtained are compared respectively according to the total bottleneck time in the model, total production amount, idle time values, line efficiency, and the number of operators working.

4.2.4.1 Simulation of Current Model with ARENA

In ARENA simulation program, the current system is composed of 12 workstations and 20 waiting stations as well as input output stations. The block diagram including all station of current system base is given in Figure 4.9.

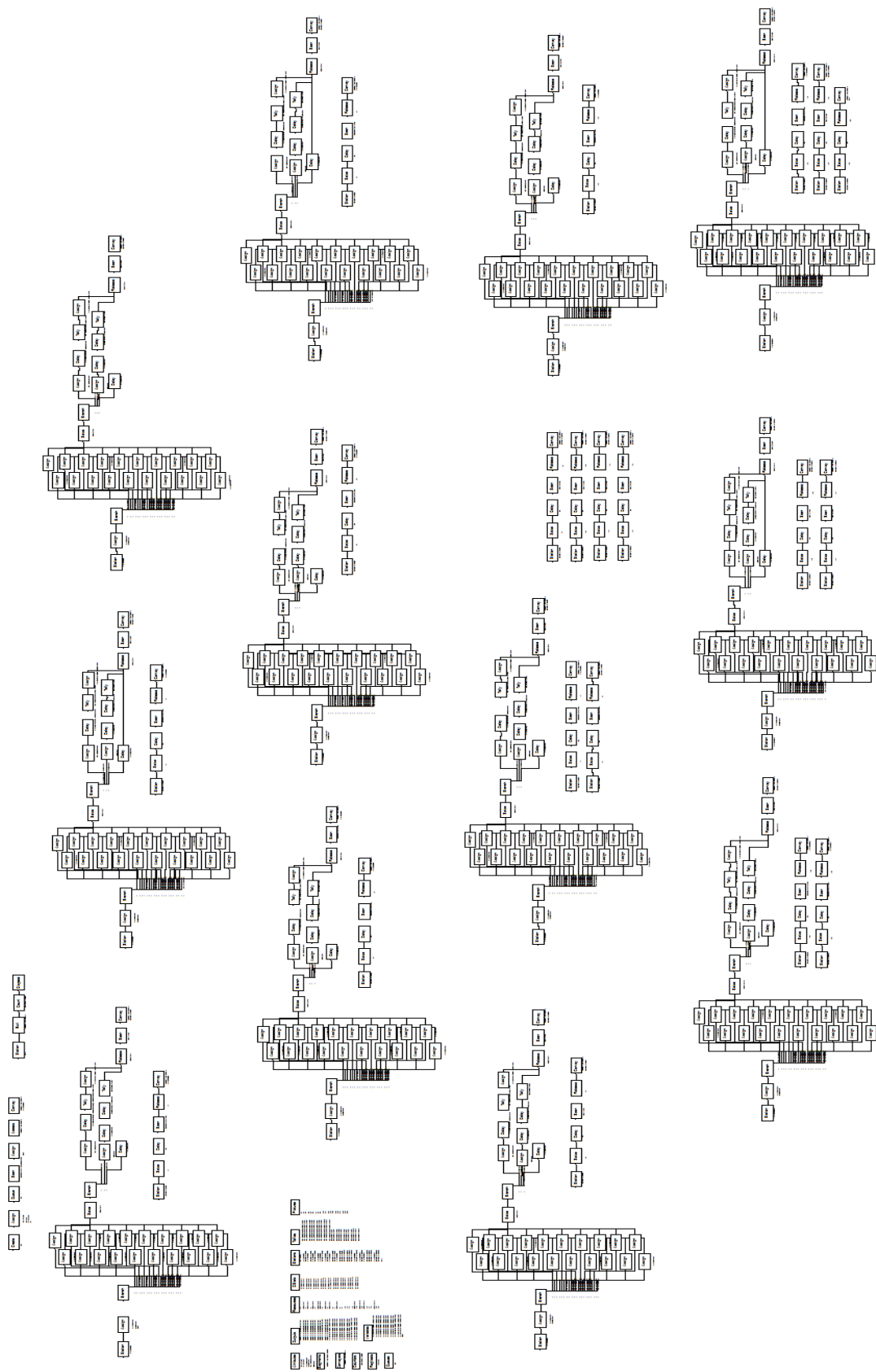


Figure 4.9 The block diagram of current model.

The simulation structure of the current model is going to be explained in four sections. The first of these sections is the entry part. In this section, entering the line frequency of products, the production distribution depending on the amount of production, constraints about products on the line and line information are given. The entrance block diagram is given in figure 4.10.

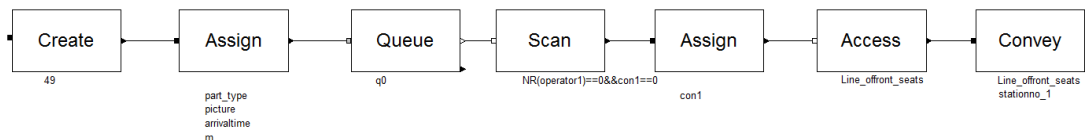


Figure 4.10 The entrance block diagram of current model.

The arrival frequency of entities going to be entering the line is determined as the cycle time with the ‘Create’ module. The cycle time is determined as 49 sec in the current situation. But, as mentioned in “Assignment tasks to workstations and bottleneck stations” part, the assembly line moves with the duration of the 7th station (59.95 sec) that has the maximum bottleneck time. For this reason, the current situation model in ARENA is run for two times with the cycle time 49 sec and 59.95 sec and results are obtained.

The tagging process that will provide convenience later on is carried out with the ‘Assign’ module. Attributes and variables are defined to the system with this module. In addition, entities belong to models going to enter the line and the distribution of these entities is given with this module. The distribution of these entities is determined as discrete distribution based on customer demand given in table 4.2.

The ‘Scan’ module checks the suitability of conditions in the system. In this system it is checked whether the operator at next station and the conveyor between stations are empty or not with “NR (operator1) ==0&&con1==0” condition sentence. If the conveyor and the station are empty, created entities enter the line and tend towards to the first station. Otherwise, these entities are kept in the queue with ‘Queue’ module.

'Access' module provides to entering the line of entities and the station where the entity is going to and the conveyor that it is going to use during this movement are defined with 'convey' module. In the current situation, there is a single conveyor system defined as 'Line_offront_seats' and the entity is moved to station 1 primarily in the entrance block diagram.

In the second section of the simulation structure, since the expression of 12 workstations is too long, just the structure of one of stations is described in general. The block diagram belong to the station 1 is given in figure 4.11.

The station definition is done with the 'Station' module, and the entrance of the entity to the station is provided. With the help of 'Assign' module, the command that the conveyor between the input station and the first station have come empty, is entered. Thus, the first condition is carried out for the entrance station releases the entity in the queue. The followed path by each model is defined with the 'Branch' module. There is 'Assign' module after every branch and the operation time belong to each model is entered as constant.

'Seize' module states that the person at the station keeps the work. The 'Branch' module used after this module provides the calculation of the measurement that is obtained according to the operation time of the model. If the cycle time is equal to operation time, the operation is done as long as the cycle time. If the cycle time is greater than the operation time, there is a bottleneck situation and bottleneck time must be calculated. In this case, operation is done during the cycle time and bottleneck times belong to models are obtained with 'Tally' module. With 'Assign' module, the total bottleneck time occurred at the station during the planning time is also calculated.

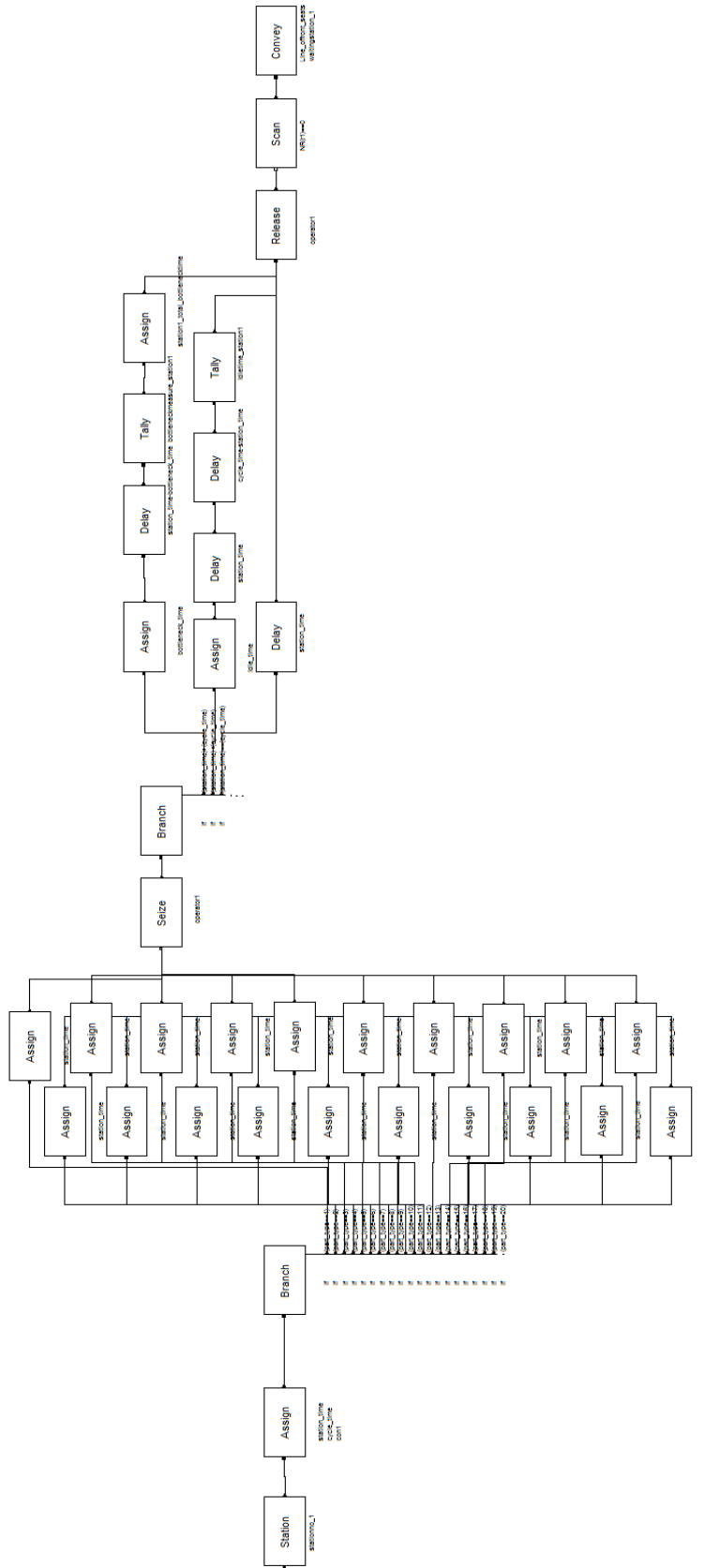


Figure 4.11 The station block diagram of current model.

If the cycle time is less than operation time, there is idle time and idle time must be calculated. In this case, operation is done as long as the operation time but entity does not leave the station until completing the cycle time. On the other hand the idle time belongs to models is obtained with ‘Tally’ module.

No matter which measurement is calculated at the end of the process, ‘Release’ module provides to releases the work by a person working at the station.

‘Scan’ module controls whether there is entity or not at the platform. “NR(r1)=0” condition sentence is used for this. Created entities when the station is empty, are tending towards to related stations. The relevant station is introduced to the ‘Convey’ module.

In the third section of the simulation structure, since the expression of 20 work stations is too long, the structure of one of stations is described in general. The block diagram belong to the waiting station 1 is given in the figure 4.12.

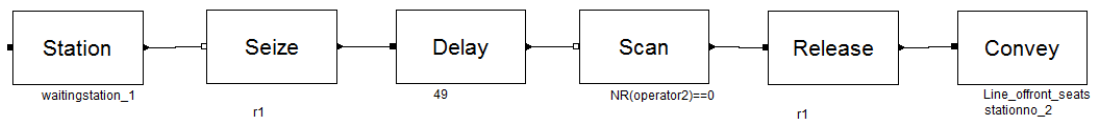


Figure 4.12 The waiting station block diagram of current model.

Although tasks of modules are same as described previously, the entity waits during the cycle time.

In the fourth section of the simulation structure, the exit of the entity from the line is provided. The exit block diagram is given in the figure 4.13.

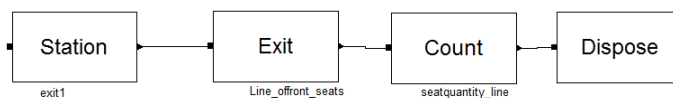


Figure 4.13 The exit block diagram of current model.

The 'Station' module provides the exit of the product from the line. It simulates the platform 34 in the current situation.

The 'Exit' module states that the product detaches from 'Line_offront_seats' line. Quantities of detaching products from the line are counted with the 'Count' module.

The 'Dispose' module states that the simulation is completed. This module can only be placed into the last section.

Results are obtained by running the program according to the required cycle time (49 sec). Production amount obtained here is close to the expected production amount. But the bottleneck time is extremely high. The summary of results obtained from the program run with 49 sec. cycle time, is given at tables 4.33, 4.34 and 4.35.

Table 4.33 Total number of production of planning period in ARENA (Cycle time = 49 sec)

Number of Production	30568
----------------------	-------

Table 4.34 Station bottleneck times in ARENA simulation software (Cycle time = 49 sec)

Cycle Time==49 sec.	Average Value	Observation	Total time
bottleneckmeasure_station1	0	0	0
bottleneckmeasure_station2	0	0	0
bottleneckmeasure_station3	0	0	0
bottleneckmeasure_station4	0	0	0
bottleneckmeasure_station5	0	0	0
bottleneckmeasure_station6	17,869	2021	36113,249
bottleneckmeasure_station7	10,952	30589	335010,728
bottleneckmeasure_station8	4,868	12530	60996,04
bottleneckmeasure_station9	0	0	0
bottleneckmeasure_station10	9,3218	985	9181,973
bottleneckmeasure_station11	0	0	0
bottleneckmeasure_station12	0	0	0
TOTAL BOTTLENECK TIME IN ALL STATION			441301,99

Table 4.35 Station idle times in ARENA simulation software (Cycle time = 49 sec)

Cycle Time==49 sec.	Average Value	Observation	Total time
idletime_station1	14,511	30600	444036,6
idletime_station2	17,771	30598	543757,058
idletime_station3	18,41	61190	1126507,9
idletime_station4	0	0	0
idletime_station5	25,863	30593	791226,759
idletime_station6	10,745	28570	306984,65
idletime_station7	0	0	0
idletime_station8	30,435	18056	549534,36
idletime_station9	7,2401	30579	221395,0179
idletime_station10	26,295	29592	778121,64
idletime_station11	6,3771	30575	194979,8325
idletime_station12	21,214	30571	648533,194
TOTAL IDLE TIME IN ALL STATION			5605077,011

The current situation model in ARENA runs again by changing the cycle time value with 59.95 sec that is time of bottleneck station and results are obtained. The summary of results obtained from the program run with 59.95 sec cycle time is given at tables 4.36, 4.37 and 4.38.

Table 4.36 Total number of production of planning period in ARENA (Cycle time = 59.95 sec)

Number of Production	24448
----------------------	-------

Table 4.37 Station bottleneck times in ARENA simulation software (Cycle time = 59.95 sec)

Cycle Time==59.95 sec.	Average Value	Observation	Total time
bottleneckmeasure_station1	0	0	0
bottleneckmeasure_station2	0	0	0
bottleneckmeasure_station3	0	0	0
bottleneckmeasure_station4	0	0	0
bottleneckmeasure_station5	0	0	0
bottleneckmeasure_station6	6,9305	1626	11268,993
bottleneckmeasure_station7	5,8586	14090	82547,674
bottleneckmeasure_station8	0	0	0
bottleneckmeasure_station9	0	0	0
bottleneckmeasure_station10	14,25	190	2707,5
bottleneckmeasure_station11	0	0	0
bottleneckmeasure_station12	0	0	0
TOTAL BOTTLENECK TIME IN ALL STATION			96524,167

Table 4.38 Station idle times in ARENA simulation software (Cycle time = 59.95 sec)

Cycle Time==59.95 sec.	Average Value	Observation	Total time
idletime_station1	25,463	24480	623334,24
idletime_station2	28,721	24478	703032,638
idletime_station3	29,36	48950	1437172
idletime_station4	0	0	0
idletime_station5	36,812	24473	900900,076
idletime_station6	21,711	22845	495987,795
idletime_station7	7,906	10379	82056,374
idletime_station8	26,971	24466	659872,486
idletime_station9	18,19	24459	444909,21
idletime_station10	36,458	24267	884726,286
idletime_station11	17,327	24455	423731,785
idletime_station12	32,163	24451	786417,513
TOTAL IDLE TIME IN ALL STATION			7442140,403

It is also mentioned in “Assignment tasks to workstations and bottleneck stations” part that the line efficiency decreases from %71.29 to %58.27 when the assembly line runs with 59.95 sec cycle time. As can be seen from above results, when the line runs with 59.95 seconds, the total bottleneck time decreases, and total production amount decreases %20 at the same time. Furthermore, operators working at stations with low operation time, wait for the expiration of 59.95 sec and increase idle time duration.

4.2.4.2 Simulation of Proposed Model with ARENA

In ARENA simulation program, the proposed system is modeled 2 separate assembly lines. There are five workstations for each of these lines, as well as input output stations. There is no waiting station in these lines. The block diagram including all information of program base is given in Figure 4.14.

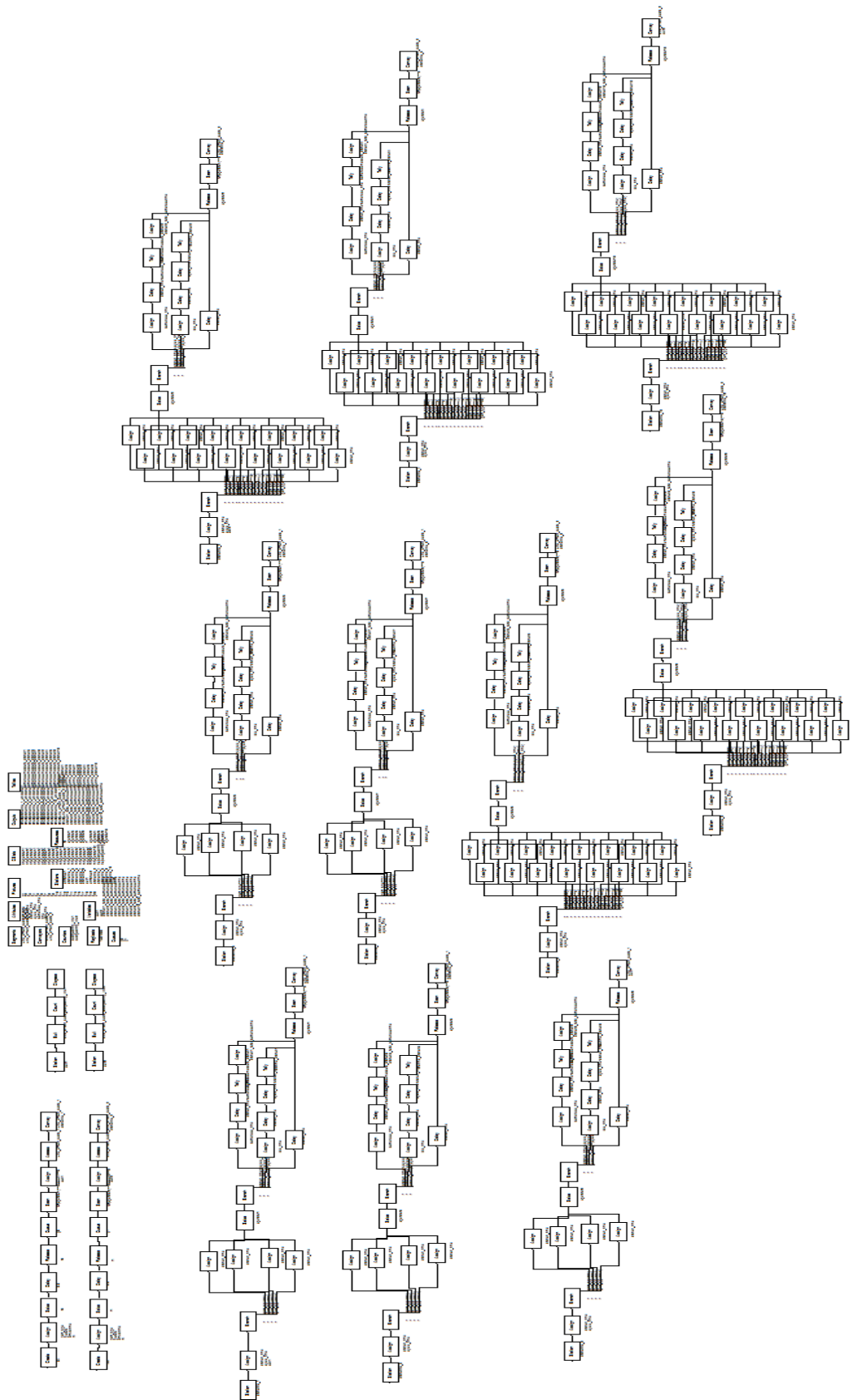


Figure 4.14 The block diagram of proposed model.

It is mentioned in “Separating into 2 parallel lines” part, in which assembly line 20 models produced in the company will run. Entrance block diagrams of line 1 and line 2 are symmetric to each other in the proposed system. The frequency of entities which is entering to the line is determined as 94 sec for line 1 and 102 sec for line 2 in the ‘Create’ module. Production distributions of models assigned to lines are processed to each line with discrete distribution with ‘Assign’ modules. The station logic in proposed situation and the exit block diagram do not differ from the current situation.

The model runs according to cycle times calculated for each line (Line1 94 sec-Line2 102 sec) and results are obtained. Production amount obtained here is close to the expected production amount and production amount of the current situation at the same time. When the proposed model runs in ARENA program, the summary of results obtained can be seen at tables 4.39, 4.40, 4.41, 4.42 and 4.43.

Table 4.39 Total number of production of planning period in ARENA (proposed model)

Number of Production Line 1	16068
Number of Production Line 2	14866
Total Number of Production Proposed Model	30934

Table 4.40 Station bottleneck times of line 1 in ARENA simulation software (proposed model)

Cycle Time==94 sec. (Line 1)	Average Value	Observation	Total time
bottleneckmeasure_station1	0	0	0
bottleneckmeasure_station3	5,027	10304	51798,208
bottleneckmeasure_station5	0	0	0
bottleneckmeasure_station7	15,357	8703	133651,971
bottleneckmeasure_station9	13,1	67	877,7
TOTAL BOTTLENECK TIME OF ALL STATION IN LINE 1			186327,879

Table 4.41 Station idle times of line 1 in ARENA simulation software (proposed model)

Cycle Time==94 sec. (Line 1)	Average Value	Observation	Total time
idletime_station1	1,2805	13403	17162,5415
idletime_station3	9,4298	5766	54372,2268
idletime_station5	23,364	16068	375412,752
idletime_station7	7,7768	6167	47959,5256
idletime_station9	27,084	14801	400870,284
TOTAL IDLE TIME OF ALL STATION IN LINE 1			895777,3299

Table 4.42 Station bottleneck times of line 2 in ARENA simulation software (proposed model)

Cycle Time==102 sec. (Line 2)	Average Value	Observation	Total time
bottleneckmeasure_station2	3,1073	10305	32020,7265
bottleneckmeasure_station4	0	0	0
bottleneckmeasure_station6	0	0	0
bottleneckmeasure_station8	0,76408	12422	9491,40176
bottleneckmeasure_station10	0	0	0
TOTAL BOTTLENECK TIME OF ALL STATION IN LINE 2			41512,12826

Table 4.43 Station idle times of line 2 in ARENA simulation software (proposed model)

Cycle Time==102 sec. (Line 2)	Average Value	Observation	Total time
idletime_station2	4,8074	5766	27719,4684
idletime_station4	48,976	16069	786995,344
idletime_station6	4,4214	14871	65750,6394
idletime_station8	7,4303	2447	18181,9441
idletime_station10	31,668	14867	470808,156
TOTAL IDLE TIME OF ALL STATION IN LINE 2			1369455,552

It is mentioned in “Initial solution with positional weight method” part that, the line efficiency of Line 1 is %82.91 with 94 sec cycle time, and the line efficiency of Line 2 is %88.46 with 102 sec cycle time.

4.2.4.3 Comparison of the Current and Proposed Situation

When the current situation is compared with the proposed situation, it can be seen that new situation data provides better results in terms of production amount and line efficiency. In addition, in the current situation, idle times can be increase just

because the assembly line is not moved in the determined cycle time. The data obtained is summarized in table 4.44.

Table 4.44 Summary of 3 situation evaluation

COMPARISON OF EACH SITUATION	Total Production	Total Bottleneck Time	Total Idle Time	Number of Workers
Current Situation (Cycle time=49sec)	30568	441301,990	5605077,011	12
Current Situation (Cycle time=59.95sec)	24448	96524,167	7442140,403	12
Proposed Situation	30934	227840,007	2265232,882	10

As can be seen from the table above, the maximum production amount can be obtained with the proposed situation. The minimum total bottleneck time can be achieved with the current situation that has 59.95 seconds cycle time. But in this case, both the amount of customer demand cannot be met and workstations with high idle times occur. As can be seen from the table, the maximum total idle time belongs to the current situation that has 59.95 seconds cycle time. In addition, in the current situation, there are 12 employees along the line. In the proposed situation, there are 5 employees of each and 10 in total at both of the lines. In other words, the best result from three results evaluated, is obtained from the proposed situation.

CHAPTER FIVE

CONCLUSION

In this study, mixed model line balancing subject is discussed. The application is run in a company working in the automotive industry that accepted Just in Time systematic. For this reason, the production is completely carried out as customer oriented. In the assembly line added to the application, 20 models are produced depending on customer demand.

In mixed model assembly line balancing, heuristic methods are used as well as analytical methods. Heuristic methods do not ensure the best solution but provide solutions valid and close to the best with a less calculation.

Firstly, the current situation is evaluated in the study. 10 time studies of each model are performed in the evaluation. With the average of 10 observation time obtained belong to the model, task times belong to the model are determined as constant. According to the production amount in the determined planning period, the cycle time to be used in the assembly line is determined for the current situation. According to the determined task times and the cycle time, there is a bottleneck situation in the current assignment.

The main objective of the study is minimizing the bottleneck situation while balancing the line. A heuristic method called neighborhood search in order to overcome the bottleneck situation is used under study. New software is obtained by integrating the algorithm of this heuristic method to MATLAB software program.

In order to overcome the bottleneck situation in the current situation, a new assignment is presented. According to the assignment presented, the U-shaped assembly line running with straight line logic is divided into two parallel lines. 20 models produced in this assembly line are divided in order to be assigned to these two lines by being classified according to production amounts and the duration of production.

The initial solution is created with ranked positional weight method (RPWM) in order to be used in neighborhood search method. This method is applied separately for task times of models separated in both of two lines. Results obtained are integrated in MATLAB software program and intended to provide the software obtain better assignment. But, a more efficient assignment could not be identified than results obtained by RPWM method.

U-shaped assembly line in the current situation and parallel assembly lines with most efficient assignment are modeled in ARENA simulation software. Total production amount, bottleneck times at stations and idle times in the current and proposed situation are determined with this simulation study. When results obtained from this study are evaluated, the improvement in the proposed situation is clearly visible in compliance with the current situation.

Since the new situation obtained in the study gives better results in compliance with the old situation, it can be said that the study is successful. But, the missing aspect of the study is that, a better result could not be obtained from the heuristic method applied. A new heuristic method can be performed in order to get better results than the new situation, as a future research.

REFERENCES

- Ajenblit, D.A., & Wainwright, R.L. (1998). Applying genetic algorithms to the U-shaped assembly line balancing problem. *Institute of Electrical and Electronics Engineers, 9*, 96-101.
- Alp, A., Çerçioğlu, H., Tokaylı, M.A., & Dengiz, B. (2001). Stokastik montaj hattı dengeleme: Bir tavlama benzetimi algoritması. *Endüstri Mühendisliği Dergisi, 12* (3-4), 32-51. Retrieved April 21, 2010, from EBSCO database.
- Amen, M. (2000). Heuristic methods for cost-oriented assembly line balancing: A survey. *International Journal of Production Economics, 68* (1), 1–14.
- Amen, M. (2001). Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time. *International Journal of Production Economics, 69* (3), 255–264.
- Askin, R.G., & Zhou, M. (1997). A parallel station heuristic for the mixed-model production line balancing problem. *International Journal of Production Research, 35* (11), 3095-3105.
- Baykasoğlu, A. & Dereli, T. (2009). Simple and U-type assembly line balancing by using an ant colony based algorithm. *Mathematical and Computational Applications, 14* (1), 1-12.
- Bryton, B. (1954). *Balancing of a continuous production line*. M.S. Thesis, Northwestern University, Evanston, IL.
- Bukchin, J., Dar-El, E.M., & Rubinovitz, J. (2002). Mixed model assembly line design in a make-to-order environment. *Computers & Industrial Engineering, 41* (4), 405-421.

- Cao, Z., & Ma, S. (2008). Balancing and sequencing optimization of the mixed model assembly lines. *International Symposium on Computer Science and Computational Technology, itesi Mühendislik Mimarlık Fakültesi Dergisi*, 08 (5), 732-736.
- Çakır, B. (2006). *Stokastik işlem zamanlı montaj hattı dengeleme için tavlama benzetimi algoritması*. Gazi Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, Ankara.
- Erel, E., & Gökçen, E. (1999). Shortest-route formulation of mixed-model assembly line balancing problem. *European Journal of Scientific Research*, 116 (1), 194-204.
- Erel, E., Sabuncuoğlu, I., & Aksu, A. (2001). Balancing of U-type assembly systems using simulated annealing. *International Journal of Production Research*, 39 (13), 3003-3015.
- Erel, E., & Sarin, S.C. (1998). A survey of the assembly line balancing procedures. *Production Planning and Control*, 9 (5), 414-434.
- Erkut, H., & Baskak, M. (1997). *Stratejiden uygulamaya, Tesis Tasarımı*. (2th ed.) İrfan Yayımcılık, İstanbul.
- Esmailian, G.R., Ismail, N., Sulaiman, S., Ahmad, M.M.H.M., & Hamedi, M. (2009). Allocating and balancing of mixed-model production through the parallel assembly lines. *European Journal of Scientific Research*, 31 (4), 616–631.
- Ghosh, S., & Gagnon, J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 27 (4), 637-670.

- Gökçen, H., & Ağpak, K. (2004). Hat dengelemede yeni bir felsefe paralel montaj hatlarının eş zamanlı dengelenmesi. *Teknoloji*, 7 (1), 181-188.
- Gökçen, H., & Ağpak, K. (2006). A goal programming approach to simple U-line balancing problem. *European Journal of Operational Research*, 171 (2), 577–585.
- Gökçen, H., & Erel, E. (1997). A goal programming approach to mixed-model assembly line balancing problem. *International Journal of Production Research*, 48 (2), 177-185.
- Jin, M., & Wu, S.D. (2002). A new heuristic method for mixed model assembly line balancing problem. *Computers & Industrial Engineering*, 44 (1), 159-169.
- Kahraman, F., & İspir, A. (2004). Montaj hattı dengeleme algoritmalarının karşılaştırılması. *Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 19 (1), 195-203.
- Keskintürk, T., & Küçük, B. (2006). Karışık modellenmiş montaj hatlarının genetik algoritma kullanılarak dengelenmesi. *Yönetim*, 17 (53), 52-63.
- Lee, O.L., Kim, Y., & Kim, Y.K. (2001). Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers & Industrial Engineering*, 40 (3), 273-292.
- Matanachai, S., & Yano, C.A. (2001). Balancing mixed-model assembly lines to reduce work overload. *Institute of Industrial Engineers Transactions*, 33 (1), 29-42.
- Mcmullen, P.R., & Frazier, G.V. (1997). A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics*, 51 (3), 177-190.

- Mcmullen, P.R., & Frazier, G.V. (2008). Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstation. *International Journal of Production Research*, 36 (10), 2717-2741.
- Merengo, C., Nava, F., & Pozzetti, A. (1999). Balancing and sequencing manual mixed-model assembly lines. *International Journal of Production Research*, 37 (12), 2835-2860.
- Miltenburg, G.J., & Wijngaard, J. (1994). The U-line line balancing problem. *Management Science*, 40 (10), 1378-1388.
- Orbak, A.Y., Özalp, B.T., Korkmaz, P., Yarkın, N., Aktaş, N., & Dinçer, A. (2009). Karışık modellenli bir montaj hattında hat dengeleme çalışmaları. *Yöneylem Araştırması ve Endüstri Mühendisliği 29. Ulusal Kongresi*.
- Özkıran, A., & Düşünür, H. (2011). *Otomotiv sektöründeki bir işletmede montaj hattının analizi ve dengeleme çalışması*. Dokuz Eylül Üniversitesi Endüstri Mühendisliği Tezi, Ankara.
- Sabuncuoğlu, I., Erel, E., & Alp, A. (2009). Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal of Production Economics*, 120 (2), 287-300.
- Salveson, M.E. (1955). The assembly line balancing problem. *Journal of Industrial Engineering*, 6 (3), 18-25.
- Scholl, A. (1999). *In balancing and sequencing of assembly lines*. (2nd ed.) New York: Physica-Verlag.
- Simülasyon ve modelleme nedir?* (n.d.). Retrieved February 27, 2011, from <http://www.donusumkonagi.net/makale.asp>

- Sparkling, D., & Miltenburg, J. (1998). The mixed-model U-line balancing problem. *International Journal of Production Research*, 36 (2), 485-501.
- Thomopoulos, N.T. (1967). Line balancing-sequencing for mixed-model assembly. *Management Science*, 14 (2), B59-B75.
- Thomopoulos, N.T. (1970). Mixed-model line balancing with smoothed station assignments. *Management Science*, 16 (9), 593-603.
- Tanyaş, M., & Baksak, M. (2003). *Üretim planlama ve kontrol*. İrfan Yayımcılık, İstanbul.
- Uzun, G. (18 August 2008). Montaj hattı dengeleme (Konum ağırlıklı dengeleme metodu). 12 March 2009, <http://gokhanca.com/algorithm/102-montaj-hatti-dengeleme.html>.
- Vilarinho, P.M., & Simaria, A.S. (2002). A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 40 (6), 1405-1420.
- Vilarinho, P.M., & Simaria, A.S. (2006). ANTBAL: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 44 (2), 291-303.
- Waldemar, G. (2009). Assembly line balancing problem single and two-sided structures. *Automation Control, Theory and Practice*, 39, 978-953.
- Wu, E.F., Jin, Y., Bao, J.S. & Hu, F. (2007). A branch-and-bound algorithm for two sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, 39, 1009-1015.

Xu, W., & Xiao, T. (2008). Mixed model assembly line balancing problem with fuzzy operation times and drifting operations. *Institute of Electrical and Electronics Engineers, Proceedings of the 2008 Winter Simulation Conference*.

Yagmahan, B. (2011). Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. *Expert Systems with Applications*, 38 (10), 12453–12461.

Yılmaz, E., & Erol, R. (2005). Montaj hatlarının değişen koşullar altında yeniden dengelenmesi. *Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 20 (1), 213-227.

APPENDICES A

Table 4.3 MCV Aa Precedence Graph

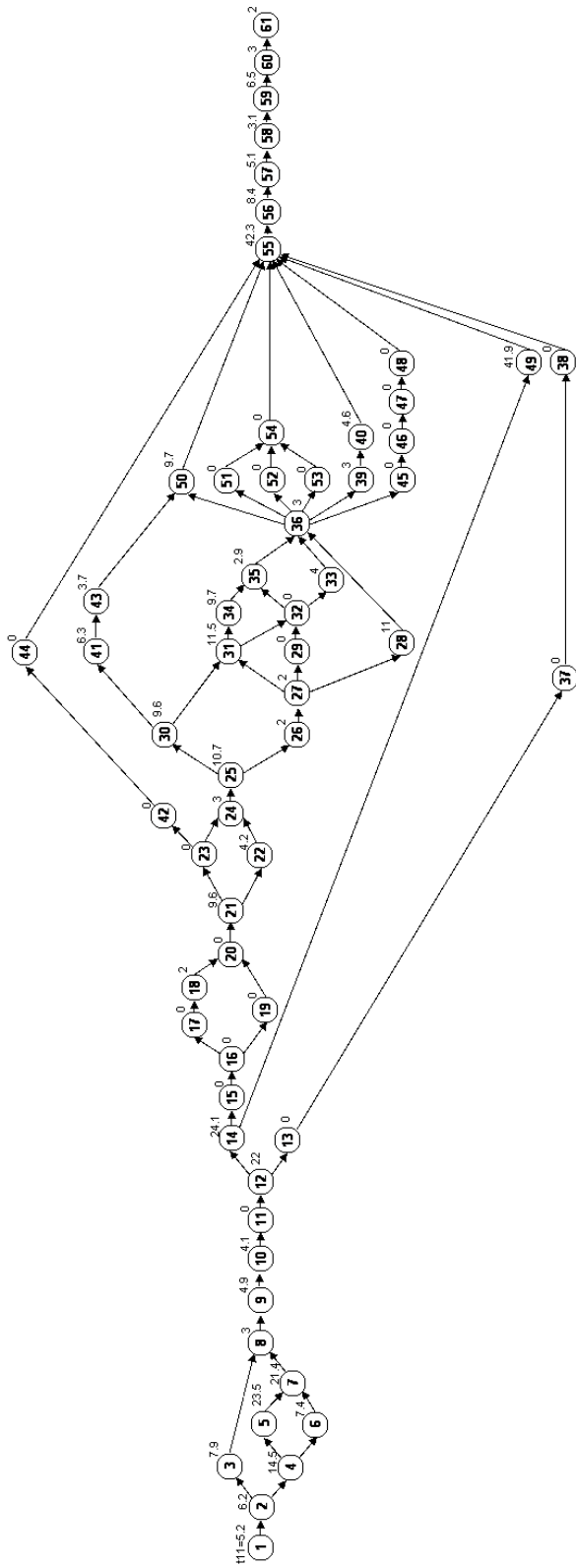


Table 4.4 MCV Ab Precedence Graph

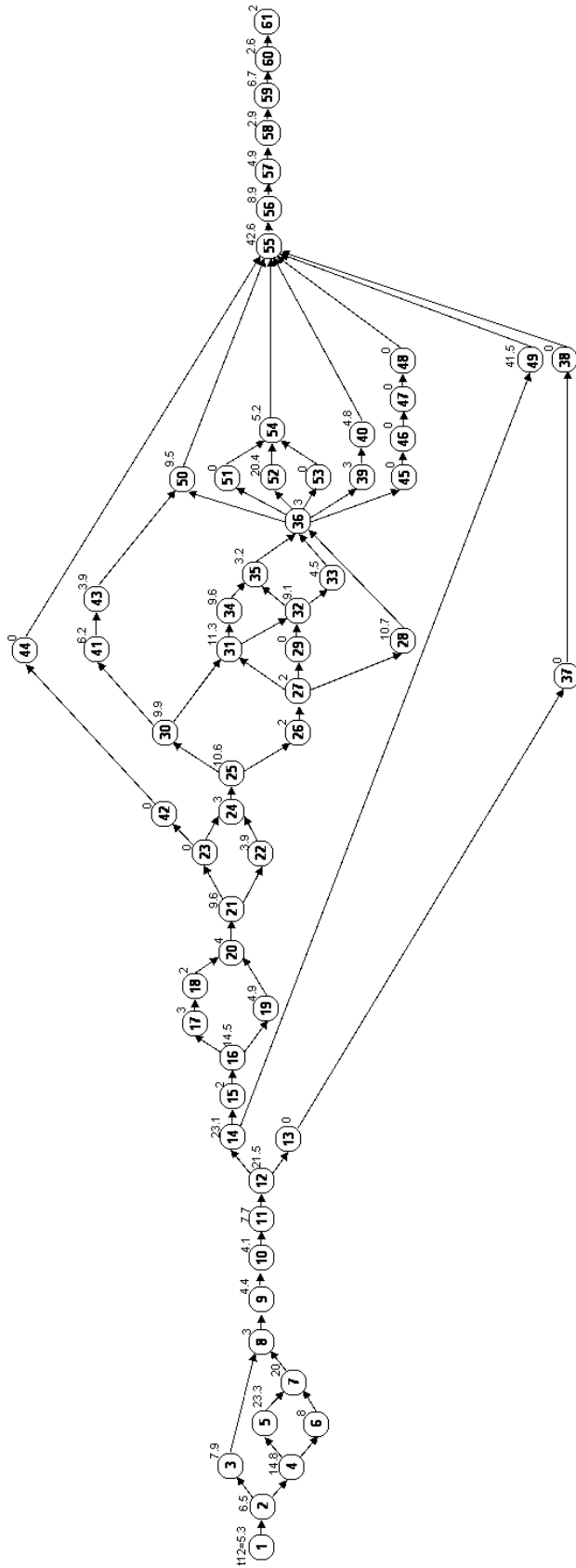


Table 4.5 MCV A1a Precedence Graph

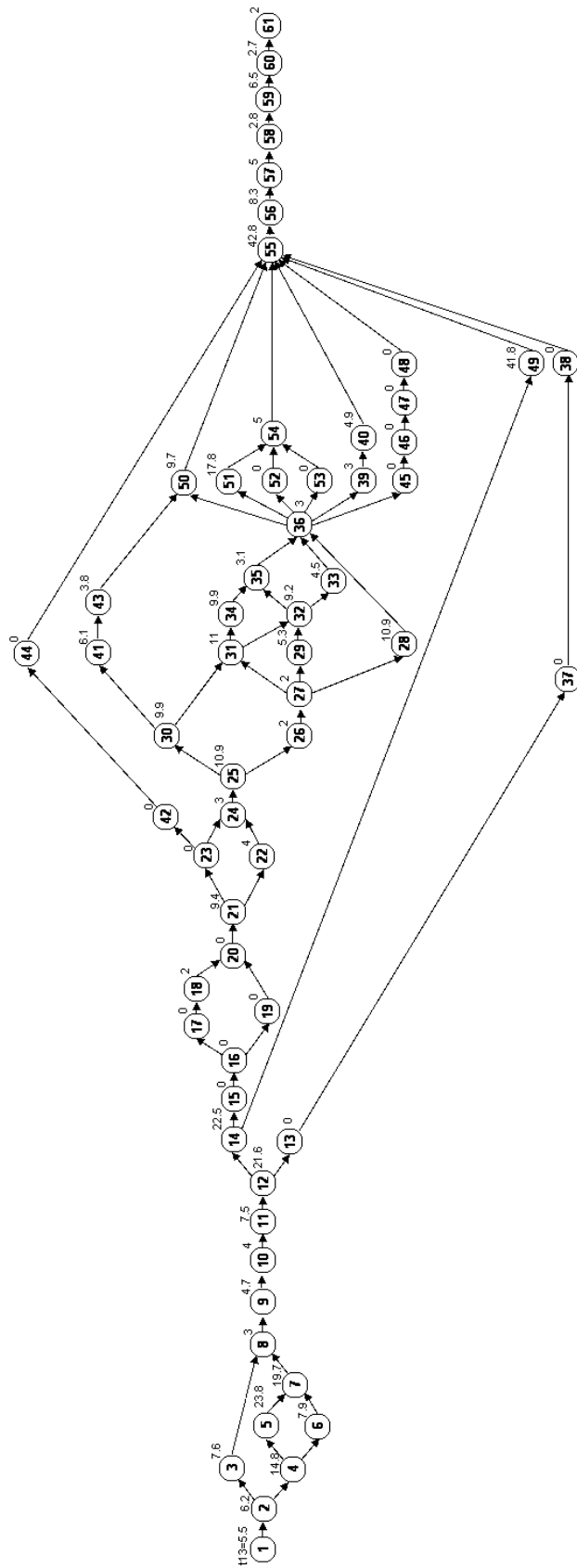


Table 4.6 MCV A1b Precedence Graph

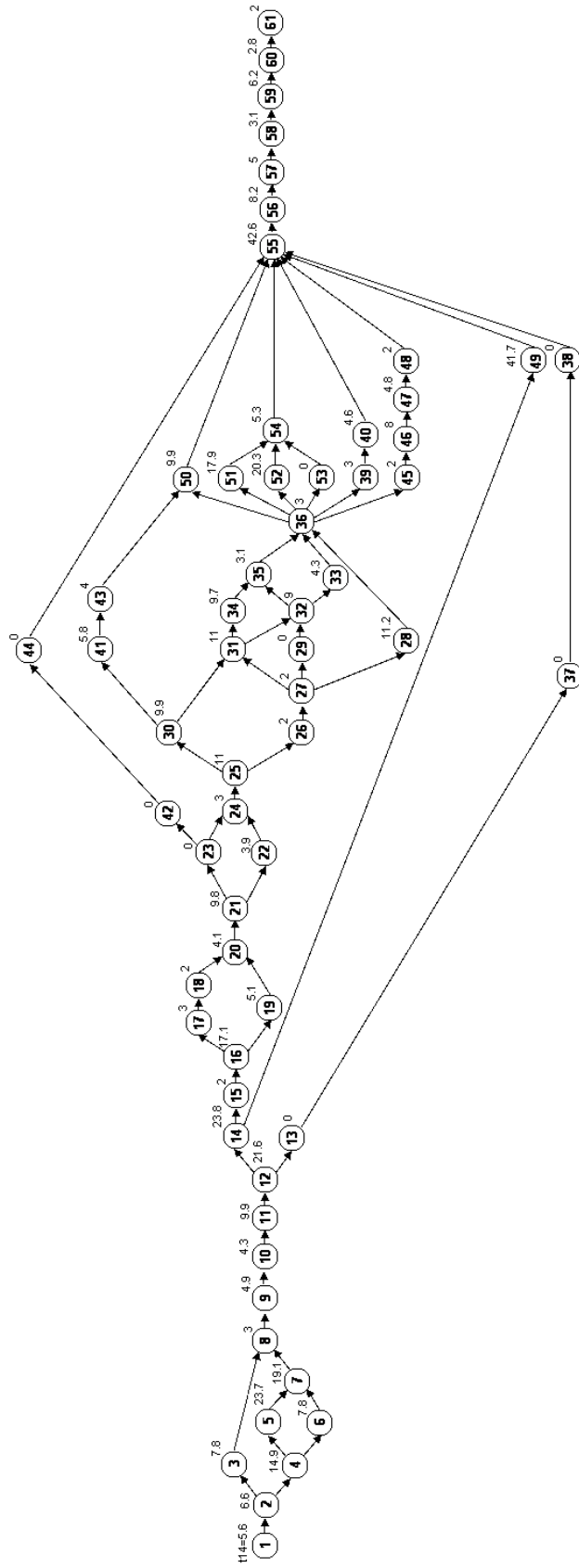


Table 4.7 MCV A2a Precedence Graph

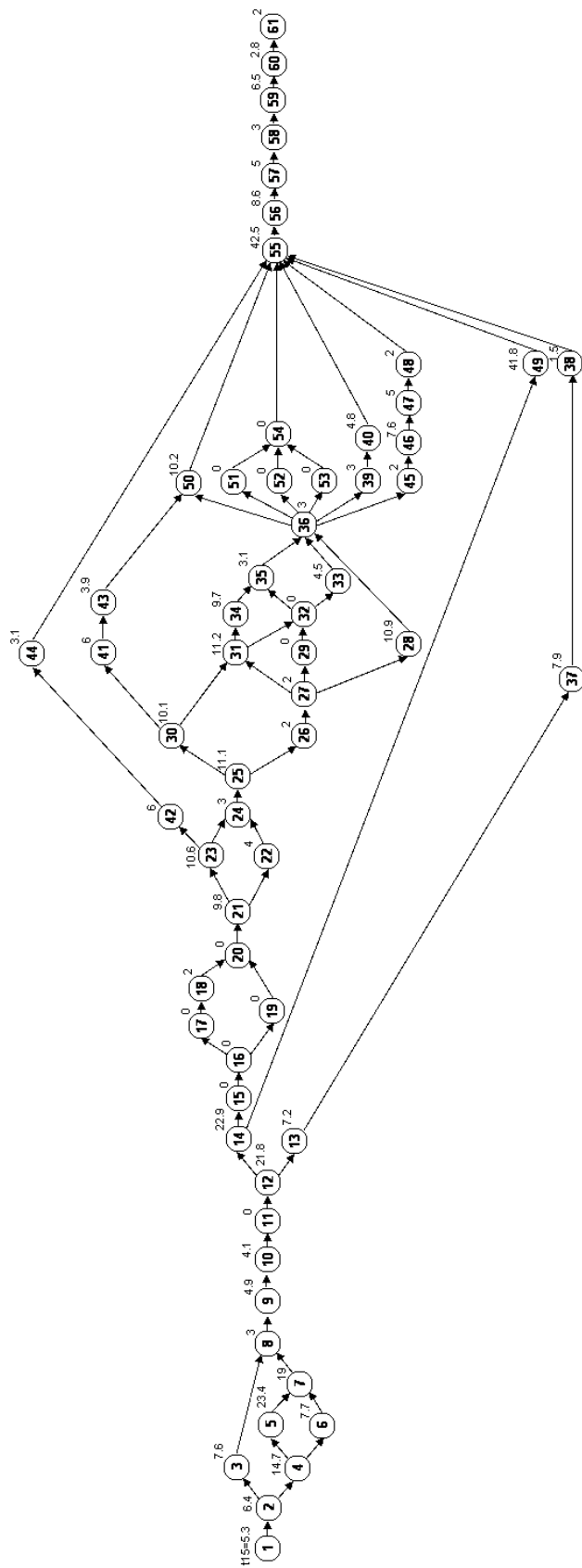


Table 4.8 MCV A2b Precedence Graph

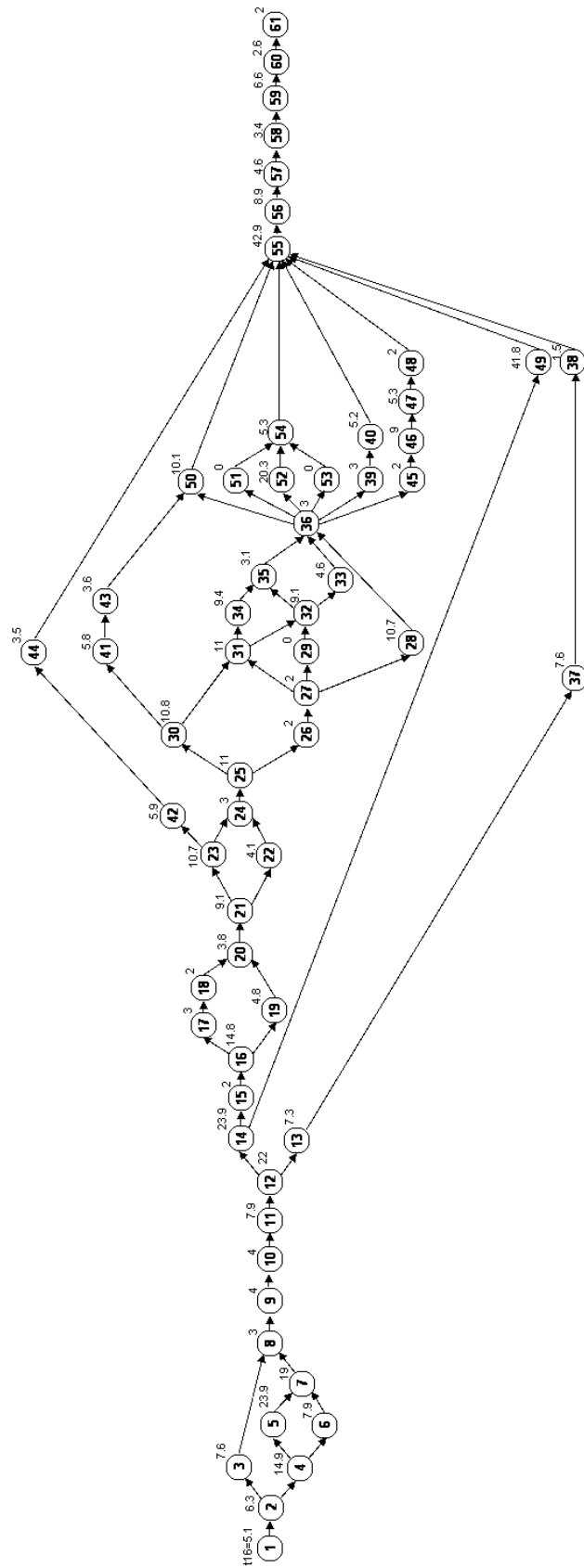


Table 4.9 MCV A2c Precedence Graph

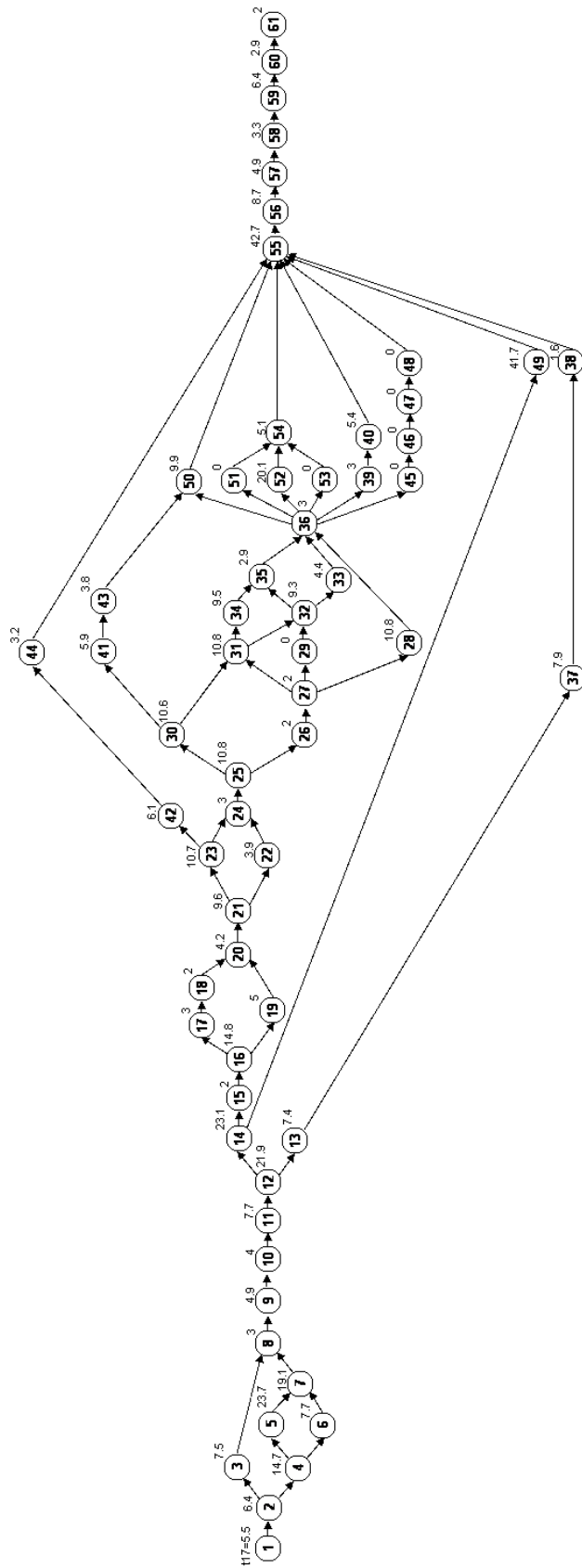


Table 4.10 MCV A3a Precedence Graph

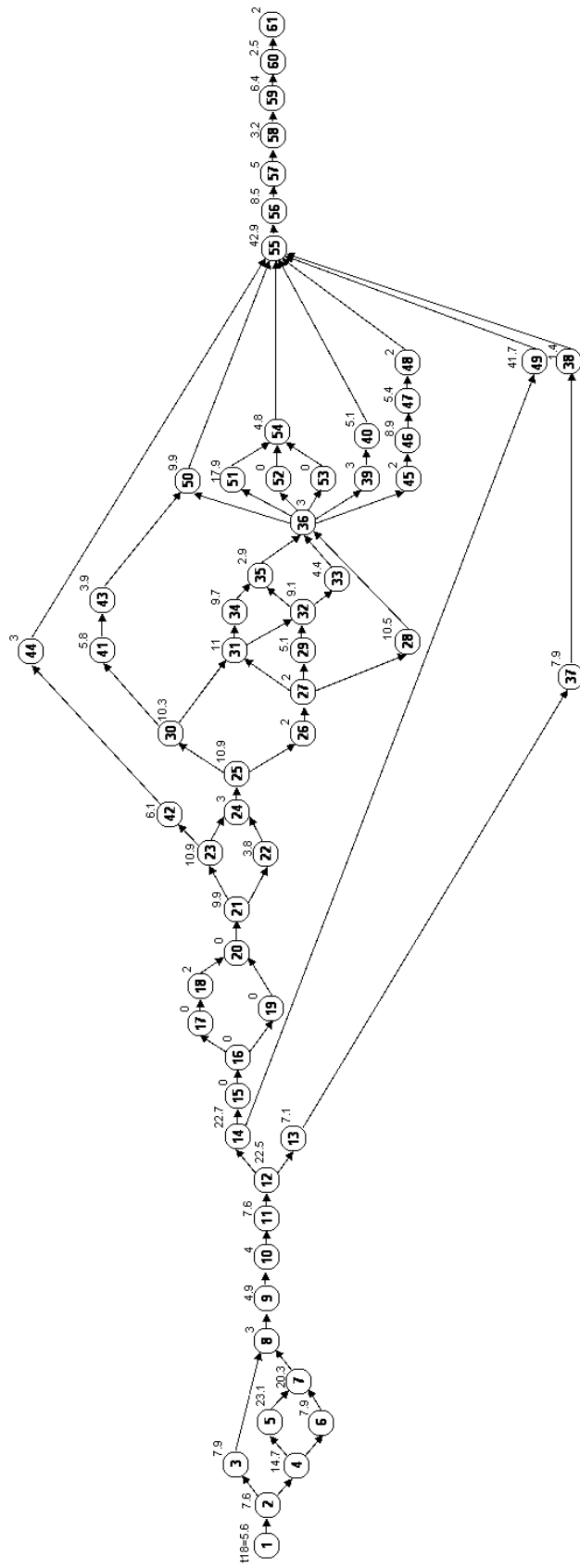


Table 4.11 MCV A3b Precedence Graph

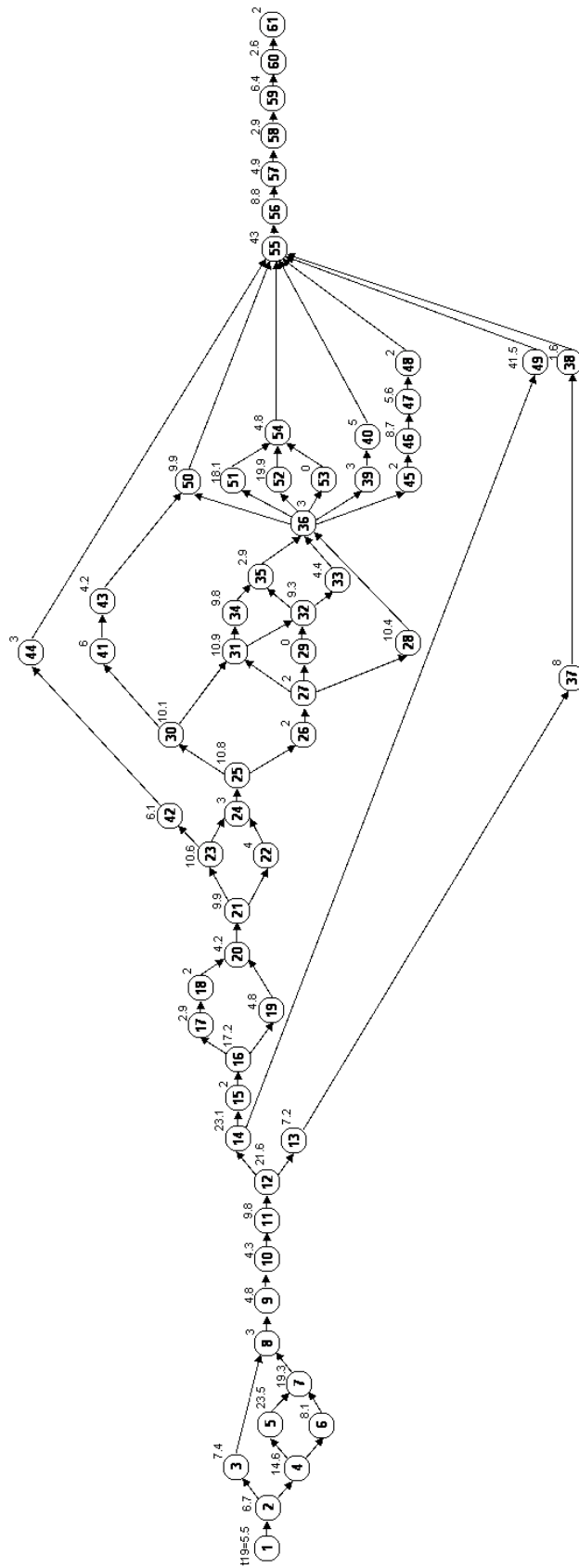


Table 4.12 MCV A3c Precedence Graph

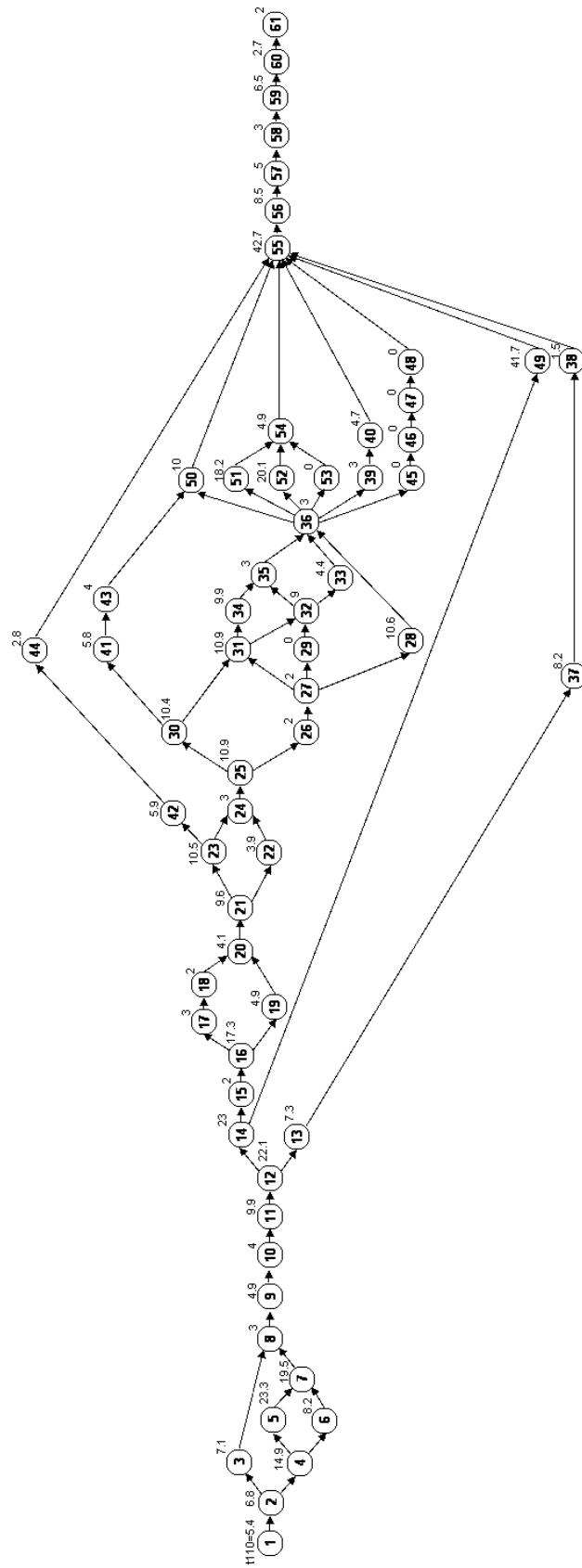


Table 4.13 D200 Aa Precedence Graph

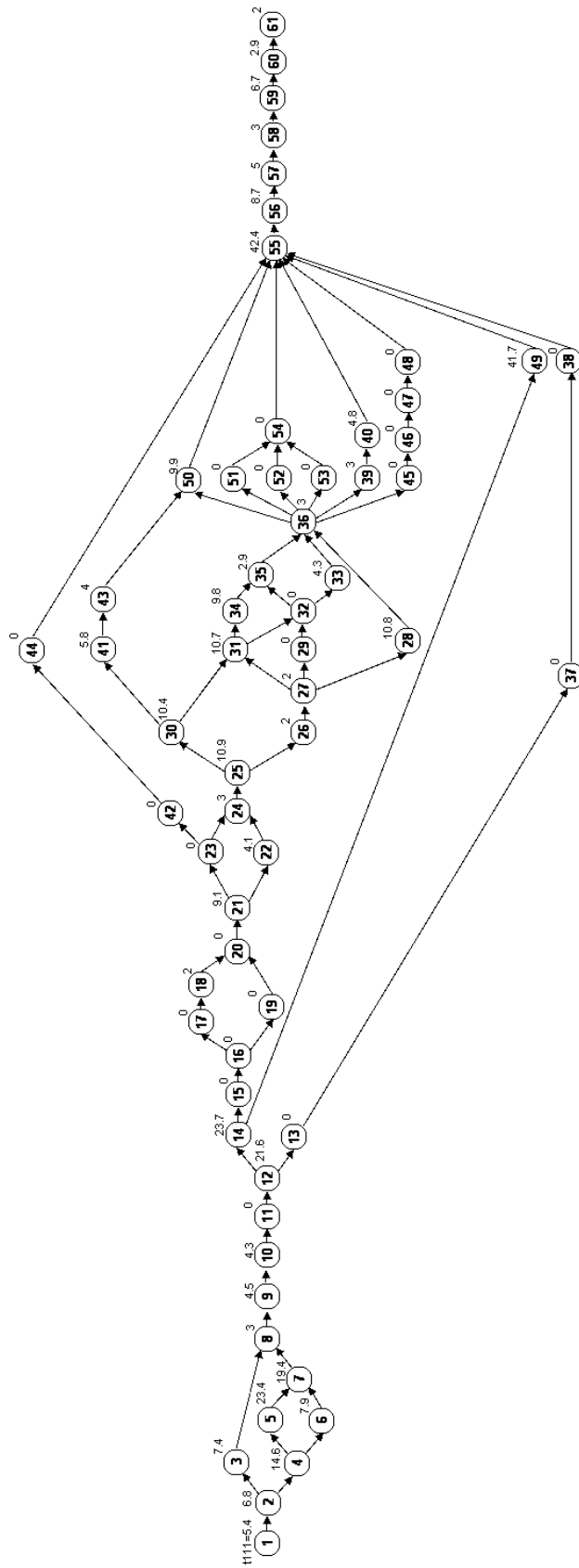


Table 4.14 D200 Ab Precedence Graph

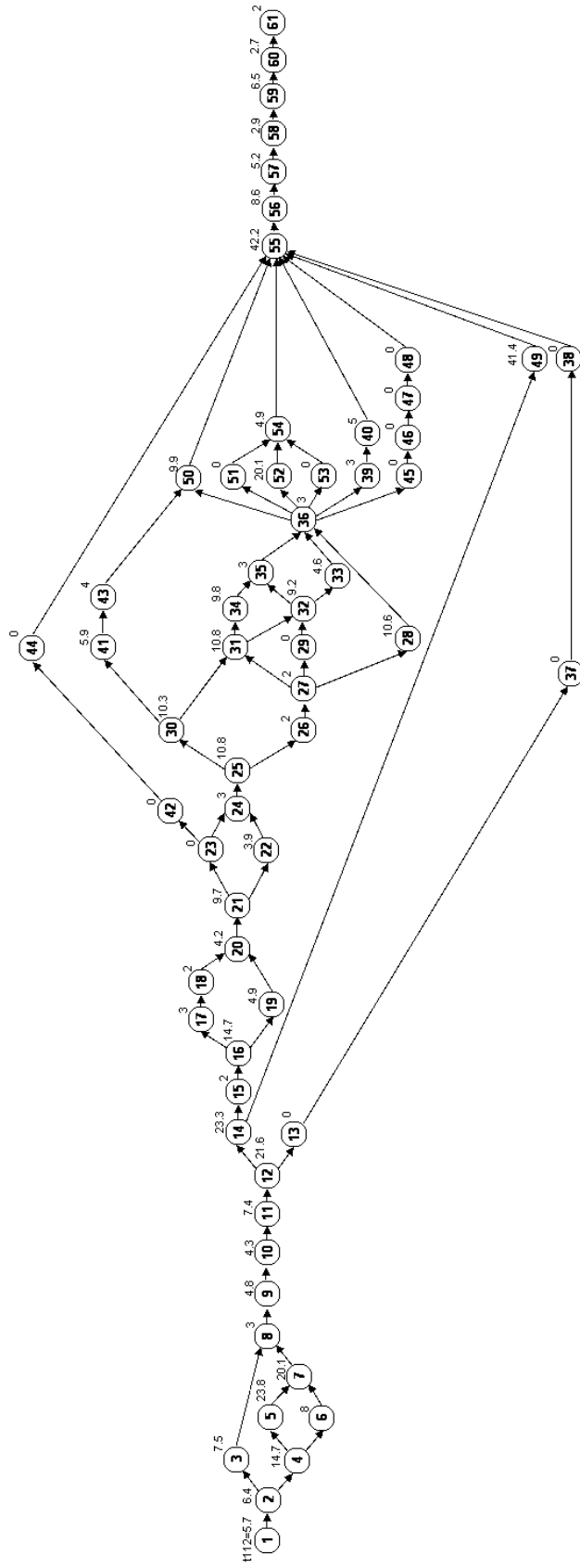


Table 4.15 D200 A1a Precedence Graph

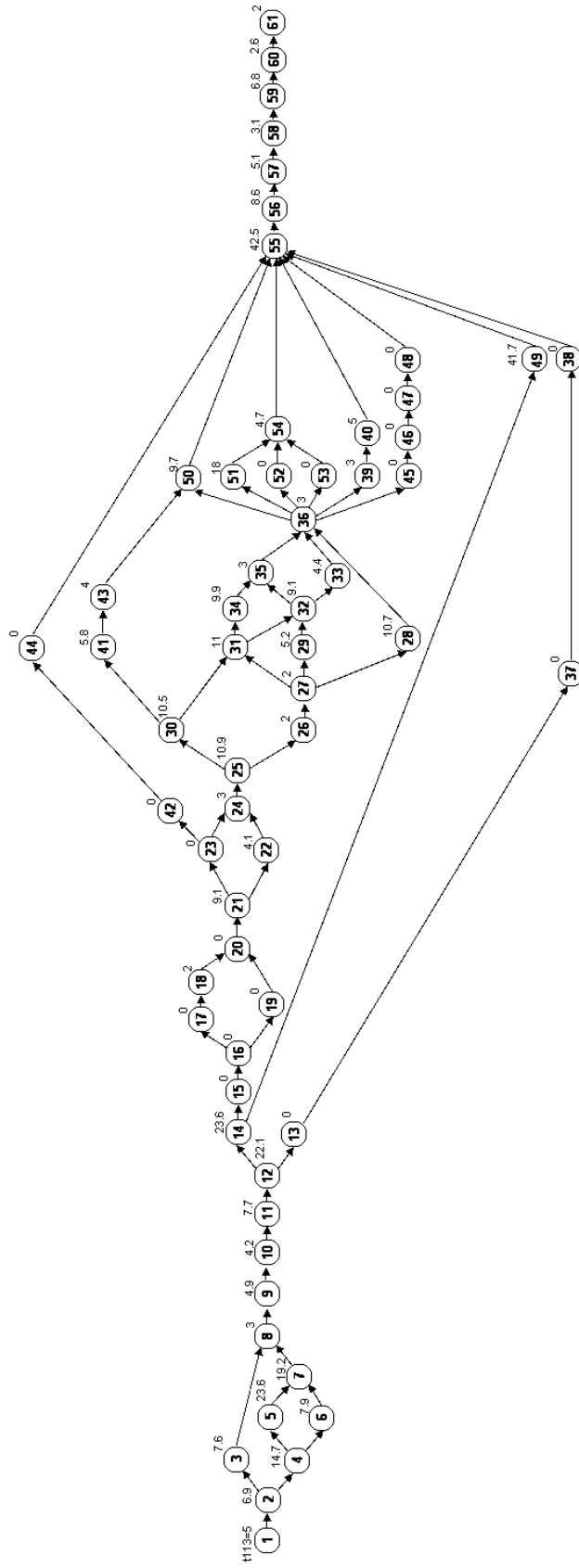


Table 4.16 D200 A1b Precedence Graph

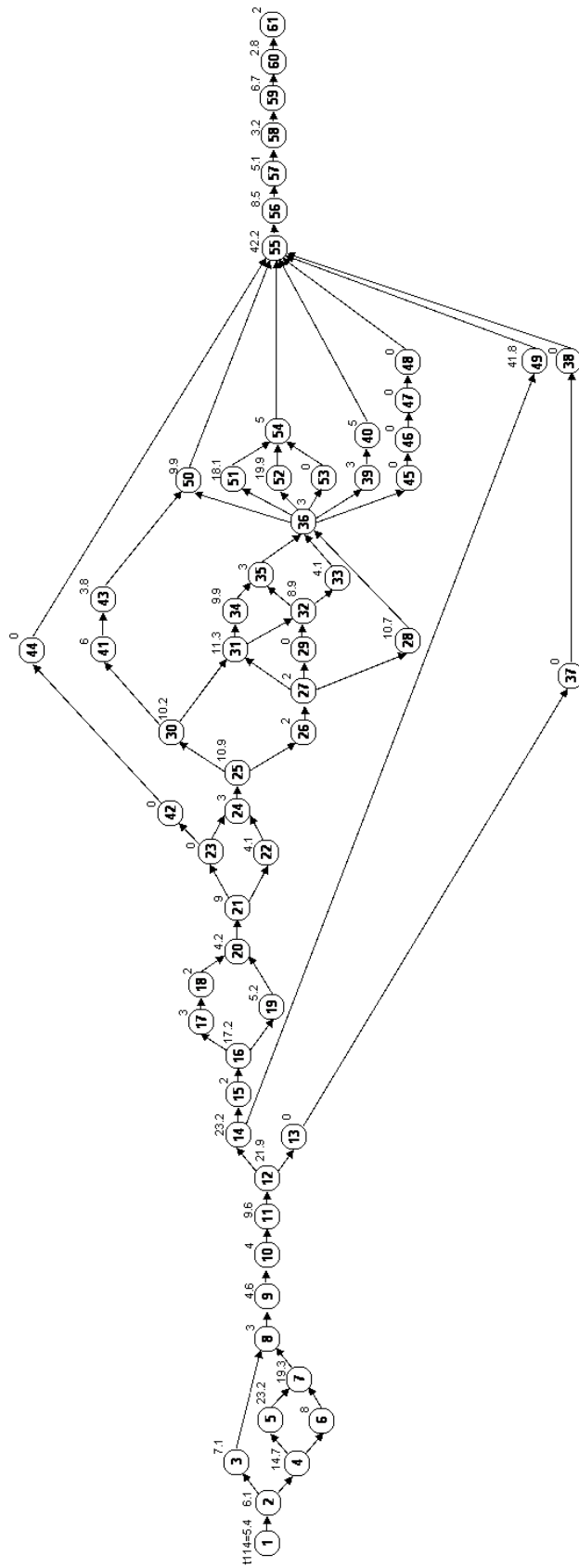


Table 4.17 D200 A2a Precedence Graph

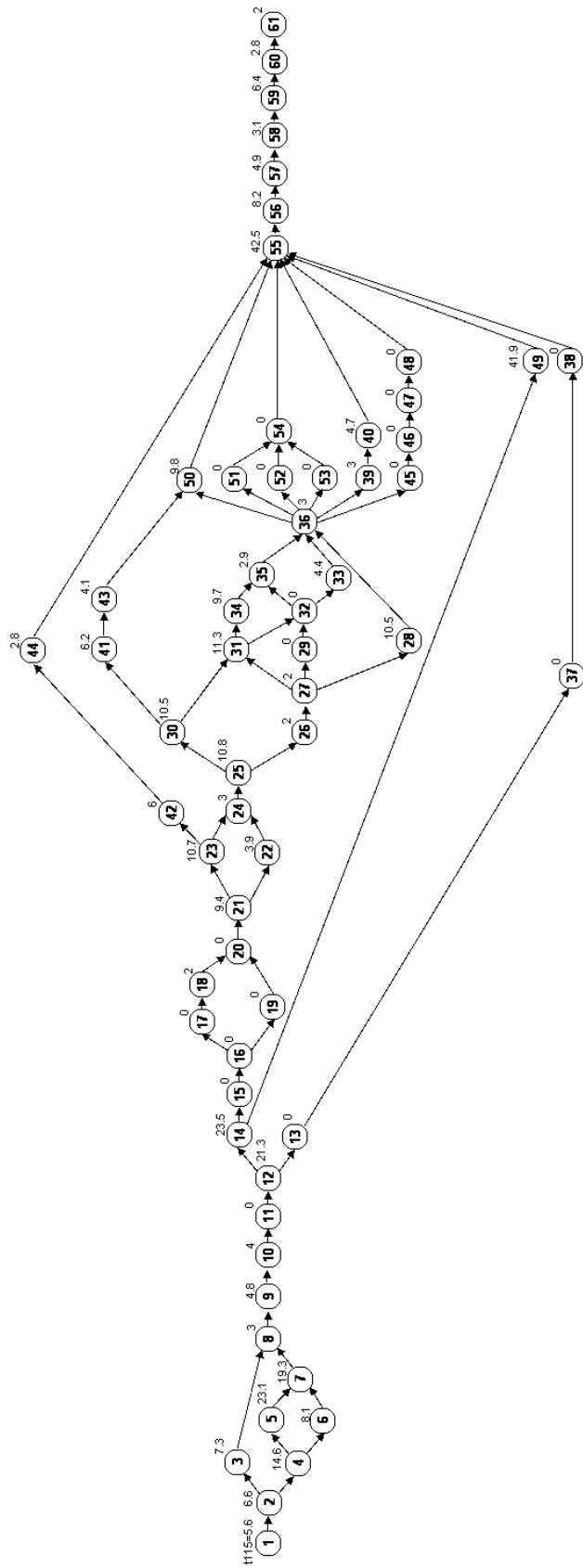


Table 4.18 D200 A2b Precedence Graph

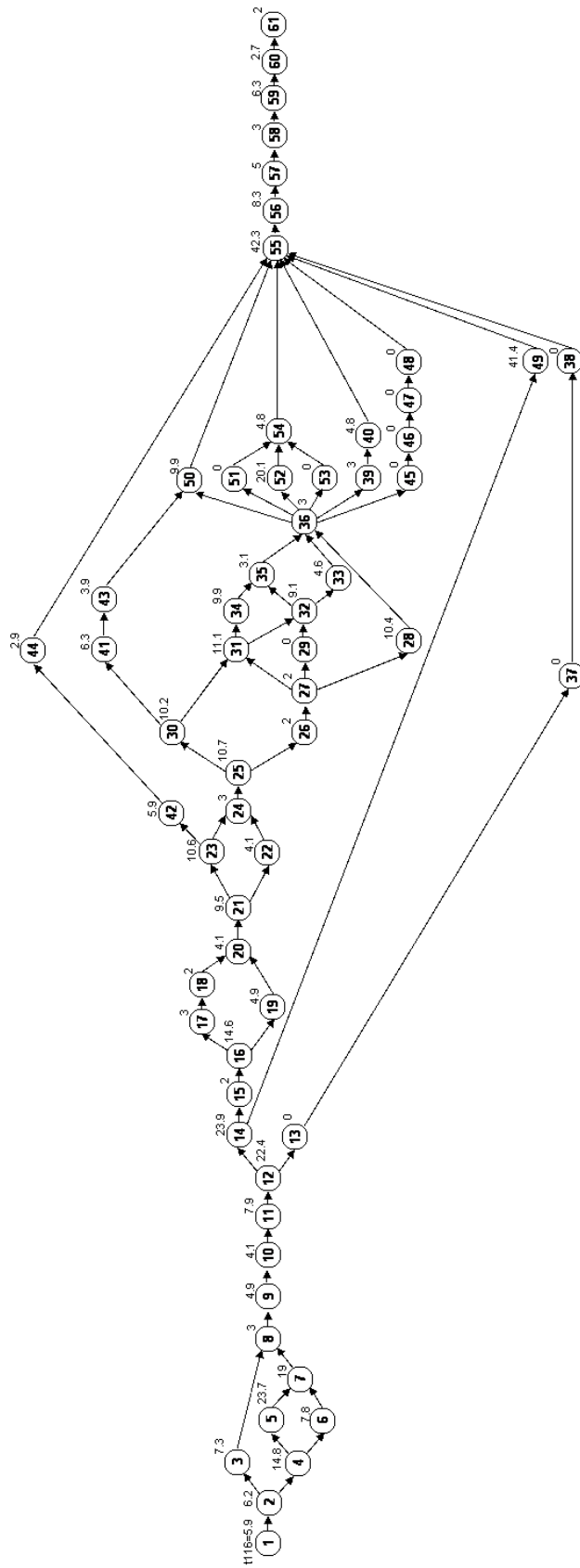


Table 4.19 D200 A3a Precedence Graph

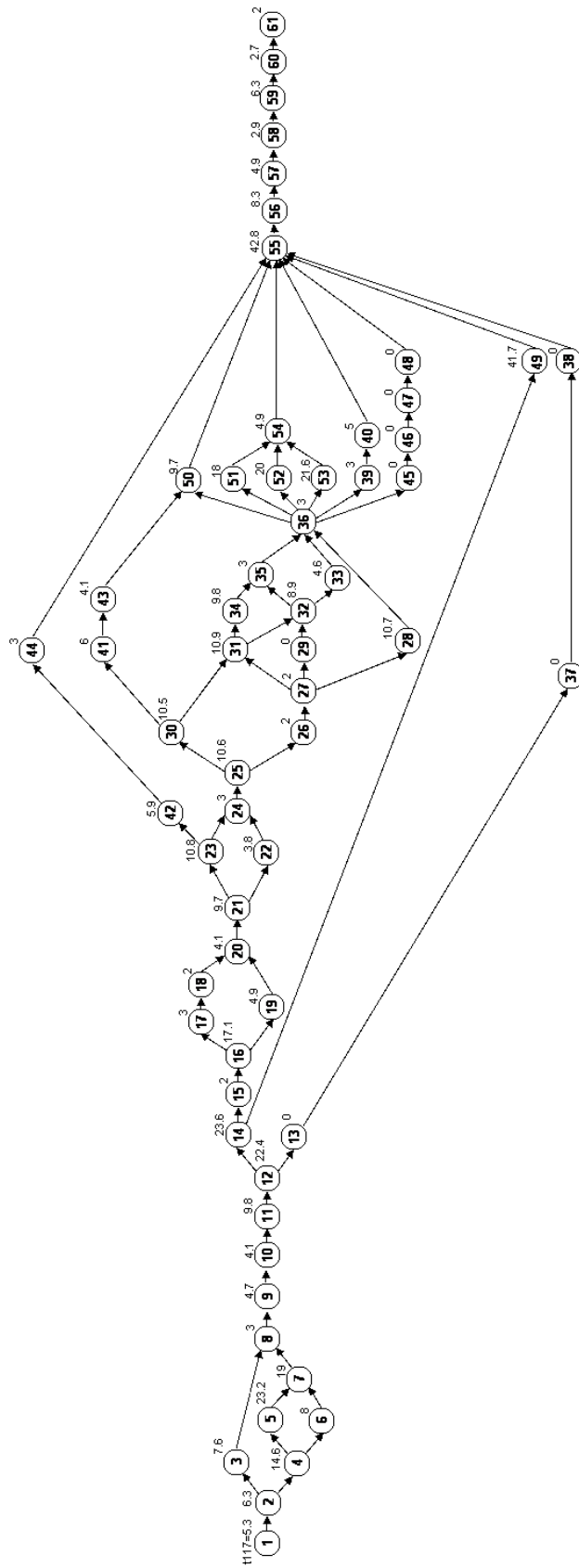


Table 4.20 D200 A3b Precedence Graph

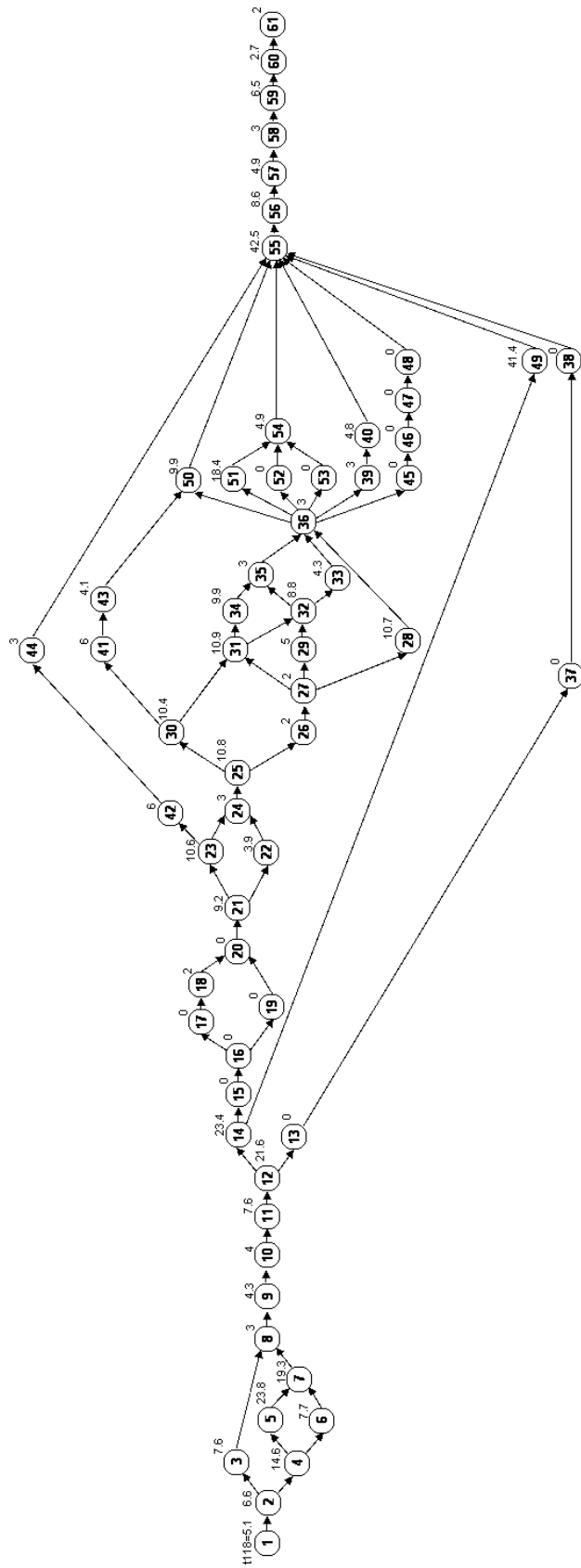


Table 4.21 D200 A3c Precedence Graph

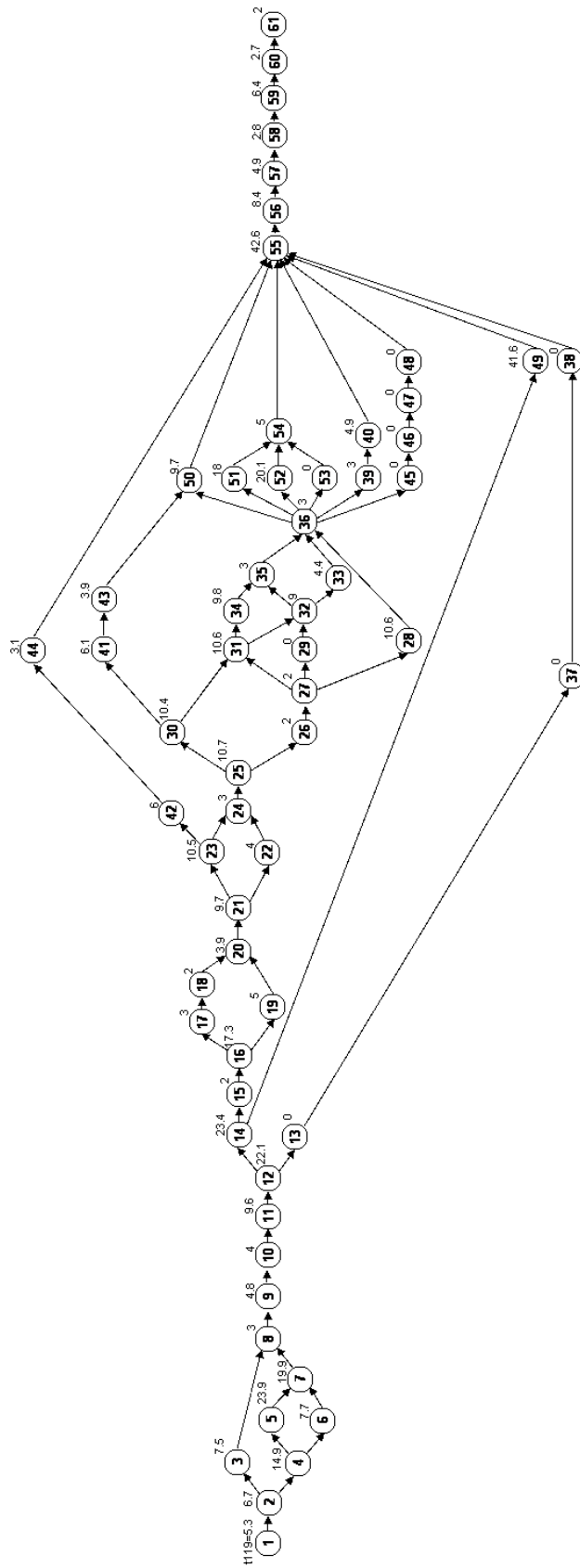


Table 4.22 D200 A3d Precedence Graph

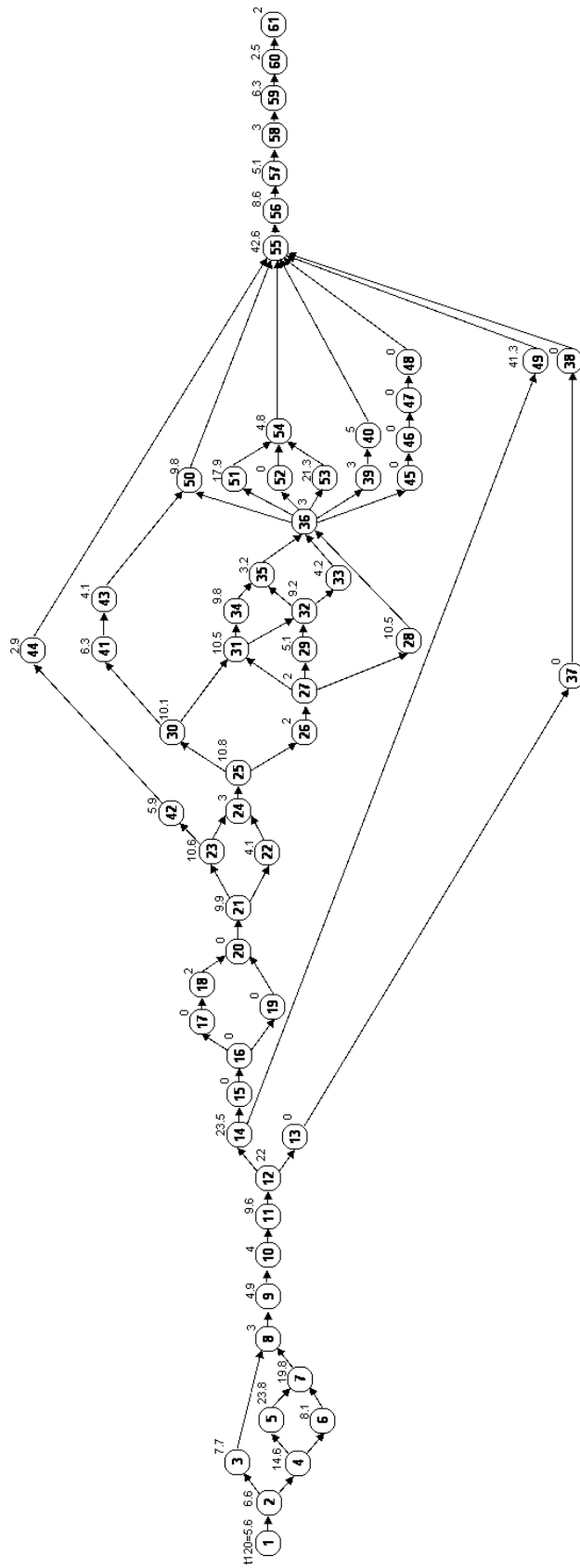


Table 4.23 Average operation times of models

DESCRIPTION OF TASKS	TIME (sec)																											WEIGHTED AVERAGE		
	MCV													D200																
	Aa	Ab	A1a	A1b	A2a	A2b	A2c	A3a	A3b	A3c	A4	A5	A6	A7a	A7b	A8a	A8b	A9a	A9b	A9c	A9d	A9e	A9f	A9g	A9h	A9i	A9j			
1 TAKING SEAT FRAME FROM RACK AND PLACING FRAME TO THE PLATFORM	5.2	5.3	5.5	5.6	5.4	5.1	5.5	5.6	5.5	5.4	5.4	5.7	5	5.4	5.6	5.9	5.3	5.1	5.3	5.6	5.40									
2 RIPPING LABEL	6.2	6.5	6.2	6.6	6.4	6.3	6.4	7.6	6.7	6.8	6.8	6.4	6.9	6.1	6.6	6.2	6.3	6.6	6.7	6.6	6.70									
3 INSERTING SLIDE STOPPER	7.9	7.9	7.6	7.8	7.6	7.6	7.5	7.9	7.4	7.1	7.4	7.5	7.6	7.1	7.3	7.3	7.6	7.6	7.5	7.7	3.05									
4 TURNING SLIPOCOVER THAT IS HOOKED TO FORWARD LEAVING IT ON THE FRAME	14.5	14.8	14.8	14.9	14.7	14.9	14.7	14.7	14.6	14.8	14.6	14.7	14.7	14.7	14.6	14.8	14.6	14.6	14.9	14.6	14.68									
5 DRESSING SEA FOAM TO THE FRAME	23.5	23.3	23.6	23.7	23.4	23.9	23.7	23.1	23.5	23.3	23.4	23.8	23.6	23.2	23.1	23.7	23.2	23.8	23.9	23.8	23.44									
6 DRESSING SEA FOAM TO THE FRAME	7.4	8	7.9	7.8	7.7	7.9	7.7	7.9	8.1	8.2	7.9	8	7.9	8	8.1	7.8	8	7.7	7.7	8.1	7.79									
7 INSERTING DOWELS INTO THE FRAME	21.4	20	19.7	19.1	19	19	19.1	20.3	19.3	19.5	19.4	20.1	19.2	19.3	19.3	19	19	19.3	19.9	19.8	19.90									
8 ROTATING THE FRAME (H9)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.05									
9 PLACEMENT OF DRESSED BACKREST ON THE SEAT	4.9	4.4	4.7	4.9	4.9	4	4.9	4.9	4.8	4.9	4.5	4.8	4.9	4.6	4.8	4.9	4.7	4.3	4.8	4.9	4.60									
10 BACKREST SEAT LABEL VERIFICATION	4.1	4.1	4	4.3	4.1	4	4	4	4.3	4	4.3	4.3	4.2	4	4	4.1	4	4	4	4	4.08									
11 RIPPING CABLES COMING FROM BACKREST THROUGH THE SEAT FRAME	0	7.7	7.5	9.9	0	7.9	7.7	7.6	9.8	9.9	0	7.4	7.7	9.6	0	7.9	9.8	7.6	9.6	9.6	4.34									
12 FIXING SCREWS OF BACKREST BY HAND	22	21.5	21.6	21.6	21.8	22	21.9	22.5	21.6	22.1	21.6	21.6	22.1	21.9	21.3	22.4	22.4	21.6	22.1	22	21.96									
13 INSERTING PORTHOLE BOTTOM COVER AND PORTHOLE COVER	0	0	0	0	7.2	7.3	7.4	7.1	7.2	7.3	0	0	0	0	0	0	0	0	0	0	3.00									
14 TORCHING BACKREST SEAT MERGING SCREWS	24.1	23.1	22.5	23.8	22.9	23.9	23.1	22.7	23.1	23	23.7	23.3	23.6	23.2	23.5	23.9	23.6	23.4	23.4	23.4	23.13									
15 LIFTING PLATFORM	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0.12									
16 RIPPING HEATER CABLE TO PULLMIFLEX WITH DOWELS	0	14.5	0	17.1	0	14.8	14.8	0	17.2	17.3	0	14.7	0	17.2	0	14.6	17.1	0	17.3	0	0.88									
17 LANDING PLATFORM	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0.17									
18 ROTATING PLATFORM (H9)	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2.00									
19 FIXING HEATER CABLE TO THE FRAME WITH DOWELS	0	4.9	0	5.1	0	4.8	5	0	4.8	4.9	0	4.9	0	5.2	0	4.9	4.9	0	5	0	0.28									
20 INSERTING HEATER CLIP TO THE PLASTIC CASING	0	4	0	4.1	0	3.8	4.2	0	4.2	4.1	0	4.2	0	4.2	0	4.1	4.1	0	3.9	0	0.23									
21 INSERTING OUTER PLASTIC CASING TO THE FRAME	9.6	9.6	9.4	9.8	9.8	9.1	9.6	9.9	9.9	9.6	9.1	9.7	9.1	9	9.4	9.5	9.7	9.2	9.7	9.9	9.59									
22 INSERTING KEYPER PLASTIC	4.2	3.9	4	3.9	4	4.1	3.9	3.8	4	3.9	4.1	3.9	4.1	4.1	3.9	4.1	3.8	3.9	4	4.1	4.00									
23 FIXING THE SCREW OF TILT ADJUSTMENT LEVER BY HAND	0	0	0	0	10.6	10.7	10.7	10.9	10.6	10.5	0	0	0	0	10.7	10.6	10.8	10.6	10.5	10.6	4.65									
24 ROTATING PLATFORM (H9)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.00									
25 INSERTING INNER PLASTIC CASING TO THE FRAME	10.7	10.6	10.9	11	11.1	11	10.8	10.9	10.8	10.9	10.9	10.8	10.9	10.9	10.8	10.7	10.6	10.8	10.7	10.8	10.89									
26 ROTATING PLATFORM (H9)	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2.00									
27 LIFTING PLATFORM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.12									
28 SCREWING BACKREST SCREWS OF INNER AND OUTER CASING	11	10.7	10.9	11.2	10.9	10.7	10.8	10.5	10.4	10.6	10.8	10.6	10.7	10.7	10.5	10.4	10.7	10.7	10.6	10.5	10.79									
29 FIXING SIDEWALL CABLE TO PULLMIFLEX WITH DOWELS	0	0	0	5.3	0	0	0	5.1	0	0	0	0	0	5.2	0	0	0	5	0	5.1	2.65									
30 FIXING THE SCREW OF SAFETY BELT BUCKLE BY HAND	9.6	9.9	9.9	9.9	10.1	10.8	10.6	10.3	10.1	10.4	10.4	10.3	10.5	10.2	10.5	10.2	10.5	10.4	10.4	10.1	10.08									
31 FIXING THE SAFETY BELT CABLE TO PULLMIFLEX BY DOWELS	11.5	11.3	11	11	11.2	11	10.8	11	10.9	10.9	10.7	10.8	11	11.3	11.1	10.9	10.9	10.6	10.5	10.5	11.10									
32 BRINGING WITH ZIP TIE	0	9.1	9.2	9	0	9.1	9.3	9	0	9.2	9.1	8.9	9	9.1	8.9	8.8	9	8.9	8.8	9	9.2	5.18								
33 BRINGING CABLES TO SLIDE HANDLE WITH YELLOW ZIP TIE	4	4.5	4.5	4.3	4.5	4.6	4.4	4.4	4.4	4.4	4.3	4.6	4.4	4.1	4.4	4.6	4.6	4.3	4.4	4.2	4.36									
34 FIXING SAFETY BELT CABLE WITH CLIP ON ZIP TIES	9.7	9.6	9.9	9.7	9.7	9.4	9.5	9.7	9.8	9.9	9.8	9.8	9.9	9.8	9.9	9.9	9.7	9.8	9.8	9.8	9.79									
35 CUTTING SURPLUSES OF ZIP TIES	2.9	3.2	3.1	3.1	3.1	3.1	2.9	2.9	2.9	3	2.9	3	3	3	2.9	3.1	3	3	3	3.2	2.96									
36 LANDING PLATFORM	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.00									
37 SCREWING PORTHOLE COVER SCREW	0	0	0	0	7.9	7.6	7.9	7.9	8	8.2	0	0	0	0	0	0	0	0	0	0	0.33									
38 INSERTING PORTHOLE COVER CORK	0	0	0	0	1.5	1.5	1.6	1.4	1.6	1.5	0	0	0	0	0	0	0	0	0	0	0.61									
39 ROTATING PLATFORM (H9)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.00									
40 SCREWING FRONT SCREW OF OUTER CASING	4.8	4.8	4.9	4.8	4.8	5.2	5.4	5.1	5	4.7	4.8	5	5	5	4.7	4.8	5	4.8	4.9	5	4.86									
41 TORCHING SAFETY BELT SCREW	6.3	6.2	6.1	5.8	6	5.8	5.9	5.8	6	5.8	5.8	5.9	5.8	6	6.2	6.3	6	6	6.1	6.3	5.99									
42 TORCHING TILT ADJUSTMENT LEVER SCREW	0	0	0	0	6	5.9	6.1	6.1	6.1	5.9	0	0	0	0	6	5.9	6	6	5.9	6.1	2.61									
43 MARKING SAFETY BELT SCREW WITH TORQUE PEN AFTER TORCHING	3.7	3.9	3.8	4	3.9	3.6	3.8	3.9	4.2	4	4	4	4	4	3.8	4.1	3.9	4.1	4.1	3.9	3.86									
44 MARKING TILT ADJUSTMENT LEVER SCREW WITH TORQUE PEN AFTER TORCHING	0	0	0	0	3.1	3.5	3.2	3	3	2.8	0	0	0	0	2.8	2.9	3	3	3.1	2.9	1.31									
45 ROTATING PLATFORM (H9)	0	0	0	0	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0.88									
46 ASSEMBLE ARMREST	0	0	0	0	7.8	9	0	8.9	8.7	0	0	0	0	0	0	0	0	0	0	0	3.62									
47 INSERTING ARMREST PEN	0	0	0	4.8	5	5.3	0	5.4	5.6	0	0	0	0	0	0	0	0	0	0	0	2.26									
48 ROTATING PLATFORM (H9)	0	0	0	0	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0.88									
49 BORN	41.9	41.5	41.8	41.7	41.8	41.8	41.7	41.7	41.5	41.7	41.7	41.4	41.7	41.8	41.9	41.4	41.7	41.4	41.5	41.3	41.78									
50 SAFETY BELT BUCKLE TEST	9.7	9.5	9.7	9.9	10.2	10.1	9.9	9.9	9.9	10	9.9	9.9	9.7	9.9	9.8	9.9	9.7	9.9	9.7	9.8	9.85									
51 SIKEMO TEST	0	0	17.6	17.9	0	0	0	17.9	18.1	18.2	0	0	18	18.1	0	0	18	18.4	18	17.9	9.43									
52 HEATER TEST	0	20.4	0	20.3	0	20.3	20.1	0	19.9	20.1	0	20.1	0	19.9	0	20.1	0	20.1	0	20.1	0.17									
53 ELECTRICAL PORTHOLE TEST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00									
54 TORCHING THE TEST LABEL	0	5.2	5	5.3	0	5.3	5.1	4.8	4.8	4.9	0	4.9	4.7	5	0	4.8	4.9	4.9	5	4.8	2.78									
55 FINAL CONTROL	42.3	42.6	42.8	42.6	42.5	42.9	42.7	42.9	43	42.7	42.4	42.2	42.5	42.2	42.5	42.3	42.8	42.5	42.6	42.6	42.63									
56 PACKAGING	8.4	8.9	8.3	8.2	8.6	8.9	8.7	8.5	8.8	8.5	8.7	8.6	8.6	8.5	8.2	8.3	8.3	8.6	8.4	8.6	8.49									

Table 4.26 Distribution of the average production amount per month on the basis of model and shift

MODEL TYPE	DESCRIPTION CODE OF PRODUCT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MCV	Aa WITHOUT OPTION	78	119	78	119	124	101	96	86	104	104	121	80	87	102	89
	Ab HEATER	2	4					5	157					108	123	128
	A1a SIDE BAG	108	139	122	112	108	128	133	162	117	122	151	106	156	124	122
	A1b SIDE BAG+HEATER+ARMREST	1	5	5	9	2	2			2	2			105	148	108
	A2a TILT MECHANISM+ARMREST+PORTHOLE	140	128	135	118	103	123	90	129	106	110	98	120	94	103	110
	A2b TILT MECHANISM+HEATER+ARMREST+PORTHOLE	2	3							3	2	5	2	3	3	4
	A2c TILT MECHANISM+HEATER+PORTHOLE	150	138	174	159	150	154	160	172	161	179	138	190	150	174	153
	A3a TILT MECHANISM+SIDE BAG+ARMREST+PORTHOLE	1	5	8	2					2	2	2			1	2
	A3b TILT MECHANISM+SIDE BAG+HEATER+ARMREST+PORTHOLE	47	72	41	71	84	60	58	51	57	63	73	38	11	51	71
	A3c TILT MECHANISM+SIDE BAG+HEATER+PORTHOLE	46	36	38	34	41	42	34	47	46	53		10		45	41
D200	A1a SIDE BAG	6	5	5	4	3	5	4	5	4	4	5	4	5	3	5
	A1b SIDE BAG+HEATER	2								2	1	3	2		4	4
	A2a TILT MECHANISM	4	4	5	6	5	5	6	6	5	6	4	7	5	6	5
	A2b TILT MECHANISM+HEATER									1					1	
	A2c TILT MECHANISM+SIDE BAG+HEATER+ELECTRICAL PORTHOLE	3	1	1	1	1	1	1	2	2	1	2	1	2	1	1
	A3a TILT MECHANISM+SIDE BAG+HEATER	585	664	604	642	0	622	621	583	662	603	645	584	664	599	648
	A3b TILT MECHANISM+SIDE BAG+ELECTRICAL PORTHOLE	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
	TOTAL NUMBER OF PRODUCTION IN A DAY	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
MCV	Aa WITHOUT OPTION	78	119	78	119	124	101	96	86	104	104	121	80	87	102	89
	Ab HEATER	2	4					5	157					108	123	128
	A1a SIDE BAG	108	139	122	112	108	128	133	162	117	122	151	106	156	124	122
	A1b SIDE BAG+HEATER+ARMREST	1	5	5	9	2	2			2	2			105	148	108
	A2a TILT MECHANISM+ARMREST+PORTHOLE	140	128	135	118	103	123	90	129	106	110	98	120	94	103	110
	A2b TILT MECHANISM+HEATER+ARMREST+PORTHOLE	2	3							3	2	5	2	3	3	4
	A2c TILT MECHANISM+HEATER+PORTHOLE	150	138	174	159	150	154	160	172	161	179	138	190	150	174	153
	A3a TILT MECHANISM+SIDE BAG+ARMREST+PORTHOLE	1	5	8	2					2	2	2			1	2
	A3b TILT MECHANISM+SIDE BAG+HEATER+ARMREST+PORTHOLE	47	72	41	71	84	60	58	51	57	63	73	38	11	51	71
	A3c TILT MECHANISM+SIDE BAG+HEATER+PORTHOLE	46	36	38	34	41	42	34	47	46	53		10		45	41
D200	A1a SIDE BAG	6	5	5	4	3	5	4	5	4	4	5	4	5	3	5
	A1b SIDE BAG+HEATER	2								2	1	3	2		4	4
	A2a TILT MECHANISM	4	4	5	6	5	5	6	6	5	6	4	7	5	6	5
	A2b TILT MECHANISM+HEATER									1					1	
	A2c TILT MECHANISM+SIDE BAG+HEATER+ELECTRICAL PORTHOLE	3	1	1	1	1	1	1	2	2	1	2	1	2	1	1
	A3a TILT MECHANISM+SIDE BAG+HEATER	585	664	604	642	0	622	621	583	662	603	645	584	664	599	648
	A3b TILT MECHANISM+SIDE BAG+ELECTRICAL PORTHOLE	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
	TOTAL NUMBER OF PRODUCTION IN A DAY	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
MCV	Aa WITHOUT OPTION	78	119	78	119	124	101	96	86	104	104	121	80	87	102	89
	Ab HEATER	2	4					5	157					108	123	128
	A1a SIDE BAG	108	139	122	112	108	128	133	162	117	122	151	106	156	124	122
	A1b SIDE BAG+HEATER+ARMREST	1	5	5	9	2	2			2	2			105	148	108
	A2a TILT MECHANISM+ARMREST+PORTHOLE	140	128	135	118	103	123	90	129	106	110	98	120	94	103	110
	A2b TILT MECHANISM+HEATER+ARMREST+PORTHOLE	2	3							3	2	5	2	3	3	4
	A2c TILT MECHANISM+HEATER+PORTHOLE	150	138	174	159	150	154	160	172	161	179	138	190	150	174	153
	A3a TILT MECHANISM+SIDE BAG+ARMREST+PORTHOLE	1	5	8	2					2	2	2			1	2
	A3b TILT MECHANISM+SIDE BAG+HEATER+ARMREST+PORTHOLE	47	72	41	71	84	60	58	51	57	63	73	38	11	51	71
	A3c TILT MECHANISM+SIDE BAG+HEATER+PORTHOLE	46	36	38	34	41	42	34	47	46	53		10		45	41
D200	A1a SIDE BAG	6	5	5	4	3	5	4	5	4	4	5	4	5	3	5
	A1b SIDE BAG+HEATER	2								2	1	3	2		4	4
	A2a TILT MECHANISM	4	4	5	6	5	5	6	6	5	6	4	7	5	6	5
	A2b TILT MECHANISM+HEATER									1					1	
	A2c TILT MECHANISM+SIDE BAG+HEATER+ELECTRICAL PORTHOLE	3	1	1	1	1	1	1	2	2	1	2	1	2	1	1
	A3a TILT MECHANISM+SIDE BAG+HEATER	585	664	604	642	0	622	621	583	662	603	645	584	664	599	648
	A3b TILT MECHANISM+SIDE BAG+ELECTRICAL PORTHOLE	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
	TOTAL NUMBER OF PRODUCTION IN A DAY	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
MCV	Aa WITHOUT OPTION	78	119	78	119	124	101	96	86	104	104	121	80	87	102	89
	Ab HEATER	2	4					5	157					108	123	128
	A1a SIDE BAG	108	139	122	112	108	128	133	162	117	122	151	106	156	124	122
	A1b SIDE BAG+HEATER+ARMREST	1	5	5	9	2	2			2	2			105	148	108
	A2a TILT MECHANISM+ARMREST+PORTHOLE	140	128	135	118	103	123	90	129	106	110	98	120	94	103	110
	A2b TILT MECHANISM+HEATER+ARMREST+PORTHOLE	2	3							3	2	5	2	3	3	4
	A2c TILT MECHANISM+HEATER+PORTHOLE	150	138	174	159	150	154	160	172	161	179	138	190	150	174	153
	A3a TILT MECHANISM+SIDE BAG+ARMREST+PORTHOLE	1	5	8	2					2	2	2			1	2
	A3b TILT MECHANISM+SIDE BAG+HEATER+ARMREST+PORTHOLE	47	72	41	71	84	60	58	51	57	63	73	38	11	51	71
	A3c TILT MECHANISM+SIDE BAG+HEATER+PORTHOLE	46	36	38	34	41	42	34	47	46	53		10		45	41
D200	A1a SIDE BAG	6	5	5	4	3	5	4	5	4	4	5	4	5	3	5
	A1b SIDE BAG+HEATER	2								2	1	3	2		4	4
	A2a TILT MECHANISM	4	4	5	6	5	5	6	6	5	6	4	7	5	6	5
	A2b TILT MECHANISM+HEATER									1					1	
	A2c TILT MECHANISM+SIDE BAG+HEATER+ELECTRICAL PORTHOLE	3	1	1	1	1	1	1	2	2	1	2	1	2	1	1
	A3a TILT MECHANISM+SIDE BAG+HEATER	585	664	604	642	0	622	621	583	662	603	645	584	664	599	648
	A3b TILT MECHANISM+SIDE BAG+ELECTRICAL PORTHOLE	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
	TOTAL NUMBER OF PRODUCTION IN A DAY	1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241
		1249	1246	0	1243	1245	1248	1248	1248	1247	1247	0	1244	1247	1241	1241

APPENDICES B

The Main Coding Page of MATLAB (neighborhood)

```

clear;clc;
station =load ('models/station.txt');
demand = load ('models/demand.txt');
disp('--> Initial station assignment:');
disp(' ');
cycle = 94;
[m,ddd] = size(station);
attack = zeros(1,11);
attackcount = 1;
slevel = 0;
modelarray = {'D200_A1a','D200_Aa','MCV_A1a','MCV_Aa'};
%modelarray =
{'D200_A1b','D200_Ab','D200_A2a','D200_A2b','D200_A3a','D200_A3b','D
200_A3c','D200_A3d','MCV_A1b','MCV_Ab','MCV_A2a','MCV_A2b','MCV_A2c'
,'MCV_A3a','MCV_A3b','MCV_A3c'};

[ddd,n] = size(modelarray);
suitablesolution = 1;
repeatoperation = true;
while repeatoperation

    measure = 0;
    botttime = 0;
    idletime = 0;

    disp(station_display(station));

    sonstat = station;
    yedstat = station;
    for i=1:n

        disp(' ');
        disp(['--[ ', modelarray{i}, ' ]-----
----']]);
        result = bottleneck(modelarray{i},cycle,yedstat);
        disp(result);

        [ddd,column] = size(result);

        for j=1:column
            if result(2,j)==0
                attackcount = attackcount+1;
                attack(attackcount,1) = i;
                attack(attackcount,2) = j;
                attack(attackcount,3) = result(3,j);
                attack(attackcount,4) = result(1,j);
                attack(attackcount,5) = j+1;
                attack(attackcount,10) = slevel+1;
                attack(attackcount,11) = slevel;

                attackcount = attackcount+1;
                attack(attackcount,1) = i;
                attack(attackcount,2) = j;
                attack(attackcount,3) = result(3,j);

```

```

        attack(attackcount,4) = result(1,j);
        attack(attackcount,5) = j-1;
        attack(attackcount,10) = slevel+1;
        attack(attackcount,11) = slevel;
    end
end

for j=1:column
    if result(2,j)==0
        measure = measure + (demand(i,2)*result(4,j));
        botttime = botttime + result(4,j);
    else
        idletime = idletime + result(4,j);
    end
end
end

end

attack(1,6) = efficiency(station,cycle);
attack(1,7) = measure;
attack(1,8) = botttime;
attack(1,9) = idletime*-1;
attack(1,10) = 0;
attack(1,11) = -1;
slevel = slevel+1;

disp(['Efficiency: %', num2str(efficiency(station,cycle))]);
disp(['Bottleneck measure: ', num2str(measure)]);
disp(['Bottleneck time: ', num2str(botttime), ' sec ']);
disp(['Idle time: ', num2str(idletime*-1), ' sec ']);

disp(' ');
disp('--[ Bottleneck Situations ]-----');
count = 0;
levelarray = find(attack(:,10)== slevel);
levelarray = levelarray';
for i=levelarray
    count=count+1;
    disp([' Situation ' , num2str(count) , ': In station ' ,
num2str(attack(i,2)) , ' task ' , num2str(attack(i,3)) , ' of ' ,
modelarray{attack(i,1)} , ' is moved to station ' ,
num2str(attack(i,5))]);
end

disp(' ');
disp('--[ Checking Bottleneck Situations ]-----');
count = 0;

for i=levelarray
    count=count+1;
    disp(' ');
    disp([' Situation ' , num2str(count) , ': In station ' ,
num2str(attack(i,2)) , ' task ' , num2str(attack(i,3)) , ' of ' ,
modelarray{attack(i,1)} , ' is moved to station ' ,
num2str(attack(i,5))]);
    yedstat = station;
end

```

```

way = sign(attack(i,5)-attack(i,2))*0.5 + 1.5;

situation = true;
    if way==1
        srow = find(yedstat(:,2)==attack(i,3));
        if yedstat(srow,11) >= attack(i,5)
            situation = false;
        end
    end

    if attack(i,5)~=0 && situation
        yedstat =
go_neighbor(attack(i,3),attack(i,2),yedstat,cycle,way);

        if yedstat==0
            disp('This move is not possible. ');
        else
            disp(station_display(yedstat));

            disp('--> This placement is controlled on models. ');
            measure = 0;
            botttime = 0;
            idletime = 0;

            for j=1:n
                disp(' ');
                disp(['--[ ', modelarray{j}, ' ]-----
-----']);
                answer =
bottleneck(modelarray{j},cycle,yedstat);
                disp(answer);

                [ddd,column] = size(answer);

                for k=1:column
                    if answer(2,k)==0
                        measure = measure +
(demand(j,2)*answer(4,k));
                        botttime = botttime + answer(4,k);
                    else
                        idletime = idletime + answer(4,k);
                    end
                end
            end

            end

            attack(i,6) = efficiency(yedstat,cycle);
            attack(i,7) = measure;
            attack(i,8) = botttime;
            attack(i,9) = idletime*-1;

            disp(['Efficiency: %',
num2str(efficiency(yedstat,cycle))] );
            disp(['Bottleneck measure: ', num2str(measure)] );
            disp(['Bottleneck time: ', num2str(botttime), ' sec
']);
            disp(['Idle time: ', num2str(idletime*-1), ' sec ']);

```

```

        end

    else
        disp('This move is not possible. ');
    end

end

[dimension,ddd] = size(attack);
situations = zeros(1,5);

situations(1,1) = attack(suitablestolution,7);
situations(1,2) = attack(suitablestolution,8);
situations(1,3) = attack(suitablestolution,9);
situations(1,4) = attack(suitablestolution,4);
situations(1,5) = attack(suitablestolution,10);
situations(1,6) = suitablestolution;

hline = 1;
for i=find(attack(:,10)==slevel) '
    if attack(i,7) ~= 0
        hline = hline + 1;
        situations(hline,1) = attack(i,7);
        situations(hline,2) = attack(i,8);
        situations(hline,3) = attack(i,9);
        situations(hline,4) = attack(i,4);
        situations(hline,5) = attack(i,10);
        situations(hline,6) = i;
    end
end

[m,j] = min(situations(:,1));

i=0;
[stoplevel,ddd]=size(situations);

while i~=stoplevel
    i=i+1;
    if situations(i,1) > m
        situations(i,:)=[];
        i=i-1;
        [stoplevel,ddd] = size(situations);
    end
end

[stoplevel,ddd] = size(situations);

if stoplevel == 1
    if situations(1,6) == suitablestolution
        repeatoperation = false;
    else
        suitablestolution = situations(1,6);
        repeatoperation = true;
    end
else

```

```

[m,j] = min(situations(:,2));

i=0;
[stopleveline,ddd]=size(situations);

while i~=stopleveline
    i=i+1;
    if situations(i,2) > m
        situations(i,:)=[];
        i=i-1;
        [stopleveline,ddd] = size(situations);
    end
end

[stopleveline,ddd] = size(situations);

if stopleveline == 1
    if situations(1,6) == suitablestolution
        repeatoperation = false;
    else
        suitablestolution = situations(1,6);
        repeatoperation = true;
    end
else

    [m,j] = min(situations(:,3));

    i=0;
    [stopleveline,ddd] = size(situations);

    while i~=stopleveline
        i=i+1;
        if situations(i,3) > m
            situations(i,:)=[];
            i=i-1;
            [stopleveline,ddd] = size(situations);
        end
    end

    [stopleveline,ddd] = size(situations);

    if stopleveline == 1
        if situations(1,6) == suitablestolution
            repeatoperation = false;
        else
            suitablestolution = situations(1,6);
            repeatoperation = true;
        end
    else

        [m,j] = min(situations(:,4));

        i=0;
        [stopleveline,ddd] = size(situations);

        while i~=stopleveline

```

```

        i=i+1;
        if situations(i,4) > m
            situations(i,:)=[];
            i=i-1;
            [stopline,ddd] = size(situations);
        end
    end
end

[stopline,ddd] = size(situations);

if stopline == 1
    if situations(1,6) == suitablestolution
        repeatoperation = false;
    else
        suitablestolution = situations(1,6);
        repeatoperation = true;
    end
else
    suitablestolution = situations(1,6);
    repeatoperation = true;

end

end

end

end

if repeatoperation
    way = sign(attack(suitablestolution,5)-
attack(suitablestolution,2))*0.5 + 1.5;
    station =
go_neighbor(attack(suitablestolution,3),attack(suitablestolution,2),st
ation,cycle,way);
    if station==0
        disp('A null value returned.');
```

else

```

        disp(' ');
        disp(' ');

disp('=====');
    write1 = modelarray{attack(suitablestolution,1)};
    write2 = attack(suitablestolution,2);
    write3 = attack(suitablestolution,3);
    write4 = attack(suitablestolution,5);
    disp(['In station ', num2str(write2) , ' task ' ,
num2str(write3) , ' of ' , write1 , ' is moved to station ' ,
num2str(write4)]);

    disp('Starting scan on all models for new attack...');
```

end

```

disp('=====');
end
end
```

```

end
disp(' ');
disp('=====
')
disp(' ');
if situations(1,6) == 1
    disp('The optimal solution is the initial solution.');
```

```

else
    disp(['The optimal solution: ']);
    if attack(situations(1,6),2) > attack(situations(1,6),5)
        way=1;
    else
        way=2;
    end
    sonstat =
go_neighbor(attack(situations(1,6),3),attack(situations(1,6),2),station,
cycle,way);
    disp(station_display(sonstat));

    for i=1:n
        disp(' ');
        disp(['--[ ', modelarray{i}, ' ]-----
----']);
        result = bottleneck(modelarray{i},cycle,sonstat);
        disp(result);
    end

    disp(['Efficiency: %', num2str(efficiency(sonstat,cycle))]);
    disp(['Bottleneck measure: ', num2str(situations(1,1))]);
    disp(['Bottleneck time: ', num2str(situations(1,2)), ' sec ']);
    disp(['Idle time: ', num2str(abs(situations(1,3))), ' sec ']);
end

```


The Function Coding Page of MATLAB (station_display)

```
function result = station_display(sta)
    if nargin==1
        [m,ddd] = size(sta);
        [ddd,b] = mode(sta,1);
        table = zeros(b(1),max(sta(:,1)));
        for i=1:m
            a = sta(i,1);
            b = sta(i,2);

            for j=1:m
                if table(j,a)==0
                    x=j;
                    break;
                end
            end

            table(x,a) = b;
        end

        result = table;    else
            disp('ERROR: Missing Parameter Error Function Station
Display')
        end
    end
end
```

The Function Coding Page of MATLAB (go_neighbor)

```

function result = go_neighbor(is,stat,sta,cycle,way)

if nargin==5

    [row, col] = size(sta);
    backupsta = sta;
    idlesta = zeros(row,col);
    countidle = 1;
    sta(:,1) = 0;
    sta = sortrows(sta,-10);
    rpw = zeros(1,2);
    srow = find(sta(:,2)==is);

    ii = 1;

if way==2

    sta(srow,1) = stat+1;

    rpw(stat+1,2) = sta(srow,10);

while (min(idlesta(:,1))==0)

    variation = false;

    for location=1:row
        if sta(location,[1 3 4 5 6 7 8 9])== 0
            if rpw(ii,2)+sta(location,10)<=cycle
                sta(location,1) = ii;
                rpw(ii,2) = rpw(ii,2)+sta(location,10);

                idlesta(countidle,:) = sta(location,:);
                countidle=countidle+1;

                for correct=1:row
                    for column=3:9
                        if sta(correct,column)==sta(location,2);
                            sta(correct,column)=0;
                        end
                    end
                end

                variation = true;

                break;
            end
        end
    end

end

if variation == false
    ii=ii+1;
    rpw(ii,1)=ii;
end
end

```

```

        if ii==stat+1
            for correct=1:row
                for column=3:9
                    if sta(correct,column)==sta(srow,2);
                        sta(correct,column)=0;
                    end
                end
            end
            idlesta(countidle,:) = sta(srow,:);
            countidle=countidle+1;
        end
    end
else
    if stat>1

        if sta(srow,11)>=stat-1
            disp('aa');
            sta(srow,1) = stat-1;
            rpw(stat-1,2) = sta(srow,10);
            while (min(idlesta(:,1))==0)
                variation = false;
                for location=1:row
                    if sta(location,[1 3 4 5 6 7 8 9])== 0
                        if rpw(ii,2)+sta(location,10)<=cycle
                            sta(location,1) = ii;
                            rpw(ii,2) = rpw(ii,2)+sta(location,10);
                            idlesta(countidle,:) = sta(location,:);
                            countidle=countidle+1;
                            for correct=1:row
                                for column=3:9
                                    if
sta(correct,column)==sta(location,2);
                                        sta(correct,column)=0;
                                    end
                                end
                            end
                            variation = true;
                        break;
                    end
                end
            end

            if variation == false
                ii=ii+1;
                rpw(ii,1)=ii;
                if ii==stat+1
                    for correct=1:row
                        for column=3:9
                            if sta(correct,column)==sta(srow,2);
                                sta(correct,column)=0;
                            end
                        end
                    end
                    idlesta(countidle,:) = sta(srow,:);
                end
            end
        end
    end
end

```

```
        countidle=countidle+1;
    end
end
end

else
    idlesta = zeros(1,1);
    countidle = 0;
end
else
    idlesta = zeros(1,1);
    countidle = 0;
end
end

if countidle > 1
    for i=1:row
        ara = backupsta(:,2)==idlesta(i,2);
        idlesta(i,3:9) = backupsta(ara,3:9);
    end
end

result = idlesta;

else
    disp('ERROR: Missing Parameter Error Function to Go Neighbors')
end
end
```

The Function Coding Page of MATLAB (bottleneck)

```

function result = bottleneck(modelname,loop,assign)

if nargin==3
    address = ['models/' num2str(modelname) '.txt'];
    productmodel = load(address);
    sta = 1;
    [m,ddd] = size(assign);
    sumloop = productmodel(1,2);
    lastwork = 1;
    table = zeros(4,max(assign(:,1)));
    for i=2:m
        if assign(i,1) == assign(i-1,1)
            sumloop = sumloop + productmodel(assign(i,2),2);
            if productmodel(assign(i,2),2)~= 0
                lastwork = assign(i,2);
            end
        else
            table(1,sta) = sumloop;
            if sumloop > loop+0.000001
                table(2,sta) = 0;
            else
                table(2,sta) = 1;
            end

            table(3,sta) = lastwork;
            table(4,sta) = table(1,sta)-loop;
            sta = sta+1;
            sumloop = productmodel(assign(i,2),2);

            if productmodel(assign(i,2),2)~= 0
                lastwork = assign(i,2);
            end
        end
    end

    if i==m
        if sumloop > loop+0.000001
            table(2,sta) = 0;
        else
            table(2,sta) = 1;
        end
        table(1,sta) = sumloop;
        table(3,sta) = lastwork;
        table(4,sta) = table(1,sta)-loop;
    end

    end

    result = table;
else
    disp('ERROR: Missing Parameter Error Function Bottleneck')
end

```

The Function Coding Page of MATLAB (efficiency)

```
function result = efficiency(sta,loop)
if nargin==2
    tuz = sum(sta(:,10));
    countsta = max(sta(:,1));

    result = 100*(1-(((loop*countsta)-tuz)/(loop*countsta)));

else
    disp('ERROR: Missing Parameter Error Function Efficiency')
end
end
```