

KADIR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING



A DECISION SUPPORT SYSTEM FOR ASSEMBLY BALANCING PROBLEM

GRADUATE THESIS

IBRAHIM UBA RINGIM

AUGUST, 2015

Ibrahim Uba Ringim

M.S.

2015

A DECISION SUPPORT SYSTEM FOR ASSEMBLY LINE BALANCING

IBRAHIM UBA RINGIM

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
INDUSTRIAL ENGINEERING

KADIR HAS UNIVERSITY

August, 2015

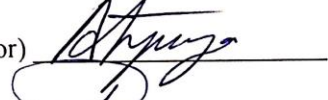
KADIR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

A DECISION SUPPORT SYSTEM FOR ASSEMBLY LINE BALANCING

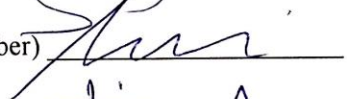
IBRAHIM UBA RINGIM

APPROVED BY:

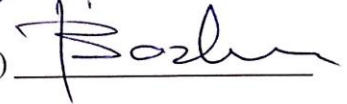
Asst. Prof. Dr. Ahmet YÜCEKAYA (Thesis Advisor)



Asst. Prof. Dr. Emre ÇELEBİ (Jury Member)



Asst. Prof. Dr. Zeki BOZKUŞ (Jury Member)



APPROVAL DATE: 30/AUGUST/2015

“I, Ibrahim Uba Ringim, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.”

IBRAHIM UBA RINGIM & SIGNATURE

ABSTRACT

A DECISION SUPPORT SYSTEM FOR ASSEMBLY LINE BALANCING

Ibrahim Uba Ringim

Master of Science in Industrial Engineering

Advisor: Doc. Dr. Ahmet Yucekaya

August, 2015

Assembly line balancing problems are generally considered to be complicated in real life. Like most complicated real life assembly line balancing problems obtaining a good solution is much easier than finding an optimal solution especially with big size problems. As a result, many heuristic approaches have been developed to find good optimal solutions to those problems. In this study, we develop a decision support system that solves a deterministic assembly line balancing problem using three heuristic approaches. The objectives considered are: minimizing the number of workstations, maximization of line efficiency and minimization of balance delay. Our aim is using the decision support system created; user can enter any value into the system and obtain 3 different results. The results obtained are feasible enough which shows that the decision support system works and can be able to solve even larger problems if the correct formula is applied

Keywords: Assembly Line Balancing, Heuristic Approach, Decision Support System.

ÖZET

YÜKSEK LİSANS TEZİ

Ibrahim Uba Ringim

Endüstri Mühendisliği, Yüksek Lisans

Danışman: Doc. Dr. Ahmet Yucekaya

Ağustos, 2015

Montaj hattı dengeleme problemleri gerçek hayatta genellikle karmaşık olarak kabul edilir. Bir çok karmaşık gerçek hayat problemleri gibi iyi bir sonuç elde etmek özellikle büyük problemlere çözüm bulmaktan çok daha kolaydır. Sonuç olarak, çok sayıda sezgisel yaklaşımlar bu sorunlara uygun çözümler bulmak için geliştirilmiştir. Biz bu çalışmada, üç sezgisel yaklaşım kullanarak deterministik montaj hattı denge sorununu çözen karar sistemi geliştiriyoruz. Düşünülen hedefler şunlardır; çalışma istasyonlarının sayısının en aza indirilmesi, hat verimliliğinin üst seviyeye çıkarılması ve denge gecikmesinin en aza indirilmesidir. Amacımız, oluşturulan karar destek sistemini kullanarak, kullanıcının sisteme herhangi bir değeri girebilmesi ve üç farklı sonuç elde edebilmesidir. Sonuçların uygulanabilecek düzeyde mümkün olduğuna inanılmaktadır. Sonuçlar karar destek sisteminin çalıştığını ve doğru formül uygulandığı takdirde daha büyük problemleri bile çözebileceğini göstermektedir.

Anahtar Kelimeler: Montaj Hattı Dengeleme, Sezgisel Yaklaşım, Karar Verme Destek Sistemi.

Acknowledgements

First of all I want to start by saying thank you to almighty Allah for all the things He made me achieved through my entire life.

I would like to express special gratitude towards my parents, D.I.G Uba Bala Ringim and Hajiya Uwani Uba Ringim whose love, guidance and support have been the greatest asset in my life as well in this thesis. Also appreciate their patience for all the period I have been away because of studies.

I would also like to express my gratitude to my supervisor Ahmet Yucekaya, whose vast knowledge, guidance and assistance has helped me get through the difficult process while preparing my thesis report.

I would like to offer a special thank you to my friend Sara Ayanoglu who listened to my compliant and motivated me during this study and my very good friends Muhammad Kabir, Birol Sak and Adem for thier words of advice and support.

Table of Contents

Abstract	
Özet	
Acknowledgements	
List of Tables	xi
List of Figures	xii
List of Abbreviations	xiii
Chapter 1	1
1 Introduction	1
Chapter 2	3
2.1 Terminology Used For Assembly Lines	3
2.2 Overview of ALB	5
2.3 Literature Review	7
Chapter 3	10
3.1 Assembly line balancing problem	10
3.2 Heuristic approaches	12
3.3 Calculations Involved in Solving ALBP	14
3.3.1 Line Efficiency.....	14
3.3.2 Balance Delay (BD).....	14
3.3.3 Smoothness Index (SI)	15
3.3.4 Cycle Time (C).....	15
3.3.5 Min. No. Of workstation.....	16
3.4 Methods of Line Balancing and Decision Support System	16
3.4.1 Largest Candidate Rule	17
3.4.2 Kilbridge and Wester's Method.....	17
3.4.3 Rank Positional Weight Method (RPW).....	18
3.4.4 Example I	19
3.5 Decision support system	23
3.5.1 History	23
3.5.2 Characteristics.....	23
3.5.3 Who needs DSS?.....	25
Chapter 4	27
4.1 Problem Definition	27
4.2 Mathematical Formulation	27
4.3 An Example Problem	29
4.4 Solution Approach	33
4.4.1 Excel Implementation.....	33
Chapter 5	40
5.1 Case Study	40
5.2 The problem in study	40
5.3 Solving the Problem	42
5.3.1 Largest candidate rule solution.....	43
5.3.2 Kilbridge and Wester's solution.....	44
5.3.3 Ranked Positional Weight.....	45
5.4 Result Explanation	46

5.5 Final Analysis.....	48
Chapter 6	49
6.1 Conclusions and Further Research.....	49
References.....	52
Appendices	55

List of Tables

Table 3.1 Task times and precedence requirements of example.....	19
Table 3.2 Tasks arranged according to their task times.....	20
Table 3.3 Tasks assigned according to Largest candidate rule.....	20
Table 3.4 Tasks arranged according to their calculated weight.....	21
Table 3.5 Tasks assigned according to RPW method.....	21
Table 3.6 Tasks divided into columns.....	22
Table 3.7 Tasks assigned according to Kilbridge & Wester's method.....	22
Table 4.1 Task times and precedence requirement.....	30
Table 5.1 Job data for our case study.....	41
Table A.1 Jobs arranged in descending order according to task times.....	56
Table A.2 Jobs assigned to stations using largest candidate rule.....	57
Table A.3 Jobs assigned according to columns from figure 5.1	58
Table A.4 Jobs assigned to stations using Kilbrigde and Wester's	59
Table A.5 Jobs arranged in order of power weight value in RPW.....	60
Table A.6 Jobs assigned to stations using Ranked positional weigh.....	61

(Note: Table 1.1 indicates the first table in Chapter 1, Table 10.1 indicates the first table in Chapter 10 and Table A.1 indicates the first table in Appendix A.)

List of Figures

Figure 2.1 Example of precedence diagram.....	4
Figure 2.2 Types of assembly lines.....	6
Figure 3.1 Classification of ALBP and related solution procedures.....	12
Figure 3.2 Task times and precedence requirements of example I.....	19
Figure 3.3 Components of decision support system.....	25
Figure 4.1 The Precedence Diagram of our problem.....	30
Figure 4.2 First efficient solution using RPW method.....	31
Figure 4.3 Second efficient solution using Kilbridge and Wester's method.....	31
Figure 4.4 Third efficient solution using largest candidate rule.....	32
Figure 4.5 Components of our decision support system.....	33
Figure 4.6 Main page of decision support system.....	34
Figure 4.7 Data entry page of decision support system.....	35
Figure 4.8 Precedence Diagram of decision support system.....	36
Figure 4.9 Solution of problem using RPW in the system.....	36
Figure 4.10 Solution of problem using Kilbridge and Wester's method in the system.....	37
Figure 4.11 Solution of problem using largest candidate rule in the system.....	38
Figure 5.1 The Precedence Diagram of our case study.....	42

List of Abbreviations

ALB: Assembly line Balancing

SALBP: Simple Assembly line Balancing Problem

GALBP: Generalized Assembly line Balancing Problem

C: Cycle Time

DSS: Decision Support System

RPW: Ranked Positional Weight

Chapter 1

1 Introduction

In 1913, Ford Motor Company led by Henry Ford began taking its first steps toward inventing a moving line that will use conveyor belts to stream components past workers who perform one or more tasks each. This early manufacturing process made automobiles affordable to almost everybody and became the idea to build on for the entire industry. Ever since, balancing assembly lines has become one of the most important departments in manufacturing industries. It was designed to increase the efficiency and productivity to the way of manufacturing products.

Assembly line is a sequential progression of an input material that enters at a particular point then goes through one or more workstations usually by some type of transportation system. At each workstation one or multiple tasks are done. The product moves from one workstation to another along a line which could be straight, u-line or parallel and perform certain operations that are required according to precedence restrictions until the final assembly is done. A product is complete when all set of tasks or operations are done.

In general, the purpose of assembly line is to help mass production systems, where the goal is to find the order in which tasks are assigned to work stations or minimizing the total processing time, so that the main objective will be achieved.

Assembly line balancing problems (ALBP) are divided into two groups; the simple assembly line balancing problem (SALBP) and the general assembly line-balancing problem (GALBP) (Baybars 1986).

Generally there are two approaches to solve these problems; the exact method that gives an optimal and feasible solution, and heuristics methods or algorithms, those based on logic and common sense rather than mathematical proof. They do not guarantee an optimal solution but can certainly provide good results.

Maximization of production rate and minimization of the number of workstations are some of the objectives in assembly line.

In this paper a type-1 SALBP and type-E SALBP are studied. Thus, it is a single model line, with a fixed cycle time and the objectives are: minimization of the number of workstations, minimizing balance delay, and maximization of line efficiency. A decision support system (DSS) is created that requires the user to enter random task times to the assembly line balancing problem. The decision support system uses three heuristic methods and provides the solution for each of them and also displays the best solution depending on workstations used, line efficiency and balance delay. Type-1 SALBP and type-E SALBP are considered because those two types of problems provide us with similarities in respect to their objectives.

The three mentioned algorithms are: The largest candidate rule, Helgeson & Birnie (ranked positional weight) and Kilbridge and Wester (column)

This thesis includes 6 chapters that are organized as follows:

In chapter 2 the terminology used in assembly line balancing is introduced, along with the general overview of assembly lines. The literature review on assembly line balancing and heuristic approaches is reviewed.

In chapter 3 the assembly line balancing problem is described in details along with the techniques used to solve them. Moreover, the decision support system used in this study is introduced.

In chapter 4 the proposed solution approach is discussed along with problem definition and mathematical formulations.

In chapter 5 the case study is presented along with results.

In chapter 6 the conclusions and suggestion for further research are presented.

Chapter 2

2.1 Terminology Used For Assembly Lines

This chapter introduces the terminology used in ALBPs, give us a brief overview on ALBP, and give a review on heuristic approaches on solving ALBP.

Assembly is a form of production system that is defined as the aggregation of all required task to successfully form a product. Assembly usually occurs on assembly line. Assembly lines can be defined as a set of workstations that are sequentially arranged in an order either parallel, u-shape or round and are connected by a means of a transportation systems e.g. conveyor belt (Boysen, Fliedner, and Scholl 2007).

Manufacturing a product on assembly line involves categorizing your total job into small sub operations. A task is the smallest work element of the total work content. The time it takes for a certain task or operation to be done is called task time or processing time. Depending on the activities that are involved in a task, sometimes several equipment are required.

Workstation is the area where certain operations are repeatedly performed, either by machines or special operators using equipment. The work content (WC) of a station is the addition of all task times of the assigned tasks to each workstation.

Cycle time (C) is the time given to each workstation to perform a task. In most problems the cycle time is considered to be more than the largest time of any task time and should be smaller than the summation of the overall task times. Therefore, the production rate of assembly line is represented by $1/C$. The idle time of any station is the difference between cycle time and the work content of that particular station.

Total work content (WTC) can be explained as the addition of all work content of each workstation; it can also be explained as the total sum of all task times in the assembly line. The sum of all idle times of all station is referred to as total slack time or balance delay. The maximum processing time to manufacture a product along a line is called assembly time. Assembly time is equal to multiplication of (C) with the number of workstations.

A precedence constraint explains order in which tasks are assigned with respect to industrial and organizational restrictions. Some tasks should not be done before a certain task is produced. Tasks that have to be done after a certain task are known as predecessors while tasks that can be done before a certain task are called successors.

A precedence diagram is use for summarizing or visualizing all the precedence relations. The figure below displays a sample of a precedence diagram with 9 tasks. Each element has a task time between 2 and 9 (mint, sec, etc.)

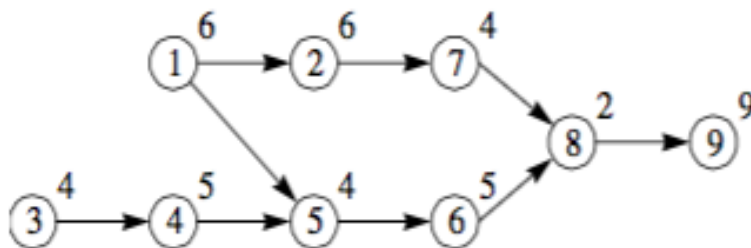


Figure 2.1 Example of precedence diagram

In the figure, the graph contains a node, which represents each task and the values inside the task are the processing times. The arc connecting tasks 1 and 2 exists because task 1 is predecessor of task 2. Accordingly, tasks 1, 4 and 5 are predecessors of task 6 and task 5 is its immediate predecessor. Task 9 is the successor of all tasks and the immediate successor of task 8.

2.2 Overview of ALB

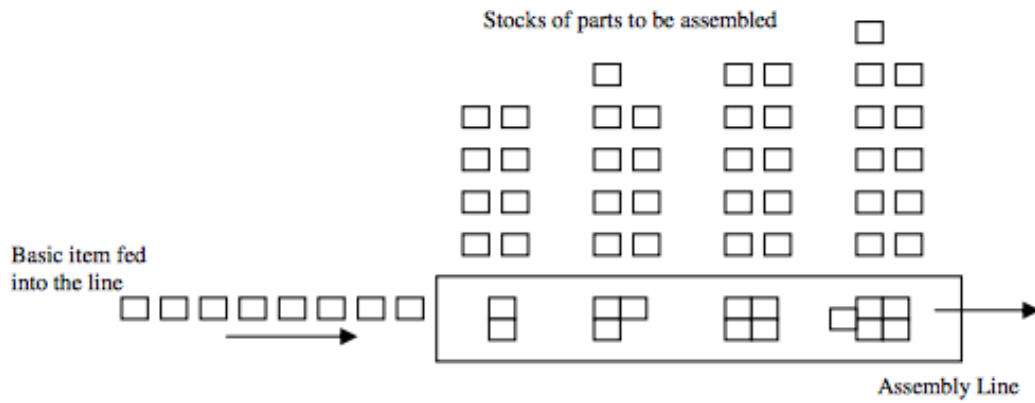
Assembly line manufacturing systems are used for manufacturing multiple types of product. All products have different characteristics, which require different production systems to produce.

There are three types of assembly line models depending on the type of product:

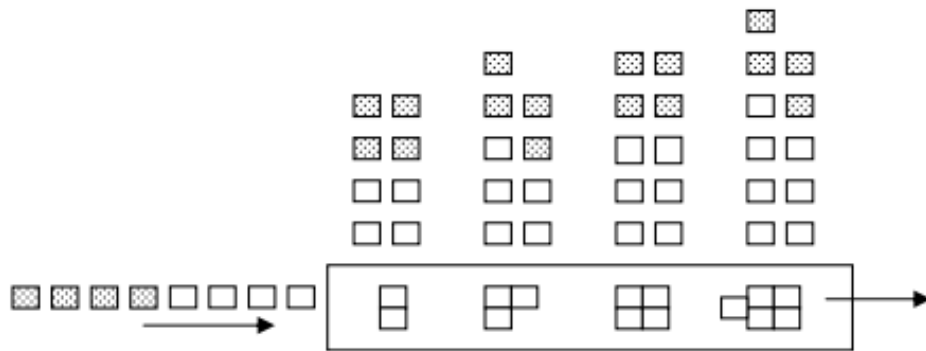
- Single model assembly line
- Multi model assembly line
- Mixed model assembly line

A single model has only one product assembled along the assembly line. If more than one product is manufactured on the same line in an intermixed sequence, then it is called multi model. The models of the products show small differences so that same operations are required for all products. In mixed model the products are manufactured on the same or multiple assembly lines. The difference from multi model is that the models of the products have significant differences between them, which make it necessary to arrange the lines in a way that it can be switched from one product to another.

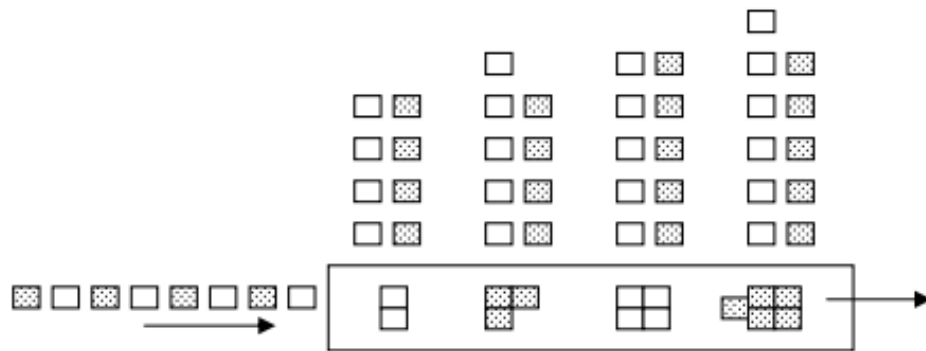
The figure below shows difference between the three models



a) Single-Model Assembly Line



b) Multi-Model Assembly Line



c) Mixed-Model Assembly Line

Figure 2.2 Types of Assembly Lines (Wild, 1972)

An assembly line can either be paced or unpaced lines. In paced assembly lines, the time spent in each workstation is fixed and limited to the cycle time. The cycle time will be the maximum time a task can spend in a workstation. “Due to the cycle time requirements, paced assembly lines have a fixed production rate” (Becker and Scholl 2006). In unpaced assembly lines, with regard to the absence of cycle time, the time a

task spends in each workstation is different. Due to the speed at which each workstation operates, some tasks need to wait then they can be processed to the next workstation or a station sometimes has to stay idle before the next task enters. Buffer stocks are required to partially overcome these difficulties between workstations.

Another characteristic of classifying assembly lines is through variation of task times or processing times. The task times can either be deterministic or stochastic. Deterministic task time is required when it is assumed that the manufacturing systems at each workstation are equipped with machines or robots that operate at a constant speed. However if reasonable changes of task times are to expected due to workers work rate, skills and motivation then stochastic task time is well fit. Beside “workers instability, systematic reductions are possible due to learning effects or successive improvements of the productions process” (Becker and Scholl 2006).

2.3 Literature Review

The literature discussed here are mostly about methods of solving assembly line balancing problem because enough studies have not been done regarding the topic presented in this paper.

Generally, ALBP works in deciding which task to be assigned to workstations. The main objective is minimizing the cycle time or number of workstations while also satisfying demand and regulations like the precedence constraints amongst tasks.

Groover (2008) named largest candidate, Helgeson and Birnie, and Kilbrigde and Wester as an example of heuristic methods that can be used to arrange and distribute task times along workstations in the systems (Becker and Scholl 2006).

(Becker and Scholl 2006) discuss a survey of SALB and gave many heuristics methods for most problems. They discussed a comprehensive survey on the GALBP. General problem characteristics such as cost function, equipment selection, parallel and U-shaped line layout were reviewed.

Mat, Muhammad and Law (2012) used largest candidate rule, Kilbride and Wester method and Helgeson and Birnie to balance the assembly line of a car manufacturing industry, with improving efficiency as the objective. The results were positive and believed to have improved the productivity as much as three times higher than before the study. The general idea in assembly line is to be able to arrange tasks to work stations in a structure that each total time of workstations should have an equal line cycle time (Dwivedi 2012).

Rubinotitz and Bukchin (1993) provide a study and present a heuristic method that is able to design and balance a robotic assembly line. The objective is to minimize the number of workstations and robots used. The restriction constraint requires a single equipment to be available for each workstation. Mahto, and Kumar (2012) studied an ALBP and applied Helgeson-Birnie heuristic method and Kilbridge-Wester heuristic method in designing assembly lines starting from the work breakdown structure to the final groupings of task at workstations. The objectives of the study were to optimize: “crew size, system utilization and the probability of tasks being completed within a certain time frame”.

Driscoll and Abdel-Shafi (1985) applied a balancing technique similar to RPW technique and linked it with simulation to judge the performance of several solutions.

Becker and Scholl (2006) conducted a literature survey and mentioned “parallel workstations and task times, cost synergies, processing alternative, zoning restrictions, stochastic and sequence-dependent processing times as well as U-shaped assembly

line” as the “additional technical or organizational aspects, which will heavily affect the structure of the planning problem”.

Scholl (1999) classified the objective of assembly line balancing problem into two categories: capacity oriented goals and cost oriented goals. Scholl and Becker (2006) provide a study on the state-of-the-art exact and heuristics solutions procedures for SALB. Also (Becker and Scholl 2006) present a survey on problems and methods of GALB

A study done by Ghosh and Gagnon (1989) classified ALBP into four categories: single model deterministic (SMD), single model stochastic (SMS), multi/mixed model deterministic (MMD) and multi/mixed stochastic model (MMS) (Gökçen, Ağpak et al. 2006).

More literature reviews about heuristic approaches are discussed in section 3.2.

Chapter 3

3.1 Assembly line balancing problem

ALBPs are classified into two categories: simple assembly line balancing problem, which is widely known as (SALBP) and Generalized assembly line balancing problems, known as (GALBP) (Baybars 1986).

SALBP considers assigning task elements to stations by using straight single lines that have precedence constraint and it is assumed that tasks are indivisible. The task times in SALBP are deterministic and the cycle time or number of workstation is fixed. There are four types of SALBP:

- SALBP-1: this type of problem has a fixed cycle time; it consists of assigning tasks to workstations. The general objective is minimizing the total number of stations with a given production rate.
- SALBP-2: it consists of assigning tasks to station with a fixed number of workstations given. The main objective is minimizing the C.
- SALBP-E: the objective here is minimizing the balance of the line. Thus, minimizing the C, along with total number of workstations.
- SALBP-F: this is a feasibility problem, where the objective is determining either if not there is feasible line regarding the combination between a given workstation and cycle time.

Only type 1 and E are considered in this thesis. They provide us with similar objective to what we want to achieve at the end of this study. The objective for SALBP-1 is to minimize the total number of stations while SALBP-E is to minimize the line efficiency. The relationship between these two objectives is that: to have an efficient

line balance, the number of stations has to be as minimum as possible which also is used in determining the line balance of the assembly line. So by considering those two types of SALPB we will be able to calculate all our objectives at the end of the study.

The other type are called GALBP, those are all types of problems that are not included or some assumptions of SALB are removed e.g. the task times are stochastic.

Most of these types of problems are “very large and contain problem extensions that might be relevant in practice including equipment selection, processing alternatives, assignment restriction etc.” (Becker and Scholl 2006). There are four types of GALBP base on (Baybars 1986) classification:

- U-line assembly line balancing problem (UALBP)
- Mixed model assembly line balancing problem (MALBP)
- Robotic assembly line balancing problem (RALBP)
- Multi-objective assembly line balancing problem (MOALBP)

For further survey on types of GALBP see Becker and Scholl (2006)

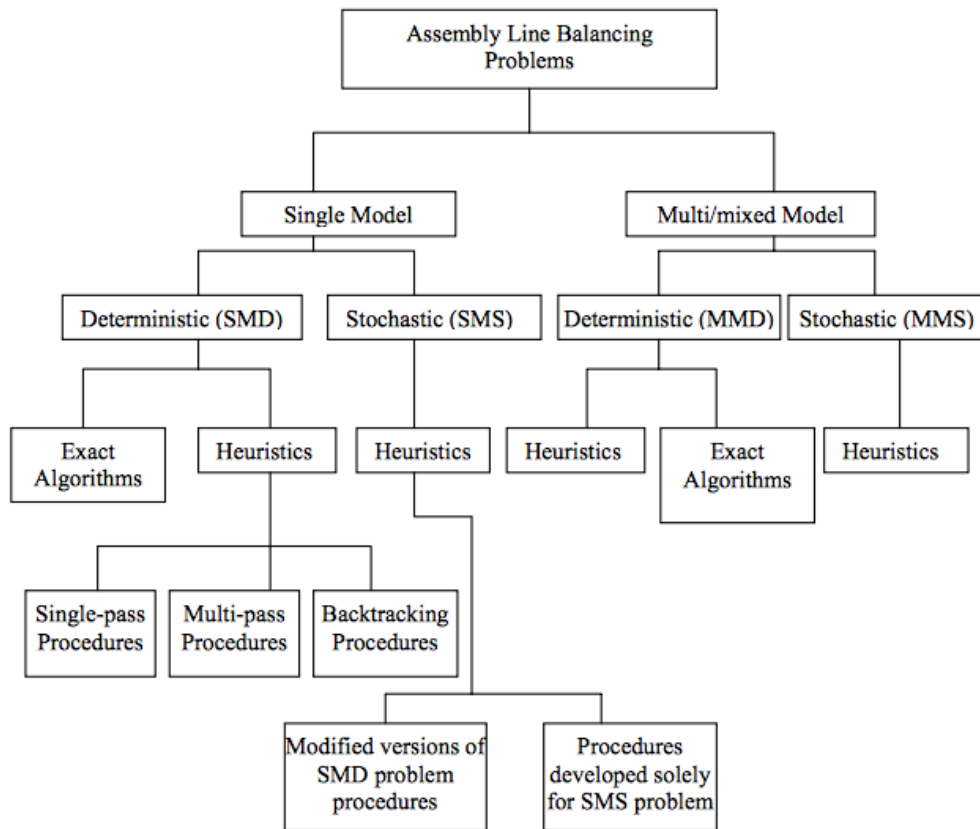


Figure 3.1 Classification of ALBP and related solution procedures

To solve ALBPs, several models and different solutions techniques have been developed. These solution techniques have been grouped into two categories, exact and heuristic methods. Branch and bound algorithm, integer programming, goal programming, dynamic programming, linear programming are some of the example of exact methods. On the other hand, a large variety of heuristic methods, like rank positional weight, random sampling, trade and transfer and largest candidate rule are developed.

3.2 Heuristic approaches

Solving an assembly line balancing problem is not always possible due to size of some certain problems. The size of some problems keep getting larger that solving them optimally becomes difficult and sometimes even impossible in an average

computational time. Hence, a lot of heuristic approaches have been developed so far to provide good solution and sometimes with better luck even an optimal solution. Finding optimal solution with most heuristic approaches is not guaranteed in reasonable time.

One of the earliest and best known heuristic method proposed was Ranked Positional Weight (RPW) due to Helgeson and Birnie (1961). The method constructs a single sequence where a task is prioritized depending on the cumulative task times of itself and its successors. Task that have more weight are assigned first. More information about this approach will be explained in details later in the study.

Hoffman (1963) proposed a heuristic that uses precedence matrix to develop all feasible assignments and the objective is to make assignment with lowest slack times. The name of the method is Successive Maximum Element Time Method.

Baybars (1986) developed a heuristic method in which he first reduces the size of the problem by eliminating some tasks.

He then breaks the problem into sub problems, solve each sub problem individually and finally put all the solutions to construct the main solution.

Raouf and Tsui (1980) developed the Critical Path Approach to help solve type I assembly line problems. The method starts by figuring out the critical path or area of the problem. The next step gives priority to those tasks that are on the critical area while assigning tasks to workstations.

3.3 Calculations Involved in Solving ALBP

As mentioned before, ideas behind assembly lines are minimization of the total number of workstations or minimize the total number of machines or workers to do a certain number of tasks in a certain amount of time. This can be referred to as minimization of balance delay. Balance delay can be explained as the degree of inefficiency along the assembly line, which occurs as a result of idle time because of incomplete distribution of work between stations. The following formulations are very much associated with the heuristic approach of assembly line balancing.

3.3.1 Line Efficiency

This is the ratio of total workstation time to the product of the C and the number of workstations. It can be expressed as:

$$LE = \left[\frac{\sum_{i=1}^k ST_i}{(K) \times (C)} \right] \times 100\%$$

Where ST_i = station time of station I

K = total number of work stations

C = Cycle Time

3.3.2 Balance Delay (BD)

This is the degree of inefficiency along the assembly line, which occurs as a result of idle time because of incomplete distribution of work between stations. It can be computed for the flow line as follows:

$$BD = (K) \times (C) - T_{wc} / (K) \times (C) \text{ or}$$

Simply, $BD = 1 - LE$

Where,

Twc = Total work content

K= total number of work stations

C = cycle time

LE= line efficiency

3.3.3 Smoothness Index (SI)

This index indicates the approximate smoothness of the ALB. A smoothness index of 0 represents a perfect balance. This is expressed as follows:

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

Where,

ST_{max} = maximum station time

ST_i = station time of station I

K= total number of workstations

3.3.4 Cycle Time (C)

It is also important to understand the idea behind the cycle time. In most heuristic approaches the C is considered to be more than the largest time of any task time and should be smaller than the summation of the overall task times, that is:

$$T_{\max} \leq C \leq \sum_{i=1}^N T_i$$

Where,

T_{max} = maximum work element time

T_i = time of task i

N = total number of tasks

C = cycle time

3.3.5 Min. No. Of workstation

No. WS = \sum task times / cycle time

3.4 Methods of Line Balancing and Decision Support System

In this section we are discussing the methods involved in solving our SALBP. We will take a look at three heuristic approaches, i.e., those based on common sense and logic rather than mathematical proof. They will surely provide acceptable results but optimality is not guaranteed. But before mentioning these methods, we should first understand the general steps of solving a SALBP. The steps are as follows:

- Determine task (operations)
- Determine sequence
- Draw precedence diagram
- Estimate task times
- Calculate C
- Calculate number of work stations
- Assign tasks
- Calculate line efficiency

As mentioned before, these are the general steps involved in solving SALBP. It should be noted that in between these general steps, there are many sub steps that must be completed. Below are the methods of line balancing.

3.4.1 Largest Candidate Rule

The reasoning behind this method is that by arranging the task times of the work elements in descending order, from largest to smallest, we will be able to understand which task should be assigned first. It is a heuristic method, which means it will give us a solution but it might not be the optimal solution. The steps are as follows:

- Lists all tasks in descending order.
- Assign elements to 1st station, from top to bottom of the list, minding constraints from the precedence diagram and make sure the sum of activities in each work station doesn't exceed the cycle time.
- Continue assigning work elements to the stations until no more tasks can be added without going beyond calculated C.
- Perform same steps for the remaining stations in the line continuously till all remaining tasks have been assigned.

3.4.2 Kilbridge and Wester's Method

This technique was first introduced in 1962. It has received a good deal of attention because it has been applied to several rather complicated line balancing problems with good success. In this heuristic technique tasks are assigned to stations according to their position in the precedence diagram. The tasks at the beginning of the precedence diagram are considered initially for entry into the solution. Generally, a better line balancing solution is acquired with this method when compared with the Largest Candidate Rule, where elements at the end of the precedence diagram might be first to consider because they have a larger task time values. Kilbridge and Wester's Method states three possible goals of the assembly line:

The procedure consists of the following steps:

- Design the precedence diagram in a way that vertically arranged nodes with identical precedence in columns.
- Arrange the tasks in ascending order according of their columns. Tasks that can be positioned in multiple columns should be listed in all columns to show the interchangeability of the task.
- When assigning tasks to work stations, start with the column I tasks. Proceed with the assigning technique in order of column number till the given C is reached. Continue with the same procedure until all tasks are assigned.

3.4.3 Rank Positional Weight Method (RPW)

In 1961, Helgeson and Bernie introduced a heuristic method called Ranked Positional Weight Method (RPW). In this method, each task is given a weight depending on the value of its task time and its position in the precedence diagram. The weight is calculated by adding the task time of that certain element and all the task time of elements that follow it in the precedence diagram.

In a sense this method combines the strategies of both Largest Candidate Rule and Kilbridge and Wester's Method, which are also heuristic methods of solving SALBPs. The steps are as follows:

- Determine the positional weight of each work element.
- Rank the tasks with respect to their positional weight in step 2. Tasks are ranked in ascending order depending on their weight. The task with most weight is ranked first and the task with least weight ranks last.
- Start assigning the tasks with most weight to the first station and continue in such manner.

- If at any workstation additional time remains after assignment of an operation, assign the succeeding ranked operation to the workstation, as long as the operation does not violate the precedence relationship diagram and the station time does not exceed the cycle time.
- Keep repeating the same steps (4 and 5) until all tasks are assigned.

3.4.4 Example I

Lets consider an example with 5 tasks. The production rate and available working hours are 300 units per day and 8 hours per day respectively. The table below illustrates the task times and precedence relations.

Task	Time	Immediate Precedence
A	0.1	-
B	0.7	A
C	1.0	B
D	0.5	C
E	0.2	D

Table 3.1 Task times and precedence requirements of example I

The following figure illustrates the precedence diagram.



Figure 3.2 Precedence diagram of example I

$$\text{Cycle time} = 480 \text{ mins/day} / 300 \text{ units/day} = 1.6 \text{ mins/unit}$$

$$\text{Min. No. Of stations} = 2.5 \text{ mins} / 1.6 \text{ mins} = 1.5625 \text{ or } 2 \text{ workstations}$$

Largest candidate solution:

We begin by arranging the task times in descending order. The table below illustrates the task times according to largest candidate rule.

Task	Time	Immediate precedence
C	1.0	B
B	0.7	A
D	0.5	C
E	0.2	D
A	0.1	-

Table 3.2 Tasks arranged according to their task times.

We continue by starting to assign task with the largest task times until all task are assign.

Station	Task	Time	Sum. Of task at station
1	A	0.1	
	B	0.7	0.8
2	C	1.0	
	D	0.5	1.5
3	E	0.2	0.2

Table 3.3 Task assigned according to Largest candidate rule

Efficiency = $2.5 / (3 * 1.6) = 52\%$

Balance delay = $1 - 52\% = 49\%$

We can see from table 3.2 that task C has the largest task time but our precedence requirement doesn't allow us to assign it to station until task A and B are completed. Also, we could assign task D and E to station 1 because it has enough idle time but task C has to be completed before you can start both task.

Ranked positional weight solution:

We begin by calculating the positional weight of each task and ranking the task base on the weight. The table below illustrates the information.

Task	Weight	Rank
A	2.5	1
B	2.4	2
C	1.7	3
D	0.7	4
E	0.2	5

Table 3.4 Tasks arranged according to their calculate weights

We then continue by assigning the task with most weight with respect to its precedence requirement. The solution in the example requires us to start from task 1. The table below illustrates the assigning of tasks to stations.

Station	Task	Time	Sum. Of task at station
1	A	0.1	
	B	0.7	0.8
2	C	1.0	
	D	0.5	1.5
3	E	0.2	0.2

Table 3.5 Tasks assigned to stations according to RPW method

Efficiency = $2.5 / (3 * 1.6) = 52\%$

Balance delay = $1 - 52\% = 49\%$

Kilbridge & Wester solution:

The first step requires us to start by dividing the precedence diagram into columns. By looking at our diagram we can easily divide it into 5 columns for since each task is the dependent on its predecessor. The table below illustrates the column each task belongs to.

Task	Column	Time
A	I	0.1
B	II	0.7
C	III	1.0
D	IV	0.5
E	V	0.2

Table 3.6 Tasks divided into columns

Now that we have divided each task to its column, we can start to assign the task to station. The Task at the first column will be assign first regardless of its precedence requirements. Table no shows the final assignment.

Station	Task	Time	Sum. Of task at station
1	A	0.1	
	B	0.7	0.8
2	C	1.0	
	D	0.5	1.5
3	E	0.2	0.2

Table 3.7 Tasks assigned according to Kilbrdge & Wester's method

Efficiency = $2.5 / (3 * 1.6) = 52\%$

Balance delay = $1 - 52\% = 49\%$

3.5 Decision support system

3.5.1 History

Decision support systems (DSS) are computer based application systems designed in a way that provides decision makers with many alternatives solutions to a problem. It can also be define as “a computer-based information system that supports business or organizational decision-making activities” (Wikipedia. retrieved 14:40, July 1, 2015)

The term decision support system was first introduced in the late 1970s by a British academic name Keen then working in the United States of America. Later around the same time Keen and Scott Morgan published a book called *Decision Support Systems: An Organizational Perspective*. They defined it in the book as “computer systems having a say in managerial decisions through providing analytical aids that can be of importance value but the final and essential decision is up to the manager’s judgment” (Keen and Morgan 1978). Thus, it can be said that Decision support system supports difficult decision-making and boosts its effectiveness.

To further understand how decision support systems works, we refer to (Bidgoli. H 1989) “that suggested there are certain requirements for a system which must be embrace a true definition of decision support system.” The requirements are:

- DSS requires hardware;
- DSS requires software;
- DSS requires human elements (designers and end-users);
- It is designed to help in decision making;

3.5.2 Characteristics

Decision support system is a business element that provides managers, and other workers with data for making informed decisions. The requirements needed for a system to be considered as a decision support system has been discussed.

Here we want to look at the characteristics or the criteria that makes a system to be considered as a decision support system

Most companies define or create their own characteristics based on the type of system that is needed. Some companies might choose to design their own mix of automatic and manual systems. While others may just decide to purchase and implement a ready-made third party package. Most third party packages are customizable which allows company to change various forms of the system. In short, most companies choose systems that have characteristics that match their needs.

But there are certain characteristics that any decision support system should have. They are as follows:

- Facilitation to ensure the system provides the accurate information and data for any decision.
- Interaction is another important characteristic of decision support system. The ability for multiple individuals to access the software and input data for decision making process.
- DSS should be task oriented, i.e. the support system should be capable of handling multiple activities in a time.

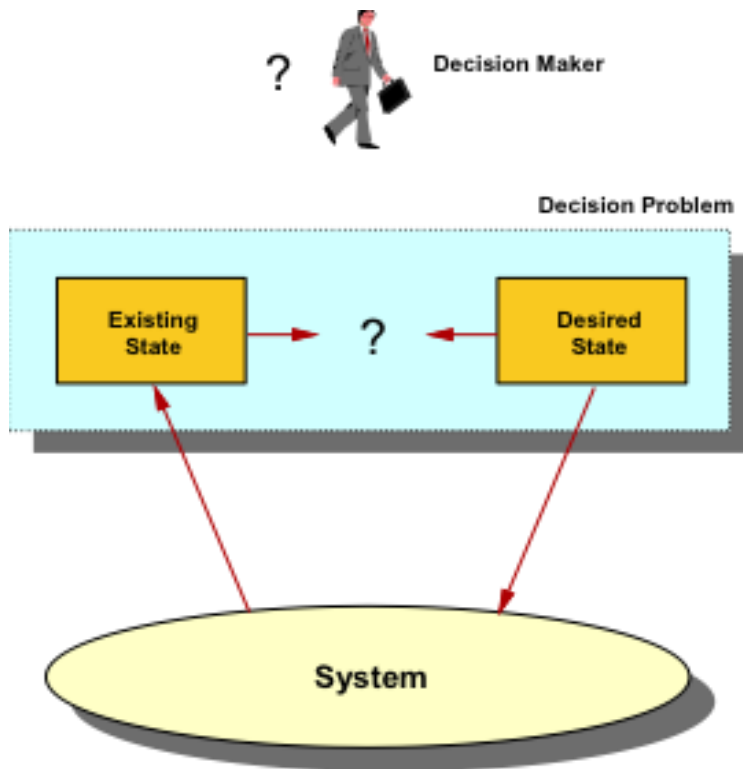


Figure 3.3 Components of decision support system

3.5.3 Who needs DSS?

Decision support system has changed the way executives make their decisions since its inception in the late 1970s. Knowledge workers use it frequently that it is possible to consider using DSS in any knowledge domain. In fact, they are so rampant in the modern society that people don't understand they are using decision support system.

Writing on a spreadsheet or using a search engine, that provides huge amount of data either in text form or, images and video are simple examples of decision support system that we use every day.

Healthcare organizations use a decision support system called clinical DSS. It is believed that if used correctly, clinical decision support system has the power to change the way we of teaching and practicing medicine.

Real estate investment companies frequently use decision support systems for running their daily businesses. By using DSS, real estate agencies are able to access information regarding each property across the enterprise. Information and data can be processed not just for day-to-day running but for future planning also.

Engineers also use DSS for integrating weather conditions and optimizing reservoir operations. Business manager use it mainly to analyze budget sheet, forecast and sales figure.

Below some of the advantages and disadvantages of DSS are given:

Advantages:

- Allows for faster decision making
- Improves collaboration and communication within groups
- Increases performance and effectiveness of the user
- Provide multiple solutions to a problem

Disadvantages:

- A lot of power given to computer

Chapter 4

4.1 Problem Definition

In this study, a DSS is created. The DSS provides the user with option of solving an assembly line balancing problem. The user is required to enter any random data into the DSS, the system solve the problem using the three mentioned heuristic methods and provide the best solution among them. The system also gives detail information about each heuristic.

The assembly line balancing problem is considered to be deterministic single model line, i.e., all input values are given and are expected to be known with certainty. Single product is continuously manufactured along the line. Cycle time, precedence relation of the tasks, task times, production rate, and available work hours all together define the problem data.

Note: The assumptions and mathematical formulation are developed depending on Baybar's SALPB mixed integer programming formulations (Baybar 1986)

The decision support system will provide the user with the results. The objective is minimization of the number of workstation with the best efficiency and balance delay.

4.2 Mathematical Formulation

Here we present our assumptions, notations and formulations of our problem

Our assumptions are listed below:

- A single product is assembled on the assembly line
- The task times vary depending on the value given by the user

- The cycle time of all workstations is given and cannot be changed
- The precedence relations between task are given and not subject to change
- Assembly tasks cannot be split among two or more stations
- A task can be performed at any workstation in the assembly line, as long as precedence relations and requirements are satisfied.

The notation and parameters used in the mathematical formulation are given below:

Indices:

i = task index

j = workstation index

Parameters:

n = number of task

t_i = processing time of task i

C = cycle time

m = number of workstation

S_i = set of immediate successors of task i

M is considered as a very large

Decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if task } i \text{ is assigned to station } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$m_j = \begin{cases} 1 & \text{if station } j \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$$

The mixed integer programming formulation of the problem is as follow:

$$\bullet \text{ Minimize } \sum_j^m m_j \quad (1)$$

Subject to

$$\bullet \sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, n \quad (2)$$

$$\bullet \sum_{j=1}^m j x_{ij} - \sum_{j=1}^m j x_{kj} \leq 0 \quad i = 1, \dots, n \text{ and } k \in S_i \quad (3)$$

- $\sum_{i=1}^n t_i x_{ij} \leq C \quad j = 1, \dots, m \quad (4)$

- $\sum_{i=1}^n x_{ij} - Mm_j \leq 0 \quad j = 1, \dots, m \quad (5)$

- $x_{ij} \in \{0, 1\} \quad i = 1, \dots, n \text{ and } j = 1, \dots, m \quad (6)$

The objective function (1) represents the number of used stations m .

Constraint (2) ensures that all tasks are assigned only once.

Constraint (3) represents the precedence relations between task x_1 and x_2 . Such that if x_1 is the immediate predecessor of x_2 , then task x_1 cannot be assigned to another workstation apart from task x_2 's workstation.

Constraint (4) is the cycle time or capacity constraints and makes sure the total summation of all work content of each station is not greater than the given C of station.

Constraint (5) ensures that station j becomes available if a task is assigned to it.

Constraint (6) ensures the indivisibility among each task. Each task can either be assigned to a workstation as a whole or not.

4.3 An Example Problem

We tested our decision support system with a SALBP example. The example consists of 11 tasks. The expected minimum number of workstations is calculated to be at least 4, which are all capable of containing the tasks. The production rate and available working hours are 7 hours per day and 500 units per day respectively. The cycle time is calculated as 50.4 seconds units. The table below illustrates the task times and precedence requirements

Table 4.1. Task times and precedence requirements

Task Number	Job	Time	Precedence
1	A	45	–
2	B	11	A
3	C		B
4	D	50	–
5	E	15	D
6	F	12	C
7	G	12	C
8	H	12	E
9	I	12	E
10	J	8	F, G, H, I
11	K	9	J

The following figure illustrates the precedence diagram.

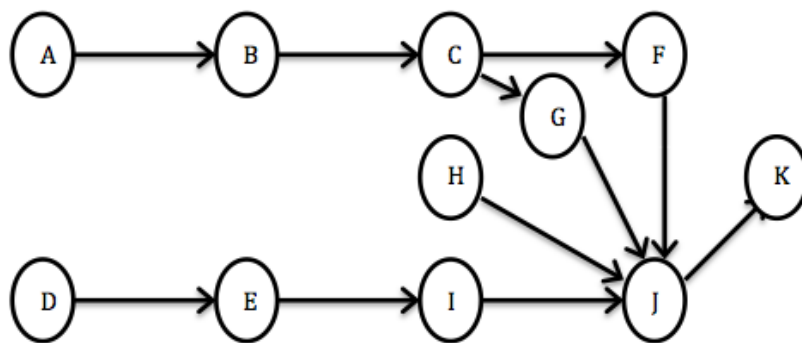


Figure 4.1. The precedence diagram of our problem

When we applied the above problem to our decision support system, three efficient solutions were obtained as depicted by the following configurations:

Solution 1:

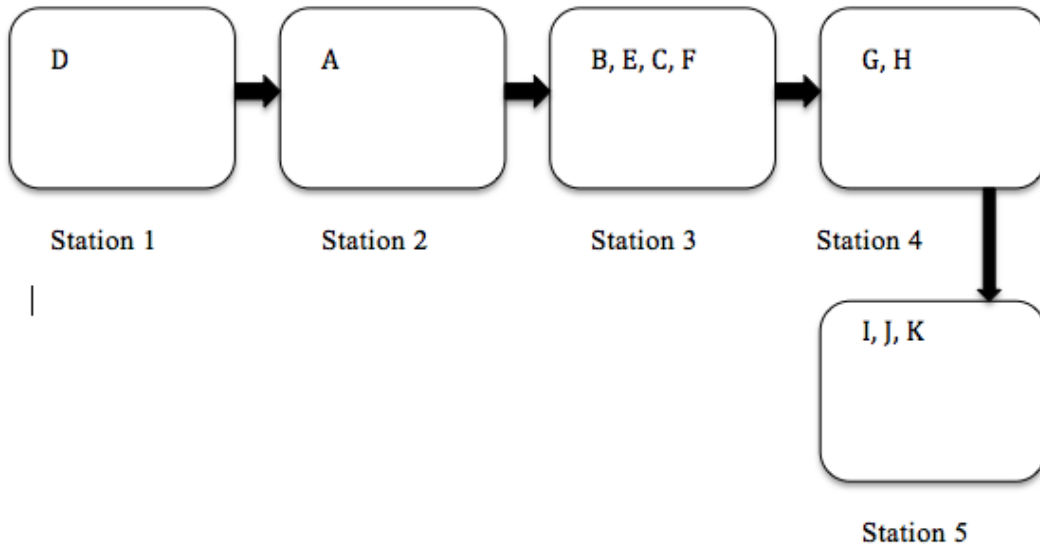


Figure 4.2 the first efficient solution using Helgeson and Bernie approach

Number of workstations = 5

Line efficiency = 77.38%

Balance delay = 22.62%

Solution 2:

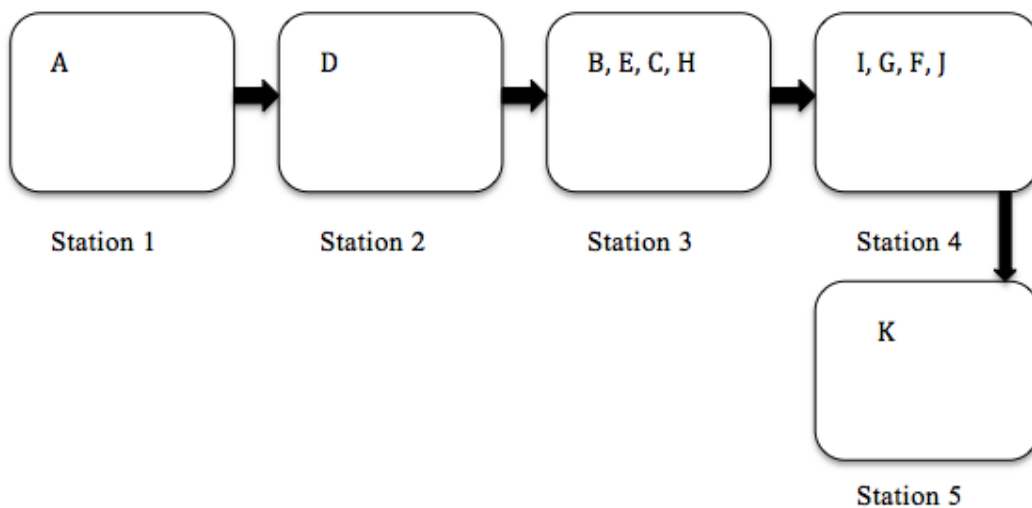


Figure 4.3 the second efficient solution using Kilbridge and Wester's approach

Number of work stations = 5

Line efficiency = 77.38%

Balance delay = 22.62%

Solution 3:

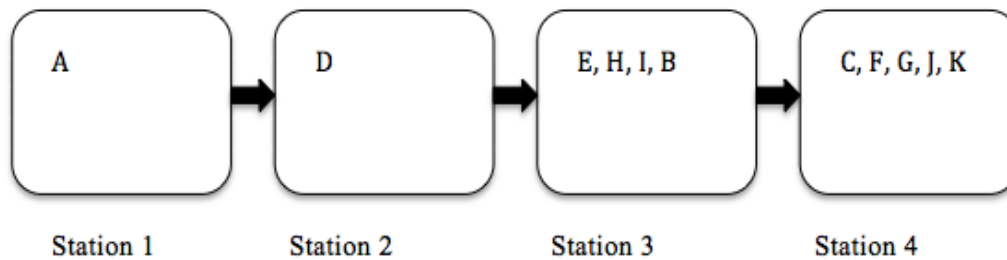


Figure 4.4 the third efficient solution using Largest candidate rule

Number of workstations = 4

Line efficiency = 97.62%

Balance delay = 3.48%

As we can see, the decision support system solved the problem using three heuristics and gave us three alternative results. Solutions 1 and 2 have the same number of workstations. That result meant we also obtained similar line efficiency and balance delay from those two solutions because the calculation for line efficiency and balance delay depends on the number of workstations used and the cycle time.

The third solution provided us with the best result for our problem. Not only did we get the minimum number of workstation (4), which is our main objective but we also obtained better line efficiency and balance delay for our assembly line balancing problem.

4.4 Solution Approach

4.4.1 Excel Implementation

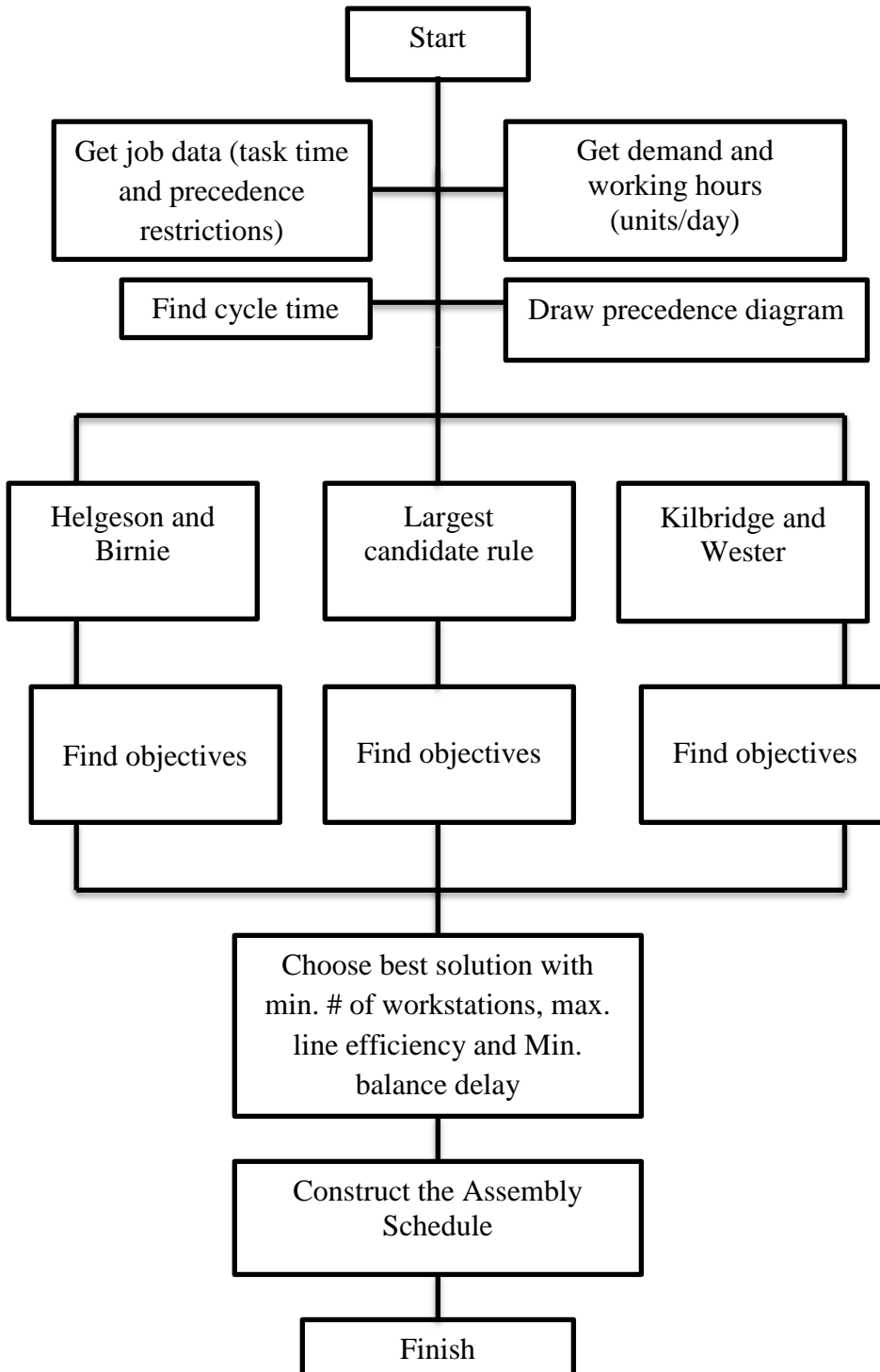


Figure 4.4 Components of our DSS

As illustrated in figure above, the flowchart shows how the problem is solved. The flow chart displays exactly how the decision support system works. It is designed in a way that involves the general steps of solving SALBP using heuristics methods. The first step is gathering all your data i.e. task times, production rate, working hours, precedence relations and so on. This is followed by the second step, where the data provided should be implemented in calculating cycle time and precedence diagram. Once the cycle time is calculated, the system can start applying the formulas implemented to solve the question or problem using our three heuristic approaches. All three approaches have similar objectives and will provide answer of similar criteria. By applying the IF function the DSS is able to choose which answer from each solution criteria is the best considering its objective. Based on the results obtained, the DSS construct the assembly line that provides the best alternative in all the objectives.

The figures below are screenshot of the DSS.

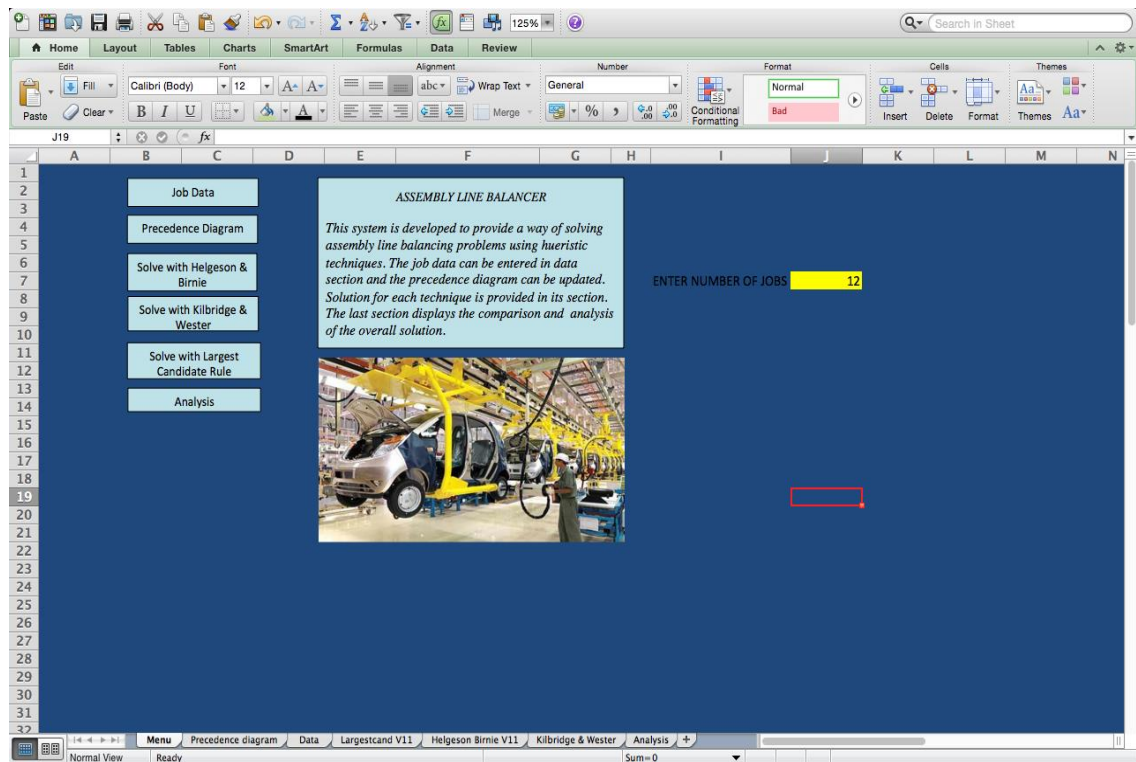


Figure 4.5 Main page of our decision support system

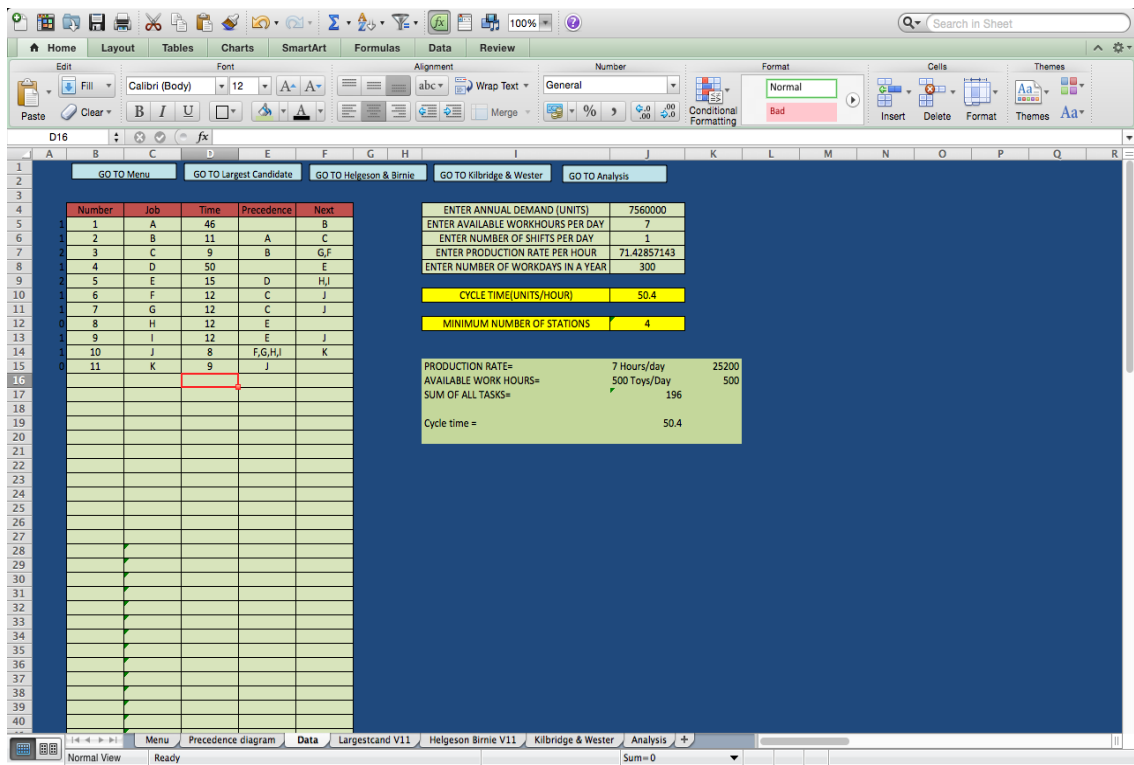


Figure 4.6 Data entry page of our decision support system

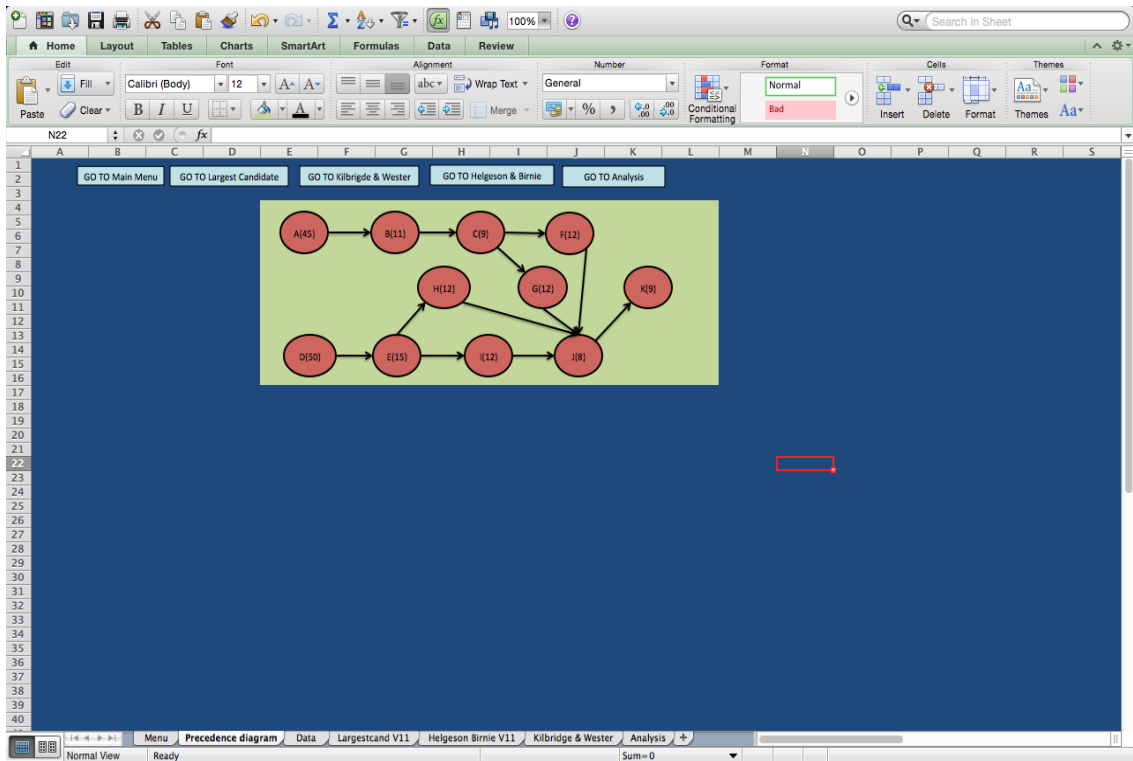


Figure 4.7 precedence diagram of our decision support system

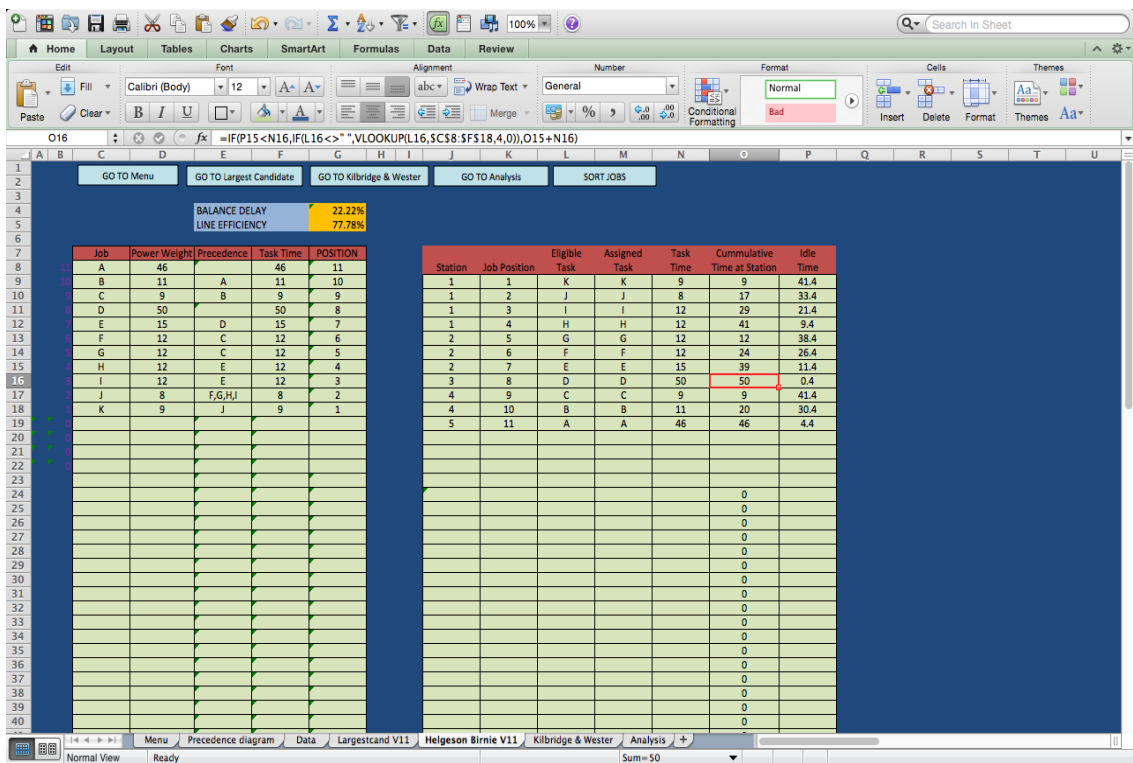


Figure 4.7 first solution to our problem using Helgeson and Birnie method (RPW)

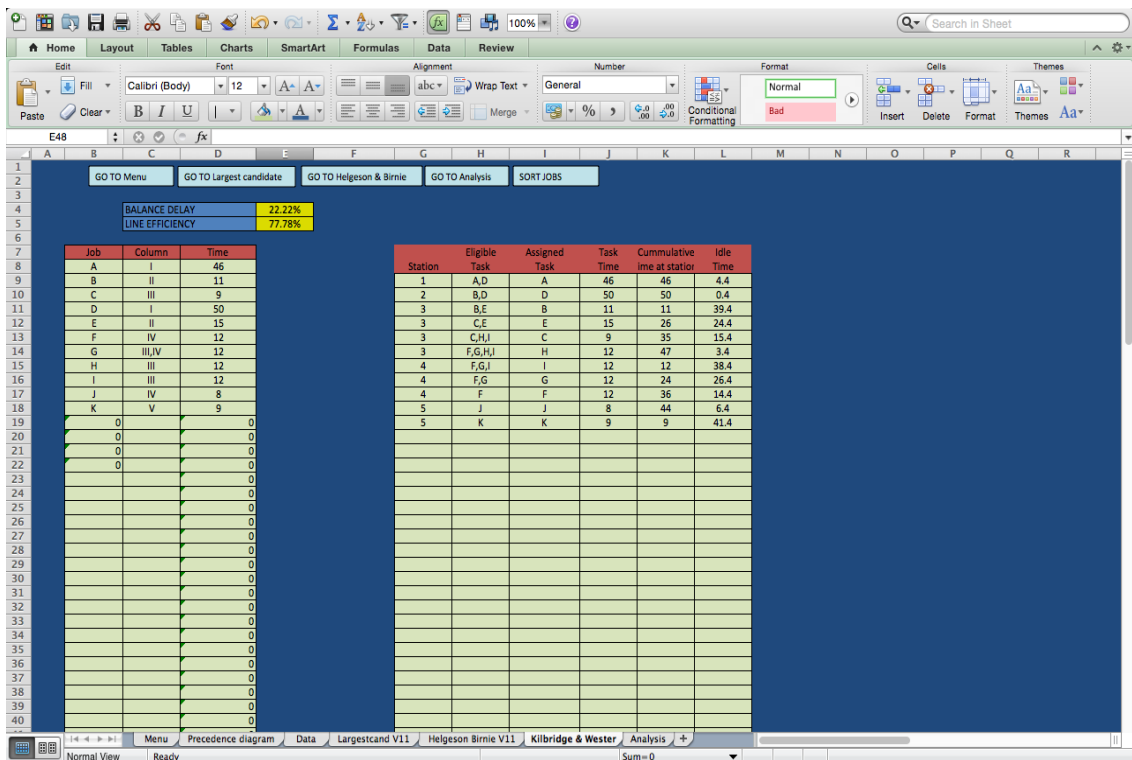


Figure 4.8 second solution to our problem using Kilbridge and Wester’s method

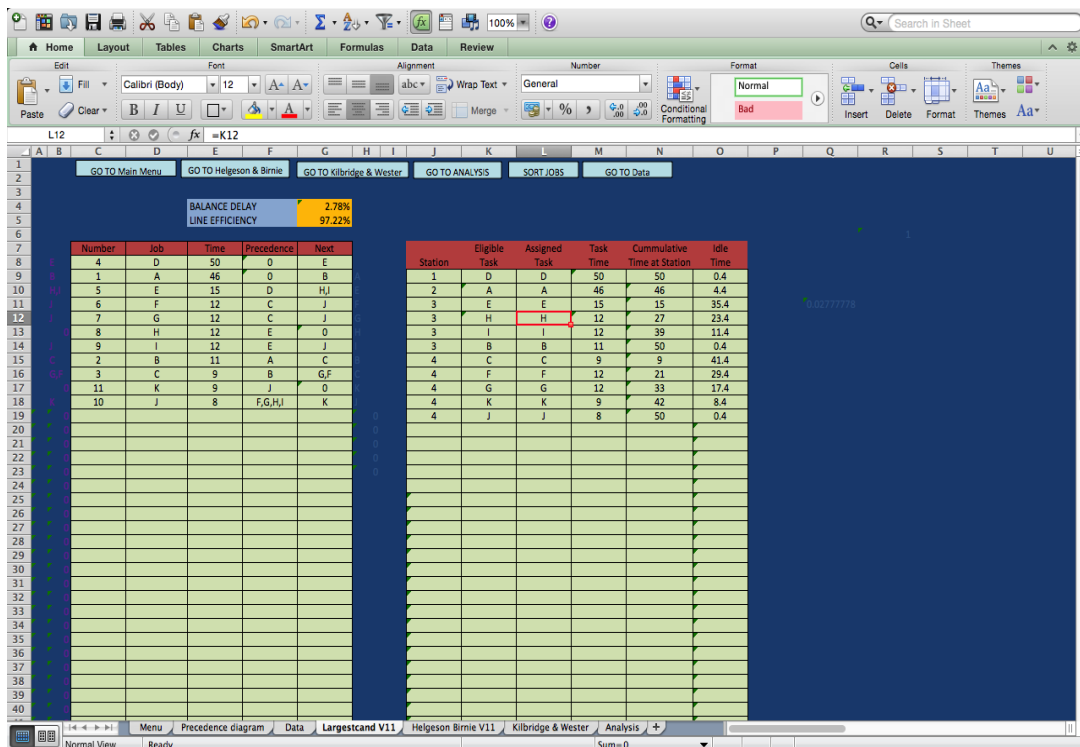


Figure 4.9 third solution to our problem using Largest candidate rule

The main menu in the excel spreadsheet presents the introduction page of the decision support system. It is arranged in a way that is easy for the user to access the other spreadsheets in the program. It also explains in details the purpose of the decision support system, how to use it and how it works.

Five macros are created as can be seen in the main menu and each performs different activity when pressed. The first is the job data. The job data menu when pressed takes the user to the initial spreadsheet that asks the user to enter random values for the assembly line problem. Also, in the job data spreadsheet more details about the problem such as: cycle time, minimum number of stations, available working hours etc. are displayed.

The next spreadsheet displays the precedence diagram. The diagram displayed consists of 11 tasks and it represents the problem from our example. We couldn't implement the exact function to help us organize each constraint requirement entered by the user.

As result we displayed the initial diagram to show how it will work when the correct function is applied.

The next three spreadsheets are named RPW, Kilbridge & Wester, and Largest Candidate respectively. Each spreadsheet provides solution to the data entered by the user and displays it using the heuristic approach named. A lot of functions have been used in these sections. To make the decision support system work correctly it is accepted that no data entry is done on any of these spreadsheets. All data and values are entered in the data section, only the solutions are displayed.

Also in each of the three solution spreadsheets the objectives, i.e., minimum number of stations, minimum balance delay, and maximum line efficiency are calculated and displayed accordingly.

As seen in figure (4.7- 4.9) each solution has tables that represent how the solution is achieved. Each solution has two tables: data table that displays the values entered by the user and the solution table that uses some functions mostly IF and V-LOOKUP to solve the problem.

The final spreadsheet displays the analysis between the three heuristic solutions. There are two tables that explain the analysis process. Table 1 illustrates the result of all heuristic methods with respect to the objectives of the problem. While table 2, using IF function we are able to display which solution is the best among each criteria of the objectives.

4.4.1.1 Formulas and Functions used in Excel

1. IF function
2. V-LOOKUP
3. SUM
4. SUBTRACT
5. MULTIPLY

Chapter 5

5.1 Case Study

In this chapter we discuss the case study to our problem. In most cases we should consider a real life problem and provide solution or apply our work and solve the problem presented but here we were not able to present such scenario. The reason for not presenting such a problem is because most real life assembly line balancing problems have zoning and positional constraints that may result in significant changes to the flow line design. Unfortunately the heuristic methods developed in our decision support system are not equipped to fully deal with these constraints efficiently.

As a result, since our aim is to develop a decision support system that solves a simple assembly line balancing problem type-I using three heuristic approaches, which we have done successfully but with our problem. We have decided to find another similar well-known problem and solve it using our decision support system.

5.2 The problem in study

The problem presented requires a new electrical device to be assembled on a production flow. The details about the task times and precedence requirements are given acquired from the industrial engineering departments based on previous similar time standards they have developed are given in table 5.1. The production rate is given as 120,000 units/year and working hours 50 weeks/year at 40 hours/week.

Table 5.1 Job date for our case study

Job No.	Job description	Time	Precedence
1	Place frame on work holder and clamp	0.2	–
2	Assemble plug, grommet to power cord	0.4	–
3	Assembly brackets to frame	0.7	1
4	Wire power cord to monitor	0.1	1, 2
5	Wire power cord to switch	0.3	2
6	Assemble mechanism plate to bracket	0.11	3
7	Assemble blade to bracket	0.32	3
8	Assemble motor to brackets	0.6	3, 4
9	Align blade and attach to motor	0.27	6, 7, 8
10	Assemble switch to motor bracket	0.38	5, 8
11	Attach cover, inspect, and test	0.5	9, 10
12	Place in tote pan and package	0.12	11

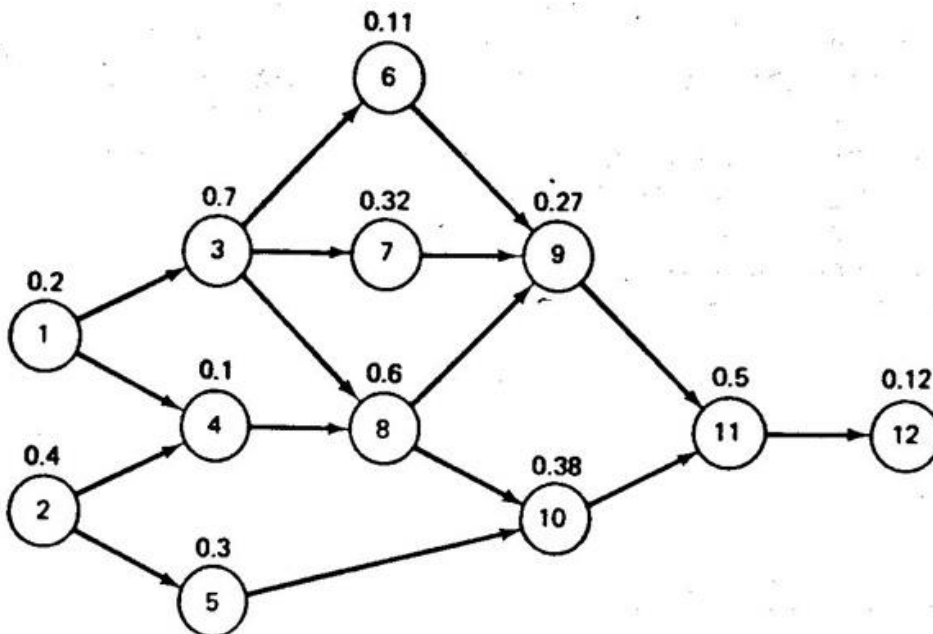


Figure 5.1 Precedence diagram of our case study.

5.3 Solving the Problem

To solve this problem, first we have to understand that our decision support system is appropriate for only less complicated line balancing problems. More complicated problems require much sophisticated techniques. Like mentioned several times, the methods applied in this study are heuristic methods. In other words, they are most probably to provide acceptable feasible solutions close to the optimal solutions but less likely to give the true optimum.

Before starting to solve the problem using our methods, there are some certain calculations required.

Firstly, we calculate the total work content i.e., summation of all the task times that are required on the line. The total work content gives us a value of 4 mints.

The cycle time is then calculated by considering the production rate as 120,000/year at 50 weeks/year and 40 hours/per. That gives us an output of 60 units/hour or 1 unit/mint from the line. The minimum number of stations is calculated through dividing the total work content (4 mints.) and the cycle time (1 mint.) and that's gives us a value of 4 minimum workstations.

Now that the initial stage of our problem solution has been done we can proceed to the solution phase. (NOTE: only the results are written in this section, the parameters and table are provided in the Appendices section).

The steps applied in the next 3 sections are all given in section 3.4.

5.3.1 Largest candidate rule solution

By solving the problem using the steps given in (section 3.4.1) the following results are achieved:

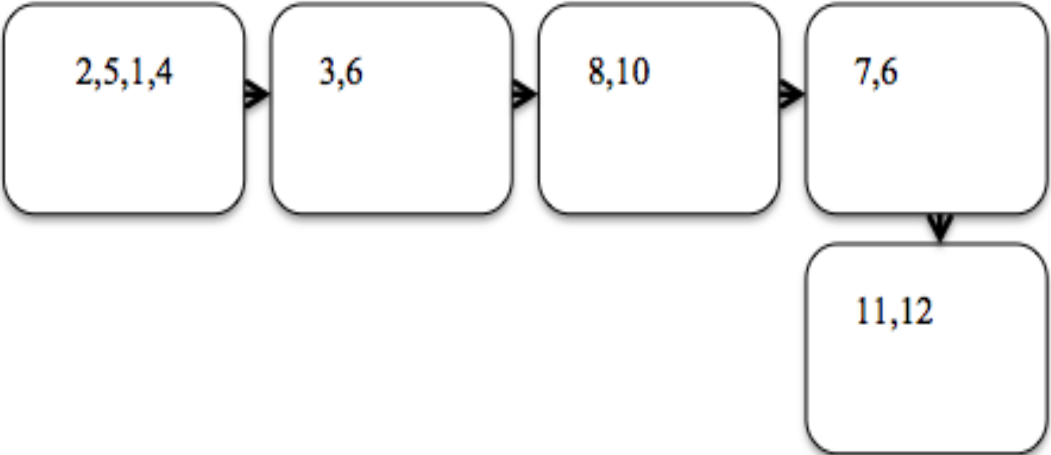


Figure 5.2 the first efficient solution using Largest candidate rule.

Number of workstations = 5

Line efficiency = 80%

Balance delay = 20%

5.3.2 Kilbridge and Wester's solution

In this method, the precedence diagram requires a lot of attention because assignment of tasks to workstation is done depending on their position on the diagram. Unfortunately our decision support system doesn't provide such services of updating each precedence constraint the user enter. It concentrates more on solving the problem based on the original implemented precedence diagram.

As a result, the initial phase of grouping the task based on columns is done separately then later implemented into the system. Refer to table A.3 for details.

Solving the problem with Kilbridge and Wester Method, the following results are achieved:

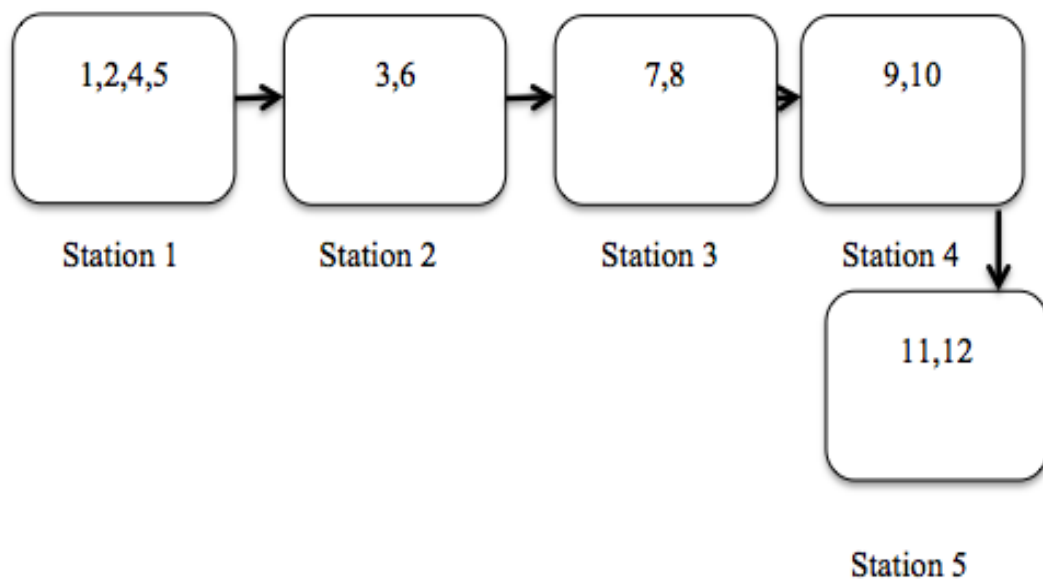


Figure 5.2 the second efficient solution using Kilbridge and Wester's Method.

Number of workstation = 5

Line efficiency = 80%

Balance delay = 20%

5.3.3 Ranked Positional Weight

The Ranked positional weight method provides us with 2 results, details will be explained in section 5.3.

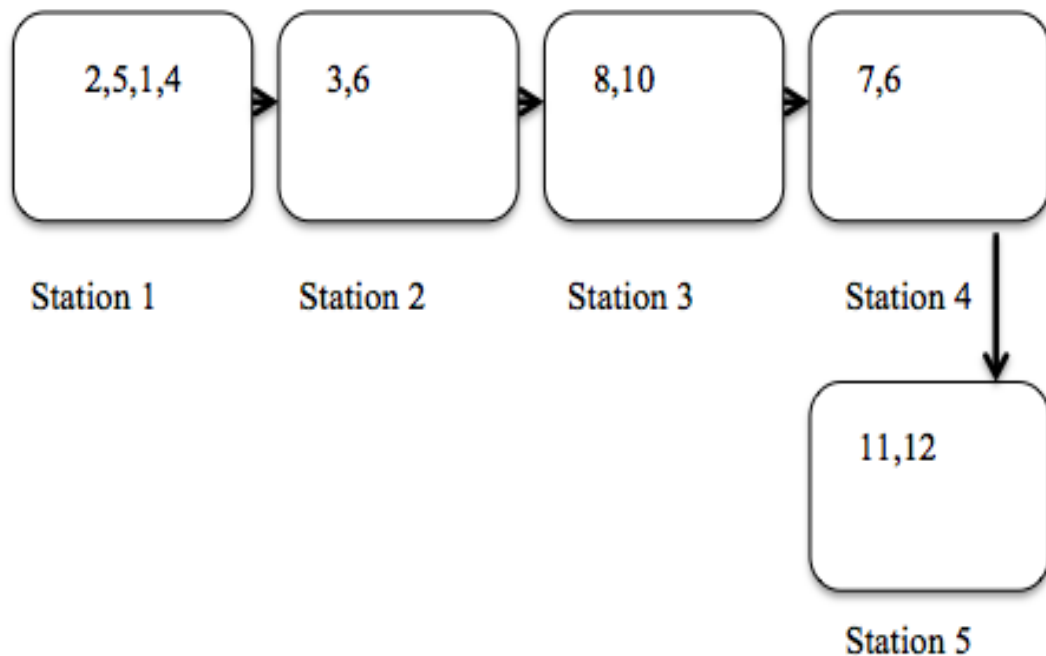


Figure 5.2 the third efficient solution using Helgeson and Birnie method.

Initial result:

Number of workstations = 5

Line efficiency = 80%

Balance delay = 20%

Updated result:

Number of workstations = 5

Line efficiency = 87%

Balance delay = 13%

5.4 Result Explanation

After the data have been entered into the DSS. The solutions are displayed in the previous section (5.3). Here explanation will be made in details to fully understand how each heuristic approach solves the problem.

For largest candidate rule, the jobs in our presented case study are listed in table A.1 in descending order, which is the initial step in this method. This step helps in determining feasibility of jobs when assigning workstations. The system starts by assigning jobs on top of the list with respect to precedence requirement given in figure 5.1. The job with largest task time is job 3 but it's not feasible because its predecessor job 1 hasn't been assigned yet. The first feasible job encountered is job 2, which is then assigned to station 1. Continuing the same procedure for the remaining task provided us with the solution that is given in table A.2.

The cycle time is calculated as 1.0 in section 5.3. Five workstations are used which gives us 80% line efficiency and 20% balance delay. The results are listed in section 5.3.1.

This solution further provide more proof that largest candidate rule is a method for less complicated assembly line problems.

Kilbride and Wester's method divided the presented case study into column based on their position on the precedence diagram. As a result, the total summation task time of each column is important, which is also displayed in table A.3 along with the columns

each task belongs to. The sum of all tasks in column I is less than the cycle time, which means all the jobs in column I can be done in station 1. Next the system looks for which job can be added to station without exceeding the cycle time. The system continues performing this operation until all task are assigned. Interestingly in this method much attention is not given to precedence requirements because ordering jobs in column takes care of any requirements.

Table A.4 shows the order in which each task is assigned along with the sum of all task times in stations. This method also provides us with similar result due to that fact that five workstations are used and similar cycle time is considered in the problem. Line efficiency is 80% and balance delay is 20%. However, it's important to notice that the jobs are not sequence in the same order. The system might provide us with similar result but the sequence changes. Most importantly, this method is considered to be able to deal with rather complicated problems unlike the largest candidate rule.

The final method ranked positional weight starts giving each job a power weight. The system calculates the weight of each job as explained in section 3.4.3 and displays it along with task times and precedence requirements in table A.5. The system then solves the problem by assigning tasks on top of list going downward to the bottom with respect to precedence requirements. Table 5.7 shows the order in which tasks are assigned to work stations.

Interestingly, 2 results are obtained from this method. Initial result gave us similar solution to our previous approaches i.e., 5 number of station, 80% line efficiency and 20% balance delay.

After considering table A.6 under the sum of task time's section, we realized that the maximum cycle time needed for all the station is 0.92 seconds in station 3. This means the line could be operated with a cycle time of 0.92 and still works perfectly. As a result the line efficiency and balance delay are recalculated and provide us with our second solution i.e., 5 number of stations, 87% line efficiency and 13% balance delay.

5.5 Final Analysis

Considering the case study tested on our decision support system, we can successfully say that ranked positional weight provide the best line efficiency and minimum balance delay. This means two of our objectives are achieved using this method. However, it is important to note that a different cycle time is considered in the calculation process than the one given by the problem.

If same calculations are to be used for largest candidate rule and Kilbridge and Wester's method, possibly we might get similar result. That's why when solving very large problems that involve thousands of tasks, using such methods is not very advisable. As a result, a pretty good number of computer programs are available when dealing with such large and sophisticated problems.

As for our case study, we can say our test is successful. Our system worked with a different example and provided us with similar results as expected.

The problem presented in this study is taken from the book: *S. Anil Kumar & N. Suresh "production and operation management Second Edition"* Retrieved from: <https://rekadayaupaya.wordpress.com/2013/05/13/6-4-the-line-balancing-problem/>.

Chapter 6

6.1 Conclusions and Further Research

In this thesis, we consider a SLBP and develop a DSS that solves any simpler and less sophisticated type of SALBP. The system uses three heuristic methods; the largest candidate rule, Kilbride and Wester's method, and Ranked positional weight method (Helgeson and Birnie).

The system developed is flexible enough to solve assembly line balancing problems with multiple objectives. The objectives considered in our study are; minimizing total number of workstation used; maximize the assembly line efficiency, and minimization of the total balance delay. The user is given the option to enter any random value to an already implemented problem into the system. The system will use the three methods mentioned then consider the objectives and provide the solution.

Our system is only capable of solving simply assembly line balancing problems. It is not complicated enough to deal with real life problems, where multiply different objectives and zoning constraints are present. Whenever a problem consists of more than one conflicting constraints, it gets more complicated and difficult to solve. Due to the fact that real life problems are not considered in this study and the lack of literature on decision support system for assembly line balancing problems, much literature review about the topic is not provided in this study. In my opinion, the reason for not having much literature on DSS that solves large scale or real life ALPB, many functions are required and enough information regarding the constraints and precedence relations makes it more complicated to start with. The variety of task times in real life problem is another factor

However, enough literature review and previous studies about heuristic methods have been discussed.

Factors such as cycle time, task times and precedence relations are very important in the solution of SLBPs. In the three heuristics approaches applied in this paper, all consider these factors in determining the objective. When minimizing the number of workstation, the sum of all tasks in that station should not be greater than given C. If such scenario occurs then a new station is needed.

Same can be said to when determining the other 2 objective; maximization of line efficiency and minimization of balance delay, where the total number of stations used and the C again are used in calculating the answer.

Microsoft Excel spreadsheet was used in developing the DSS. Functions such as IF and VLOOKUP were applied in developing the system. The functions solve the problems from the data entered by the user. Macros are created to make it easier to navigate the system created, making it easier to go from one spreadsheet to another. Also, a different spreadsheet is created to display and compare the results and also suggest the best solution among them.

To solve the SALBP, the system considers five constraints in evaluating the solutions. Each constraint plays a vital role in determining how the objective function is achieved, regarding the three heuristic approaches. Largest candidate rule considers the precedence constraint even when assigning the task to work station based on descending order of their task times. Kilbridge and Wester's method considers the cycle time constraint when assigning tasks to work station. Several tasks can be on the same column but may not be performed on the same station if the sum of tasks will exceed that cycle time. Same can be said of ranked positional weight where a task with higher positional weight might not be assigned to a station if it will be larger than the calculate C of the station.

Two simply assembly line examples with deterministic task times are tested on the system created. The results show that the system is capable of solving a less complicated problem with reasonable number of tasks.

For each test of the assembly line balancing problems, the results are analyzed separately. The results can be summarized as follows:

The initial example is acquired through trial and error. It is a SALBP-1 that has 11 tasks. The remaining parameters were all calculated using the data given. The solution gave positive results. Both Ranked positional weight and Kilbride and Wester's method gave similar results.

But largest candidate rule provided the best solution, considering the objectives considered.

The system is then finally tested on the case problem presented. Due to the fact the system cannot handle real life problem it is tested on SALBP-I from the literature. 12 tasks are considered this time around. Almost similar results are acquired. Ranked positional weight gave us the best solution with regards to the objective mentioned in the studies. The remaining two approaches have similar results to each other.

The whole study shows that a DSS can be developed that solves any type of assembly line problem whether single, multi, or mixed models. But it also shows that a lot of functions and objective functions have to be considered. In our study the results shows that the relationship between having an efficient assembly line and good balance delay depends on the number of workstation used in the assembly lines.

There are several number of further research in the area studied. The most notable ones are listed below:

- [1] There is the need for more heuristic approaches that should deal large scale problems.
- [2] More literature review is needed about decision support systems on assembly line balancing problems.

References

- [1] A. Dwivedi (2012) “An Analysis and Development of Software for Assembly Line Balancing Problem of Manufacturing Industry,” VSRD International Journal of Mechanical, Automobile & production Engineering, **Vol. 2** (2), 74-87.
- [2] Baybars, I., (1986). “A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem”. Journal of Management Science **32** (8), 909–932.
- [3] Bidgoli, H. (1989). "Decision support systems: Principles and practice." St Paul: West Publishing Company. Retrieved from: <http://www2.aku.edu.tr/~icaga/kitaplar/decision-making-support-systems.pdf>.
- [4] C. Becker and A. Scholl (2006). "A survey on problems and methods in generalized assembly line balancing." European Journal of Operational Research **168**(3): 694-715.
- [5] D. Mahto, A Kumar (2012), “An Empirical Investigation of Assembly Line Balancing Techniques and Optimized Implementation Approach for Efficiency Improvements,” Global Journal of Researches in Engineering Mechanical and Mechanics Engineering.
- [6] Decision support system. (2015, June 7). In Wikipedia, The Free Encyclopedia. Retrieved 14:40, July 7, 2015.

- [7] Gong, J., Liu, W., Liu, X. and Cui, T., (2010) "A Simulation-Based Performance Comparison between Multi-Model Assembly Lines and Assembly Cells in a Just-In-Time Environment". Retrieved from: <http://www.researchgate.net/publication/251932635>.
- [8] Gökçen, H., et al. (2006). "Balancing of parallel assembly lines." International Journal of Production Economics **103**(2): 600-609.
- [9] Helgeson, W. P., and Birnie, D. P., (1961). "Assembly Line Balancing Using the Ranked Positional Weight Technique." Journal of Industrial Engineering, **12** (6), 394-398.
- [10] Hoffmann, T. R., (1963). "Assembly Line Balancing with a Precedence Matrix", Journal of Management Science, **9** (4), 551-562.
- [11] Keen, P. G. W. and M. S. Scott Morton (1978). "Decision support systems : an organizational perspective." Reading, Mass., Addison-Wesley Pub. Co. Retrieved from: <http://diuf.unifr.ch/ds/courses/dss2002/pdf/DSS.pdf>.
- [12] Kilbridge, M. and Wester, L., (1961). "The Balance Delay Problem," Journal of Management Science, **8** (1), 69-84.
- [13] M.P. Groover, (2008), "Automation, Production Systems, and Computer-Integrated Manufacturing Third Edition," Pearson Education, Inc., Upper Saddle River, New Jersey, Retrieved from: <https://www.hellolibrary.org/automation-production-systems-and-computer-integrated-manufacturing-solutions.pdf>.
- [14] N. Boysen, M. Fieldner, and A. Scholl (2007). "Assembly line balancing : which model to use? " European Journal of Operational Research **183**(2): 674-693.

- [15] S. Anil Kumar & N. Suresh “production and operation management Second Edition,” Retrieved from: <https://rekadayaupaya.wordpress.com/2013/05/13/6-4-the-line-balancing-problem/>.
- [16] S. Mat, Z. Muhamad, and C. Law (2002), “Work Improvement at a Car Manufacturing Company,” Jurnal Teknologi, (36A), 55-63.
- [17] Scholl, A. and Becker, C., (2003). "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing", European Journal of Operational Research.
- [18] Wild, R. (1972) “The design and operation of production flow-line systems, John Wiley & Sons, London, New York,” Journal of Mass Production Management.

Appendices

In this appendix, we give the detailed information about the case study problem taken from the literature. The information about which task is assigned to which stations is provided. Also, other parameters regarding the precedence relationship, task times, immediate predecessor, columns, and station capacity are all given in the tables below.

Table A.1 Jobs arranged in descending order according to their task times.

Job Number	Time	Immediate Predecessors
3	0.7	1
8	0.6	3, 4
11	0.5	9, 10
2	0.4	–
10	0.38	5, 8
7	0.32	3
5	0.3	2
9	0.27	6, 7, 8
1	0.2	–
12	0.12	11
6	0.11	3
4	0.1	1, 2

Table A.2 Jobs assigned to stations using the largest candidate rule.

Station	Job	Time	Sum of task Times in station
	2	0.4	
	5	0.3	
	1	0.2	
1	4	0.1	1.00
	3	0.7	
2	6	0.11	0.81
	8	0.6	
3	10	0.38	0.98
	7	0.32	
4	9	0.27	0.59
	11	0.5	
5	12	0.11	0.62

Table A.3 Job assigned according to columns from figure 5.1.

Job Number	Column	Time	Sum of Column Task times
1	I	0.2	
2	I	0.4	0.6
3	II	0.7	
4	II	0.1	
5	II, III	0.3	1.1
6	III	0.11	
7	III	0.32	
8	III	0.6	1.03
9	IV	0.27	
10	IV	0.38	0.65
11	V	0.5	
12	VI	0.12	0.12

Table A.4 Jobs assigned to stations using Kilbridge and wester's method.

Station	Job	Time	Sum of task Times in station
1	1	0.2	0.2
1	2	0.4	0.6
1	4	0.1	0.7
1	5	0.3	1.00
2	3	0.7	0.7
2	6	0.11	0.81
3	7	0.32	0.32
3	8	0.6	0.92
4	9	0.27	0.27
4	10	0.38	0.65
5	11	0.5	0.5
5	12	0.12	0.62

Table A.5 Jobs arranged in Order of Power Weight value in RPW method.

Job	Power Weight	Time	Predecessors
1	3.30	0.2	–
3	3.00	0.7	1
2	2,67	0.4	–
4	1.97	0.1	1, 2
8	1.87	0.6	3, 4
5	1.30	0.3	2
7	1.21	0.32	3
6	1.00	0.11	3
10	1.00	0.38	5, 8
9	0.89	0.27	6, 7, 8
11	0.62	0.5	9, 10
12	0.12	0.12	11

Table A.6 Jobs assigned to stations using Ranked Positional Weight.

Station	Job	Time	Sum of task Times at station
1	1	0.2	0.9
1	3	0.7	
2	2	0.4	0.91
2	4	0.1	
2	5	0.3	
2	6	0.11	
3	8	0.6	0.92
3	7	0.32	
4	10	0.38	0.65
4	9	0.27	
5	11	0.5	0.62
5	12	0.12	