NOT TO BE TAKEN

# A SURVEY
# of
# ASSEMBLY LINE BALANCING ALGORITHMS
# and
# A PROPOSED ALGORITHM

by

Alpaslan KARAYALÇIN

B.Sc. in I.E.

St.MARY'S UNIVERSITY

OF SAN ANTONIO, 1978

Bogazici University Library

39001100316374

14

Submitted to the Faculty of the School of Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering

BOĞAZİÇİ UNIVERSITY

1981

This thesis Submitted by Alpaslan KARAYALÇIN, to the Industrial Engineering Department of BOĞAZİÇİ University in partial fulfillment of the requirements for Degree of Master of Science is approved:

Yük.Müh.Çetin EVRANUZ
(Thesis Advisor)

Doç.Dr.Selçuk ERDEN

Dr.Ceyhan UYAR

# ACKNOWLEDGMENTS

This study was conducted under the Supervision of Çetin EVRANUZ to whom I wish to express my sincere thanks for his valuable and close guidance. I should also give my thanks to the Faculty of Industrial Engineering Department for their contribution to my graduate study in Industrial Engineering.

# A B S T R A C T

In this thesis, Assembly Line Balancing (ALB) Process is studied in detail including the problem and available Algorithms. Basic concepts and measure of effectiveness of the line-balancing process were defined. The exact model was presented by using Mixed-Integer Programming.

A comprehensive literature survey is presented with the addition of proposed algorithm which is developed in this thesis. Among the algorithms covered in this literature survey; i) Kilbridge and Wester Algorithm, ii) Ranked Positional Weight Algorithm, iii) COMSOAL and, iv) Largest Candidate Rule. Following these two of the commonly used computerized techniques of HOFFMANN and ARCUS are also discussed. Then Stochastic Line Balancing is studied and described by using BRENNECKE's Algorithm. Finally Cost-Oriented Approach and a Preference Order Dynamic Programming for Stochastic ALB are discussed. Flow charts for three of the algorithms and computer programs in FORTRAN IV are written and run for sample problems.

The basic contribution of this thesis is the Proposed Algorithm which was developed during this research. Algorithm offers manual technique to achieve balances with reasonable level of efficiency by savings in computation time.

The Proposed Algorithm is also applied to a real case taken from ÖZKÖSEOGLU Manufacturing Company which makes Industrial Furnaces, in order to compare its performance against the most commanly used computerized technique of ARCUS.

It was observed that for this size of problem, the Proposed Algorithm performed equally efficient, without using any computer time.

# Ö Z E T

Bu tezde, Montaj Hatlarının Dengelenmesi problemi yapısı ve mevcut algoritmalarla birlikte ayrıntılı olarak incelenmiştir. Hat Dengeleme prosesinin temel kavramları ve etkinlik ölçüsü tanımlanmıştır. Tam Model, Karma-Tamsayılı Programlama kullanılarak ortaya konmuştur.

Bu tezde sunulan, Algoritma Teklifi ile birlikte, etraflı bir literatür taraması yapılarak verilmiştir. Bu literatür incelemesinde başlıca i) Kilbridge ve Wester Algoritması, ii) R.P.W. pozisyon ağırlıklarına göre sıralama algoritması, iii) COMSOAL ve iv) LCR-Büyük Aday kuralı ve bunları takiben en yaygın olarak kullanılan kompüterize teknikler olarak Hoffman ve Arcus Algoritmaları tartışılmıştır. Bundan sonra, SLB, Stokastik Hat Dengelemesi incelenmiş ve özellikle BREENNECKE Algoritması kullanılarak açıklamalar yapılmıştır. Daha sonra da COA-Maliyete Dayalı Yaklaşım ve Stokastik Montaj Hattı Dengelemesi için tercihli sıraya göre dinamik programlama açıklanmıştır. Üç algoritma için akış diagramları ve Fortran IV diliyle kompüter programları yapılmış ve örnek problemler için çalıştırılmıştır.

Bu tezin, temel katkısı ise, araştırma sırasında geliştirilen PA-Teklif Algoritmadan oluşmaktadır. Bu teklif algoritma, dengelemeyi, makul bir verim oranı ve hesaplama zamanında önemli tasarrufla veren bir manuel teknik oluşturmaktadır.

Teklif Algoritma, Endüstriel Fırınlar imalatcısı olan ÖZKÖSEOĞLU İmalat Anonim Şirketi'nden alınan bir gerçek probleme uygulanarak, performansı, çok yaygın olarak kullanılan ve kompüterize bir teknik olan ARCUS Algoritması ile karşılaştırılmıştır.

Çalışmada, bu çaptaki problemler için, teklif edilen algoritmanın herhangi bir kompüter zamanına ihtiyaç göstermeksizin, Arcus Algoritması kadar etkili olduğu saptanmıştır.

# LIST OF TABLES

# LIST OF FIGURES

# TABLE OF CONTENTS

# CHAPTER I

# INTRODUCTION

Frederick W.Taylor and others have indicated a division of labor, that is another group of specialists largely mechanical and industrial engineers, were required to develop a science of planning for manufacturing just as the necessity of development in science of design, earlier.

The manufacturing problems had developed to such an extent that the ordinary mechanic could no langer cope with all the combinations of art and science necessary to deliver the goods on time. As the machines became more complex and production schedules became larger and more rigid, it was impossible to continue under the conventional approach of mechanic. This conventional approach is being modified since the industrial revolution by techniques such as sequencing, inventory control techniques, assembly line balancing techniques etc.

In this thesis assembly line balancing techniques are being studied.

Since early 1950's assembly lines are tackled in order to achieve better balances. Some have tackled them as queueing systems, some others seeked for an exact solution by making use of mathematical programming. Simulation was also an alternative tool. But, as in all complex production problems most successive algorithms are heuristic approaches.

## I.1. The Line-Balancing Problem

Balance refers to the equality of capacity or output of each of the successive operations (stations) in the sequence of a line. If they are all equal, perfect balance is achieved and smooth flow is expected. If they are unequal, the maximum possible output for the line as a whole will be dictated by the slowest operation in the sequence. This slow or bottleneck operation restricts the flow of parts on the line in the same way that a half-closed valve restricts the flow of water, even though the pipes in the system might be capable of carrying twice as much water.

To achieve balance to the best of an ability, there is a need to know the performance times for the smallest possible whole units of activity, such as tightening a bolt or making a solder joint, and the knowledge of the flexibility in the sequence of these tasks or activities. There are certain limitations on the sequence of the tasks. This sequence flexibility is important in order to specify groups of tasks making up operation or station for a line that achieves the best balance.

## I.2. Different Approaches to Line-Balancing Problem

Waiting line theory has not been a fruitful approach in the balance problem, even though it gives useful insights into how a line functions. Actually, it could be seen that, there is a work-time distribution to deal with, and the balance problem is a stochastic one. However, most of the work on line-balancing, has simplified the basic problem and assumed that the service times were constant values and that there is always work available at each stage; that is, they have worked out solutions for the deterministic case.

In this thesis firstly in chapter II. the definition and the exact formulation of the Assembly-line-Balancing Problem is given, by means of an example. Following this chapter, literature survey is presented with the addition of proposed algorithm which is developed in this thesis. Among the algorithms covered in this literature survey;

i) Kilbridge and Wester Algorithm,
ii) Ranked Positional Weight Algorithm,
iii) COMSOAL and
iv) Largest Condidate Rule are the oldest ones. Following these are two of the commonly used computerized techniques of HOFFMAN and ARCUS. Then stochastic line balancing is studied and described by using BRENNECKE's Algorithm. Finally chapter is closed by describing Cost-Oriented Approach and a Preference Order Dynamic Program for Stochastic Assembly line-Balancing.

Before the discussion of results in conclusion chapter the proposed algorithm is tested against the most commonly used technique of ARCUS' on the case study at ÖZKÖSEOĞLU ISI SANAYİİ ve TİCARET A.Ş. who manufactures Industrial Furnaces, Burners, etc. The line for ELM-50 Flour Sifting and Loosening Machine has been studied and balanced.

# CHAPTER II

## DEFINITION AND FORMULATION OF THE ASSEMBLY LINE BALANCING PROBLEM

When we speak of continuous systems, we normally think of manufacturing system that are organized and physically laid out by product. Whatever is being processed moves through a sequence of operations at rates that approach continuous movement, and thus the name arises. Continuous movement is, of course, a relative term to distinguish the character of such systems from batch processing where, by contrast, movement is intermittent.

The managerial decision to organise the work on a product or line basis is significant one. If we are to organise for continuous flow, the following requirements should be met.

i) Volume adequate for reasonable equipment utilization,
ii) Reasonably stable product demand,
iii) Product standardization,
iv) Part interchangeability,
v) Continuous supply of material.

Each of these requirements needs to be qualified. The concept of an adequate volume presumes an economic analysis to determine the breakeven volume between line organization and alternatives. Reasonably good equipment utilization is associated with high volume. Stable demand is required in terms of a minimum sum that would at least cover the special tooling cost for a line. Thus stable demand is associated with product standardization. Engineering changes in product designs can be accomodated by production lines, but they cannot be too frequent. We must have our economical sum to cover the cost of redoing and relayout that design changes may require. Part interchangeability is required so that no special reworking or refitting of parts is needed during assembly. If parts are not interchangeable at assembly, the flow of work is distrupted because of in balance. Finally, where we have the high-volume standardized product situation described by the foregoing requirements, continuous supply of material is crucial. The lack of supply of a simple part or item of raw material can force the entire process to be stopped, and the resulting downtime costs can be very large.

## II.1. Problem Definition

Assembly lines are characterized by the movement of the workpiece from one work station to the next. The individual tasks required to complete the product are divided and assigned to the work stations so that each station performs the same operation on every unit of product.

An assembly line balancing problem consists of assigning the individual tasks to the work stations, without violating the precedence relations, in such a way that some appropriate measure of line performance is optimized. The most commonly used measure of line performance is balance delay. In next section, while the formulation of problem is given, balance delay is defined as well.

## II.2. Problem Formulation

At this stage it is possible to formulate assembly line balancing problem in mathematical terms. Let,

C. - cycle time,
$t_i$ - the duration of $i^{th}$ work element,
d - balance delay,
J - the total sum of work elements,
N - number of work stations.

Firstly, the balance delay in general is;

$$d = N.C - \sum_{i=1}^{J} t_i \qquad N \text{ integer.}$$

The objective function is then defined as, minimization of the balance delay, d. From the above expression the objective function reduces to minimizing the product

$$N.C$$

Assuming that the sequence of the operations does not influence their duration).

Therefore assembly line balancing problem could be stated as either one of the following,

i) Minimize the number of work stations for a given cycle time, or
ii) For a given number of work stations, minimize the cycle time.

As, it is the case in most optimizations, there are restrictions in balancing problem. The balancing restrictions in general are;

i) The cycle time has to be larger than the duration of the largest work element:

$$t_{i_{max}} \leq C$$

ii) The work content of any single station has to be smaller than the cycle time:

$$\sum_{i=1}^{J} t_i \cdot x_{ij} \leq C \qquad j = 1,2,\ldots N.$$

Where,

$$x_{ij} \begin{cases} 1 - \text{if work element } i \text{ is assigned to work station } j. \\ 0 - \text{if work element } i \text{ is not assigned to work station } j. \end{cases}$$

iii) Each work element is to be assigned only once:

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1,2,\ldots J.$$

iv) Technological restrictions referring to the sequence of the operations can be formulated as a precedence matrix and must not be violated.

The assembly line balancing problem has received considerable attention in the past several years, and, as a result, there are a great many solution procedures. The procedures may be classified generally as either exact methods that quarantee an optimal solution, or heuristic methods that may yield only approximately optimal solution. The exact methods are usually mathematical programming formulations. Many types of heuristic methods have been proposed, ranging from simple procedures easily performed by hand to complex algorithms that require a computer for problem solutions. Some of these methods are reviewed in the next chapter.

## II.3. The Mixed-Integer Programming Model

The model will be discussed by means of a numerical example, which is given in Table II.1.

Let us assume that in this example the required cycle time is C = 10.

TABLE II.1. A Numerical Example

| Work Element | Duration | Direct Predecessors |
|---|---|---|
| 1 | 6 | - |
| 2 | 2 | 1 |
| 3 | 5 | 1 |
| 4 | 7 | 1 |
| 5 | 1 | 1 |
| 6 | 2 | 2 |
| 7 | 3 | 3,4,5 |
| 8 | 6 | 6 |
| 9 | 5 | 7 |
| 10 | 5 | 8 |
| 11 | 4 | 9,10 |

The minimum number of work stations is being asked. Since we do not know whether the $N_{min} = 5$ is sufficient or not (considering the existing constraints) the model is formulated for seven work stations (A-G). Writing $A_i$ for the duration of work element i if assigned to station A, the following constraints are obtained:

i) Capacity constraints.

$$A_1 + A_2 + A_3 + \dots + A_{11} \leq 10$$

$$B_1 + B_2 + B_3 + \dots + B_{11} \leq 10$$

$$\vdots$$

$$G_1 + G_2 + G_3 + \dots + G_{11} \leq 10$$

ii) Each work element isto be assigned only once:

$$A_1 + B_1 + C_1 + \dots + G_1 = 6$$

$$A_2 + B_2 + C_2 + \dots + G_2 = 2$$

$$\vdots$$

$$A_{11} + B_{11} + C_{11} + \dots + G_{11} = 4$$

iii) Indivisibilityof the work elements:

$$\frac{1}{6} A_1 + A_1' = 1, \quad \frac{1}{6} B_1 + B_1' = 1, \quad \dots \quad \frac{1}{6} G_1 + G_1' = 1$$

$$\frac{1}{2} A_2 + A_2' = 1, \quad \frac{1}{2} B_2 + B_2' = 1, \quad \dots \quad \frac{1}{2} G_2 + G_2' = 1$$

$$\vdots$$

$$\frac{1}{4} A_{11} + A_{11}' = 1, \quad \frac{1}{4} B_{11} + B_{11}' = 1, \quad \dots \quad \frac{1}{4} G_{11} + G_{11}' = 1$$

$A_1', \dots, G_{11}'$ can be 0 or 1 (77 of them)

iv) Precedence Constraints: (91 constraints)

$$\frac{1}{2} A_2 \leq \frac{1}{6} A_1$$

$$\frac{1}{2} B_2 \leq \frac{1}{6} A_1 + \frac{1}{6} B_1$$

$$\frac{1}{2} C_2 \leq \frac{1}{6} A_1 + \frac{1}{6} B_1 + \frac{1}{6} C_1$$

$$\vdots$$

$$\frac{1}{4} G_{11} \leq \frac{1}{5} A_{10} + \frac{1}{5} B_{10} + \frac{1}{5} C_{10} + \frac{1}{5} D_{10} + \frac{1}{5} E_{10} + \frac{1}{5} F_{10} + \frac{1}{5} G_{10}$$

The objective function is;

$$\text{minimize } Z = 1\ F_{11} + 3\ G_{11}$$

with 186 constraints (154 integer variables) mentioned above.

The objective function contains the sixth and seventh work station, ones that have been added to the minimal number of work stations (Fand G). These two work stations are weighted with increasing cost factors. For this case the theoretical minimum has been reached with following solution;

station A: work elements 1,2,6
station B: work elements 5,8
station C: work elements 3,10
station D: work elements 4,7
station E: work elements 9,11

# CHAPTER III

# MOST COMMONLY USED ASSEMBLY LINE BALANCING METHODS SINCE 1950's

In this chapter Assembly Line Balancing Methods will be discussed under two categories such as, Deterministic and Stochastic Algorithms, followed by the proposed algorithm in last section.

## III.1. Deterministic Assembly Line Balancing Algorithms

Since 1954 many papers and books have been published reporting the development of new techniques for balancing line production systems. Perhaps the best known among these are those rely on heuristic procedures. Four of these are;

i) Kilbridge and Wester Method(11)
ii) Ranked Positional Weights(19)
iii) The Largest Candidate Rule(19)
iv) COMSOAL(1).

These heuristic procedures will be described in some detail in following sections. In sections III.1.5 and III.1.6 following computerized algorithms are briefly described by means of an hypothetical example;

i) Line Balancing with "Precedence Matrix" - T.R.HOFFMANN(3,5).
ii) Random Generation of Solutions - ARCUS(1)

### III.1.1. Kilbridge and Wester Method

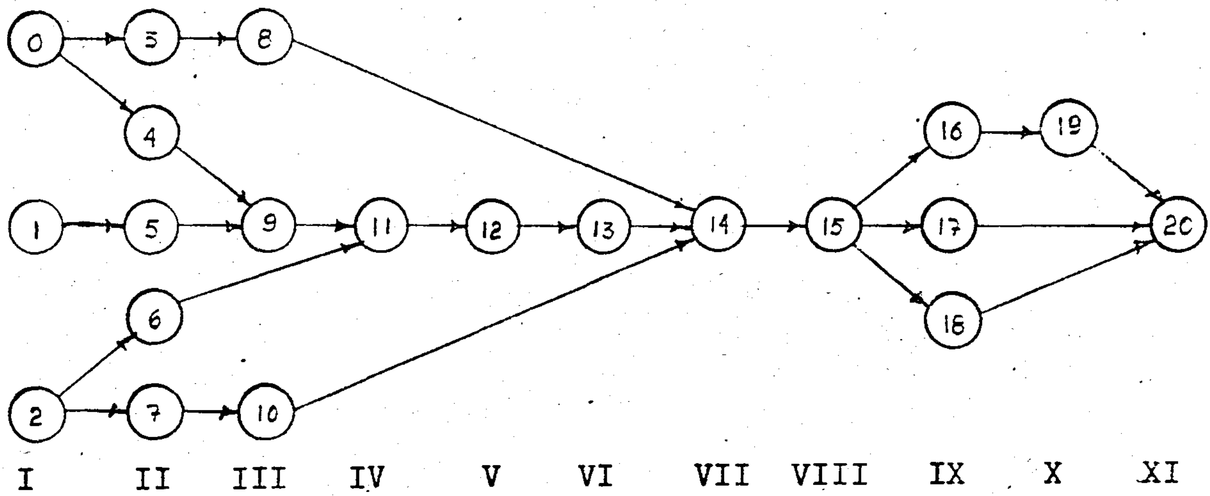
This simple, one of the earliest, heuristic method of flow-line balancing is best described by means of an example. Assume an assembly of desk lamp with 21 parts as data given below at Table III.1.

TABLE III.1. Data for the Desk Lamp

| Elements | Element Times(min) | Precedence(must follow) |
|---|---|---|
| 0 | 6 | - |
| 1 | 5 | - |
| 2 | 8 | - |
| 3 | 9 | 0 |
| 4 | 5 | 0,1 |
| 5 | 4 | 1 |
| 6 | 5 | 2 |
| 7 | 6 | 2 |
| 8 | 10 | 3 |
| 9 | 5 | 4,5 |
| 10 | 10 | 7 |
| 11 | 2 | 9,6 |
| 12 | 5 | 11 |
| 13 | 4 | 12 |
| 14 | 12 | 13,8,10 |
| 15 | 10 | 14 |
| 16 | 5 | 15 |
| 17 | 15 | 15 |
| 18 | 10 | 15 |
| 19 | 5 | 16 |
| 20 | 6 | 17,18,19 |

From Table III.1. the precedence diagram could be drawn as in Figure III.1.

Figure III.1. Precedence Diagram for the Desklamp.




Roman figures above the Figure III.1. are column numbers, which will be needed for transferability later on. Elements appearing in column I can be started immediately, those in column II can begin after one or more in column I have been completed, and so on. The data show in this diagram can now be represented in useful tabular form.

Column (c) in Table III.2. describes the lateral transferability of elements among columns. For example, element 6 can be performed in column III as well as in Column II without violating precedence constraints. Suppose that the cycle time is 36 minutes to balance the assembly line.

TABLE III.2. Tabular Form of Precedence Diagram.

| Column (a) | Element (b) | Transferability (c) | Element dur. (d) | Dur.of Col. (e) | Cumulative Dur. (f) |
|---|---|---|---|---|---|
| | 0 | | 6 | | |
| I | 1 | | 5 | | |
| | 2 | | 8 | 19 | 19 |
| | 3 | III-V(with 8) | 9 | | |
| | 4 | | 5 | | |
| II | 5 | | 4 | | |
| | 6 | III | 5 | | |
| | 7 | III-VI(with 10) | 6 | 29 | 48 |
| | 8 | IV-VI | 10 | | |
| III | 9 | | 5 | | |
| | 10 | IV-VI | 6 | 21 | 69 |
| IV | 11 | | 2 | 2 | 71 |
| V | 12 | | 5 | 5 | 76 |
| VI | 13 | | 4 | 4 | 80 |
| VII | 14 | | 12 | 12 | 92 |
| VIII | 15 | | 10 | 10 | 102 |
| | 16 | | 5 | | |
| IX | 17 | X | 15 | | |
| | 18 | X | 10 | 30 | 132 |
| X | 19 | | 5 | 5 | 137 |
| XI | 20 | | 6 | 6 | 143 |

Procedure:

1. Is there a duration in column f equal to the cycle time of 36? No.
2. Select the largest duration in column f less than 36, i.e. 19 for column I.
3. Subtract 19 from 36; 17.
4. Does one or more of the elements in the next column (II) equal to 17? No.
5. Select the smallest duration from column f which is larger than 36, i.e. 48 for columns I and II.
6. Can one or more of the elements in column I and II be transferred beyond column II so as to reduce the duration to 36? No, but element 3 (with 8) plus 6 can be
7. Select next largest duration from column f, i.e. 69 for columns I,II and III.
8. Can one or more of the elements in columns I,II and III be transferred beyond column III so as to reduce the duration to 36? No, the nearest is elements 3,8,7 and 10 which would give a duration of 38 which is too large.
9. Will an improved allocation of elements for station I be obtained by considering a larger duration from column f? No.
10. Adopt the best allocation found previously, i.e. step (4) which gave a work station time of 35 by column I plus elements 4,6 and 7.
11. Rewrite the table and calculate new figures for column f. (Table III.3).
12. Follow the same procedure for next stations from the new tables. Final result is given in Table III.4.

TABLE III.3. Tabular form of Precedence Diagram after the First Station

| Column (a) | Element (b) | Transferability (c) | Element dur. (d) | Dur.of Col. (e) | Cumulative Dur. (f) |
|---|---|---|---|---|---|
| | 0 | | 6 | | |
| I | 1 | | 5 | | |
| | 2 | | 8 | | |
| | | | | | Station I |
| | 4 | | 5 | | |
| II | 6 | | 5 | | |
| | 7 | | 6 | | (35) |
| | 3 | IV-V(with 8) | 9 | | |
| III | 9 | | 5 | | |
| | 5 | | 4 | | |
| | 10 | IV-VI | 6 | 24 | 24 |
| IV | 8 | V-VII | 10 | | |
| | 11 | | 2 | 12 | 36 |
| V | 12 | | 5 | 5 | 41 |
| VI | 13 | | 4 | 4 | 45 |
| VII | 14 | | 12 | 12 | 57 |
| VIII | 15 | | 10 | 10 | 67 |
| | 16 | | 5 | | |
| IX | 17 | X | 15 | | |
| | 18 | X | 10 | 30 | 97 |
| X | 19 | | 5 | 5 | 102 |
| XI | 20 | | 6 | 6 | 108 |

TABLE III.4. Tabular form of the Solution for Desk Lamp by Kilbridge & Wester

| Station Number | Elements | Station Work Content (sec) |
|---|---|---|
| I. | 0 | |
| | 1 | |
| | 2 | |
| | 4 | |
| | 6 | |
| | 7 | (35) |
| II. | 3 | |
| | 9 | |
| | 5 | |
| | 10 | |
| | 8 | |
| | 11 | (36) |
| III. | 12 | |
| | 13 | |
| | 14 | |
| | 15 | |
| | 16 | (36) |
| IV. | 17 | |
| | 18 | |
| | 19 | |
| | 20 | (36) |

The balancing loss for this solution is;

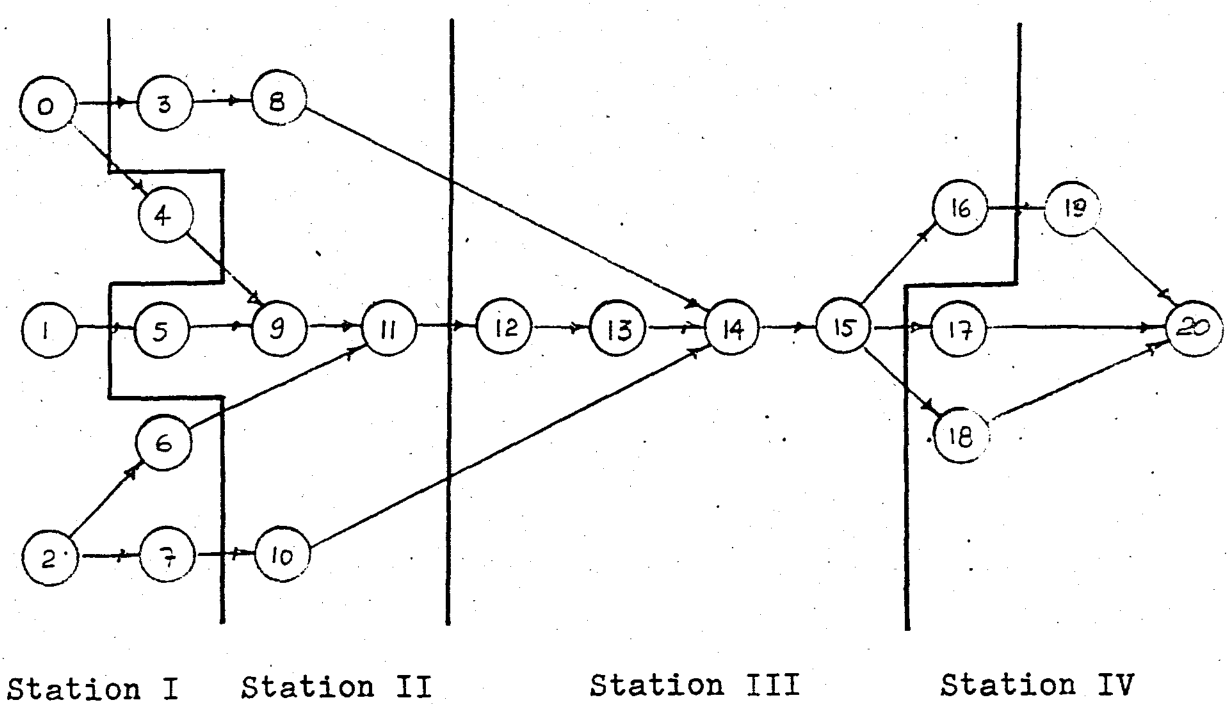
$$100.((N.T_c - \sum_{i=1}^{n} T_{E_i})/N.T_c) = \underline{0.7}\ \%$$

where $T_c$ - cycle time

$T_{E_i}$ = element time

$N$ = number of stations.

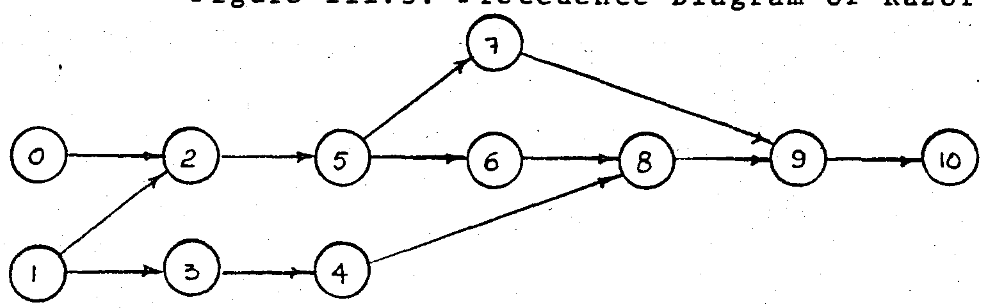
Figure III.2. Precedence Diagram with Work Stations drawn on it.



### III.1.2. Ranked Positional Weights

This procedure was developed by Helgeson and Birnie(19) of the General Electric Company of the U.S.A. in the late 1950's. It is a rapid but approximate method, which has been shown to provide acceptably good solutions more quickly than many of the alternative methods. It is capable of dealing with precedence constraints. This technique will be demonstrated by means of an assembly of razor with 11 parts. Precedence diagram is given in Figure III.3.

Figure III.3. Precedence Diagram of Razor




The following table is called the Precedence Matrix where the element number and time is given in first two columns. The middle of the table shows the precedences. For example, element 0 is immediately followed by element 2, which in turn is followed by 5, which is followed by 6 and 7 and so on. A simple mark indicates the elements which follow immediately and crosses indicate elements which follow because of their relationship with other elements. The final column of the table gives the positional weight for each task, which is defined as following; summing the elements own standard time and the standard times for all following elements. For ins-instance for element 0;

p.w. = element 0 (.32)
+ element 2 (.20)
+ element 5 (.23)
+ element 6 (.20)

+ element 7 (.05)
+ element 8 (.32)
+ element 9 (.10)
+ element 10(.32) = 1.72

The positional weight is therefore a mesaure of the size of an element. Then we rewrite Table III.5 in order of decreasing p.w. in Table III.6.

TABLE III.5. Precedence Matrix for Razor

| Element Number | Element Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Positional Weights. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | .32 | | | 1 | | | + | + | + | + | + | + | 1.72 |
| 1 | .10 | | | 1 | 1 | | + | + | + | + | + | + | 1.65 |
| 2 | .20 | | | | | | 1 | + | + | + | + | + | 1.40 |
| 3 | .05 | | | | | 1 | | | | + | + | + | .87 |
| 4 | .10 | | | | | | | | | 1 | + | + | .82 |
| 5 | .23 | | | | | | | 1 | 1 | + | + | + | 1.20 |
| 6 | .20 | | | | | | | | | 1 | + | + | .92 |
| 7 | .05 | | | | | | | | | | 1 | + | .45 |
| 8 | .32 | | | | | | | | | | 1 | + | .72 |
| 9 | .10 | | | | | | | | | | | 1 | .40 |
| 10 | .30 | | | | | | | | | | | | .30 |

TABLE III.6. Positional Weights in Decreasing Order

| Element Number | 0 | 1 | 2 | 5 | 6 | 3 | 4 | 8 | 7 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Element Time | .32 | .10 | .20 | .23 | .20 | .05 | .10 | .32 | .05 | .10 | .30 |
| Positional Weight | 1.72 | 1.65 | 1.40 | 1.20 | .92 | .87 | .82 | .72 | .45 | .40 | .30 |
| Immediate Predecessors | - | - | 0.1 | 2 | 5 | 1 | 3 | 4.6 | 5 | 7.8 | 9 |

It is required to design an assembly line with the minimum number of stations to provide a cycle time of .55 hours. Using the Table III.6., elements are allocated to work station in order of decreasing P.W. and without violating the precedence constraints. Element 0 with highest p.w. of 1.72 is allocated first to station I. This is acceptable because element 0, has no immediate predecessor and furthermore its element time is less than the space time available in station. I. element 1 is next to be allocated since it has the next highest P.W. Everytime we assign one of the elements to stations we rearrange Table III.6 by dismissing that element and apply the same rule for assignments and so on.

Solution for this simple hyphotetical example problem is given in Table III.7.

TABLE III.7. Solution Achieved for Razor by Ranked P.W.

| Work Station | Element Number | Positional Weight | Station Work Content |
|---|---|---|---|
| I | 0 | 1.72 | .32 |
| | 1 | 1.65 | .42 |
| | 3 | .87 | (.47) |
| II | 2 | 1.40 | .20 |
| | 4 | .82 | .30 |
| | 5 | 1.20 | (.53) |
| III | 6 | .92 | .20 |
| | 8 | .72 | (.52) |
| IV | 7 | .45 | .32 |
| | 9 | .40 | .37 |
| | 10 | .30 | (.47) |

This solution yields a balancing loss of;

10.4 %

It is not difficult to criticise this simple Assembly line Balancing procedure or to suggest improvements. Mansoor(13) has developed and improved R.P.W. technique involving a 'backtracking' procedure which appears to provide better results in certain conditions. Flow diagram for Line Balancing for a given cycle time using R.P.W. method is given in Figure III.4.

Figure III.4. Flow Diagram for R.P.W. Technique.

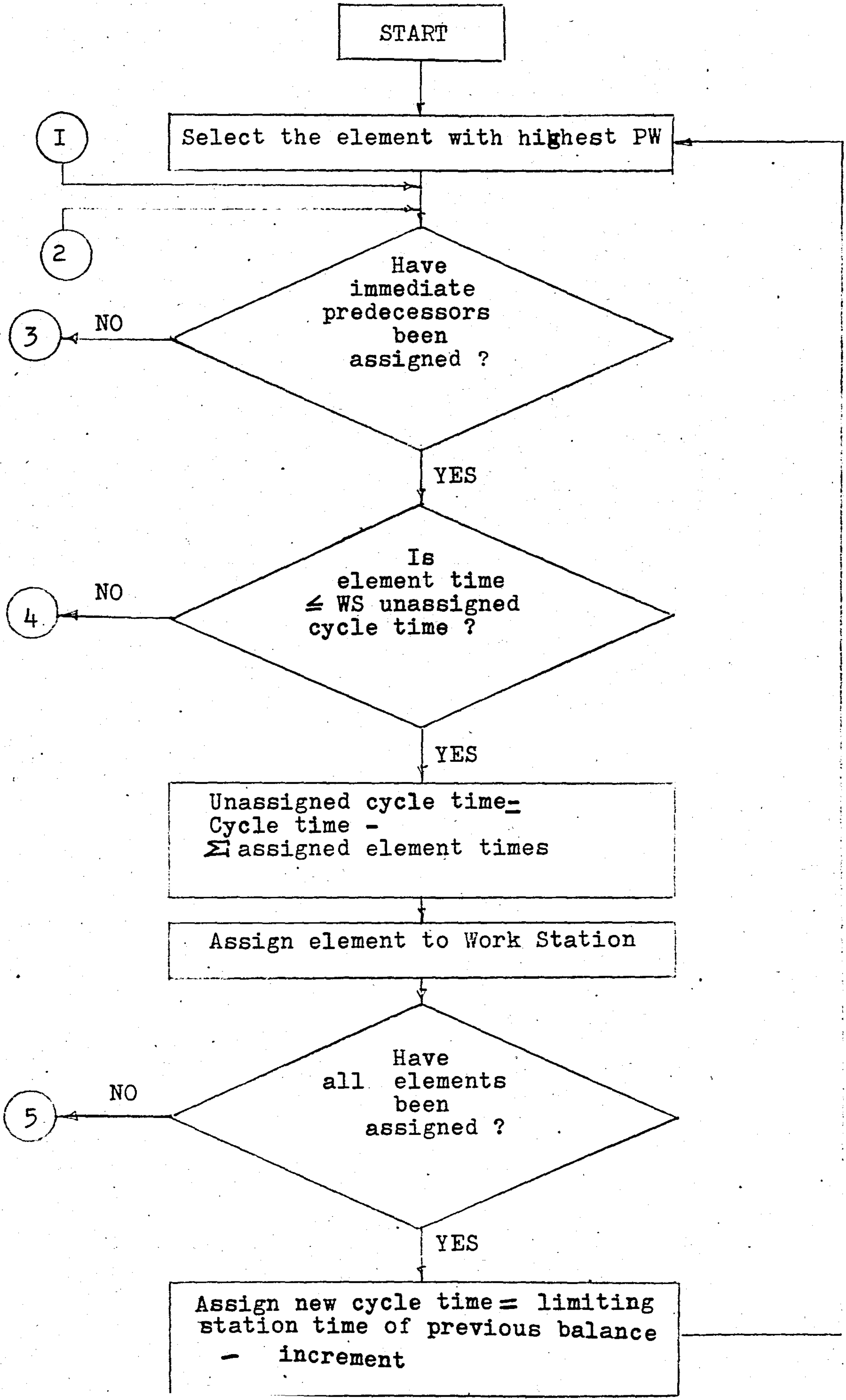
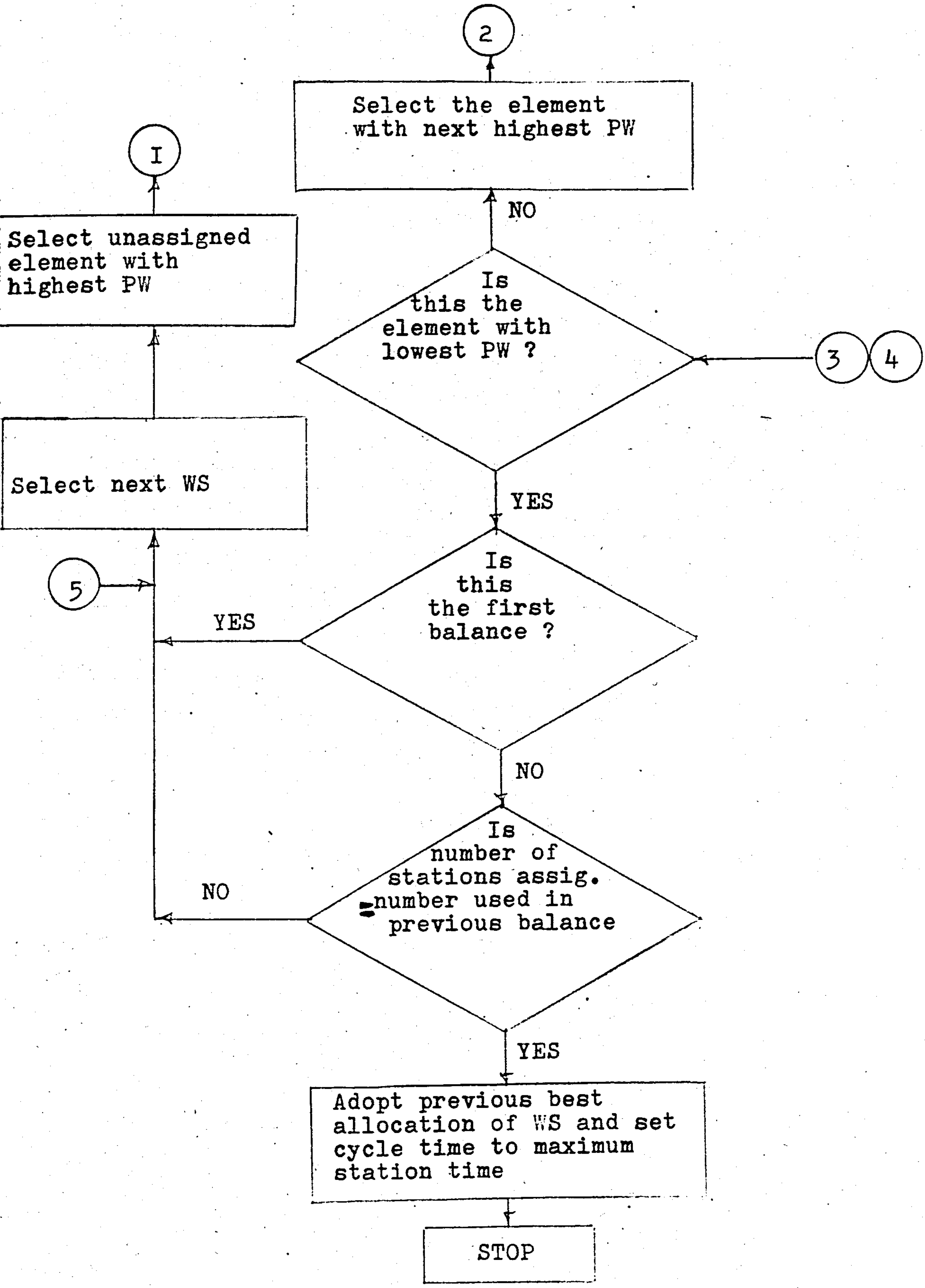


Figure III.4.Cont.



### III.1.3. COMSOAL

COMSOAL(1), an acronym for a Computer Method of Sequencing Operations for Assembly Lines, is a method of balancing large complex machine-paced assembly lines. It combines four powerful and versatile tools; a high speed digital computer, sampling concepts, the Monte Carlo technique, and simulation. In order to explain this algorithm assume the assembly of an pencil sharpener, given in Table III.8.

TABLE III.8. Data. For the Pencil Sharpener

| Element Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of Immediate Predecessors | - | - | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 |

Procedure:

1. Construct a list of (COMSOAL LIST A) total number of immediate predecessors of elements.

2. Compile the list B of elements with no immediate predecessors. This list is called Partition List B.

| element number | 0 | 1 |
|---|---|---|
| number of immediate predecessors | - | - |

3. Place, in List C, tasks which have times no greater than time left available at station being worked. List C is called Fit List of COMSOAL.

4. Select, randomly, one element from Fit List.

5. Eliminate selected element (0) from both list B and C.

6. Update list A by scanning followers of task (0) and deduct 1 from total number of preceding tasks (Redo List) predecessors of 2 becomes 1, etc.

7. Redo list B by transferring from List A all tasks with no predecessors.

8. Transfer from List B to C those tasks which fit the remaining time to be assigned.

9. Select from fit list another task for station I.

Repeat the same until station I is full and continue station by station. Following table gives the solution for C=.55.

TABLE III.9. Solution to the problem with Lists of COMSOAL

| List A | | List B | List C | assigned elements | |
|---|---|---|---|---|---|
| 0 | - | 0 | 0 | t(0) = .32 | |
| 1 | - | 1 | 1 | | |
| 2 | 2 | | | | |
| 3 | 1 | | | | |
| 4 | 1 | | | | |
| ⋮ | ⋮ | | | | |
| 10 | 1 | | | | |
| 1 | - | 1 | 1 | t(0) = .32 | |
| 2 | 1 | | | t(1) = .10 | |
| 3 | 1 | | | .42 | |
| 4 | 1 | | | | |
| 5 | 1 | | | | |
| ⋮ | ⋮ | | | | |
| 10 | 1 | | | | |
| 2 | - | 2 | 3 | t(0) = .32 | |
| 3 | - | 3 | | t(1) = .10 | station |
| 4 | 1 | | | t(3) = .05 | I |
| 5 | 1 | | | .47 | |
| ⋮ | ⋮ | | | | |
| 10 | 1 | | | | |
| ⋮ | ⋮ | | | | |
| 7 | - | 7 | 7 | t(7) = .05 | |
| 9 | 1 | | | | |
| 10 | 1 | | | | |

BOĞAZİÇİ ÜNİVERSİTESİ KÜTÜPHANESİ

Stations become as following:

Station I - elements 0
1
3 with total S.W.C. of 0.47

Station II - elements 2
5
4 with total S.W.C. of 0.53

Station III - elements 6
8 with total S.W.C. of 0.52

Station IV - elements 7
9
10 with total S.W.C. of 0.45

this solution yields a balancing loss of; 10.4 %.

### III.1.4. Largest Candidate Rule

This simplest of the algorithms work as following: Allocate elements to stations beginning with the first station by scheduling from those elements that are feasible (precedence constraints) in descending order of size. For the same example problem Largest Candidate Rule yields the same solution as Ranked Positional Weight but it is quicker.

### III.1.5. Computer Program for Line Balancign With Precedence Matrix (T.R.HOFFMAN)(3)(5)

In this section the computer program for line-balancing with precedence matrix will be discussed. The flow diagram is presented in Figure III.5. This program is written for the UNIVAC system at Bosphorus University in FORTRAN IV.

Description of the technique:

The precedence matrix to be used by this method is constructed in the following manner;

i) Construct a table with task numbers as the top row and left side column. Term row numbers as (i) and column numbers as (j).

ii) If the element of row (i) immediately precedes the element of column (j), a 1 is placed as (ij) entry.

iii) All other entries are zero.

To use this matrix in generating all the feasible permutations, the following steps are used;

i) Sum each column and term the total a "code number" $c_i (i=1,2,...,n)$. Place this row at the bottom of the matrix.

ii) Label the diagonal of the matrix with any arbitrary letter or a large number (the program uses 16000).

Balance is obtained by inspecting the code number, $C_i$, for zeros. Zero as a code number indicates that job may be assigned to station K, because it has no precedence. The scheme for generating the feasible combination and balancing the line station by station is as follows:

i) Search left to right in the Code Number for zero.

ii) Select the element which heads the column in which the zero is located.

iii) Subtract the element's time from cycletime remaining.

iv) If the result is positive, go to step (v). If it is negative go to (vi)

v) Subtract from the Code Number the row corresponding to the element selected and use this result as a new Code Number. Go to step (vi).

vi) Go to step (i) and start search one element to the right of the one just selected and repeat steps (i-vi). Until all the columns have been examined, then goto step (vii).

vii) Subtract the remaining cycle time (the slack time) from the slack time of the previous combination generated.

viii) If zero or negative, go to step (ix). If positive, then this set of elements just generated becomes the new combination for this station. Go to step (x).

ix) Go back one Code Number, $C_i$, and go back to step (i), starting one element to the right of the element which has been selected from that Code Number. Repeat this procedure until the last column of the first Code Number has been tested; the result is that the last combination generated by step (viii) is the one having the maximum elemental time for this station.

Figure III.5. Flow Chart for HOFFMANN'S Technique

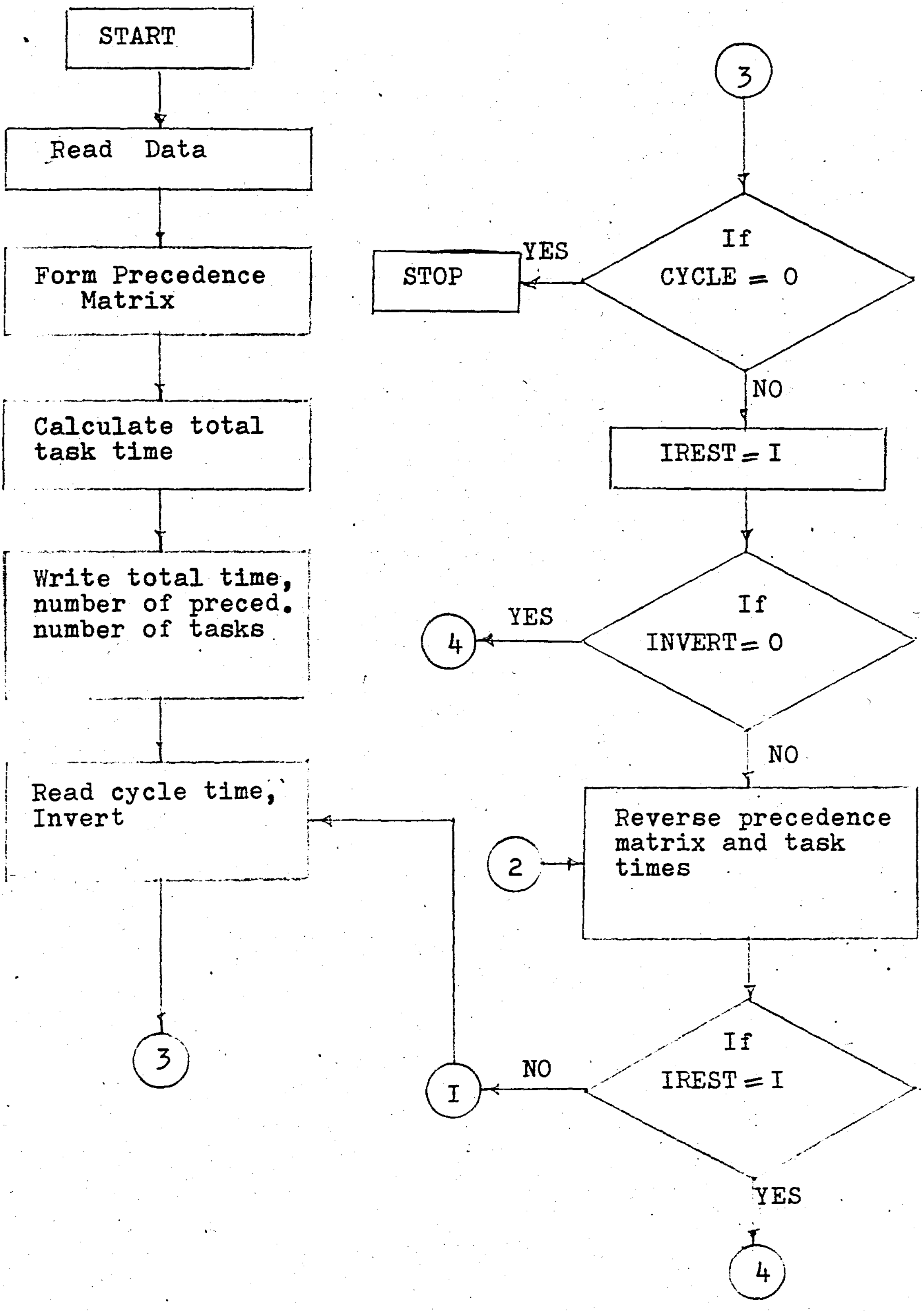
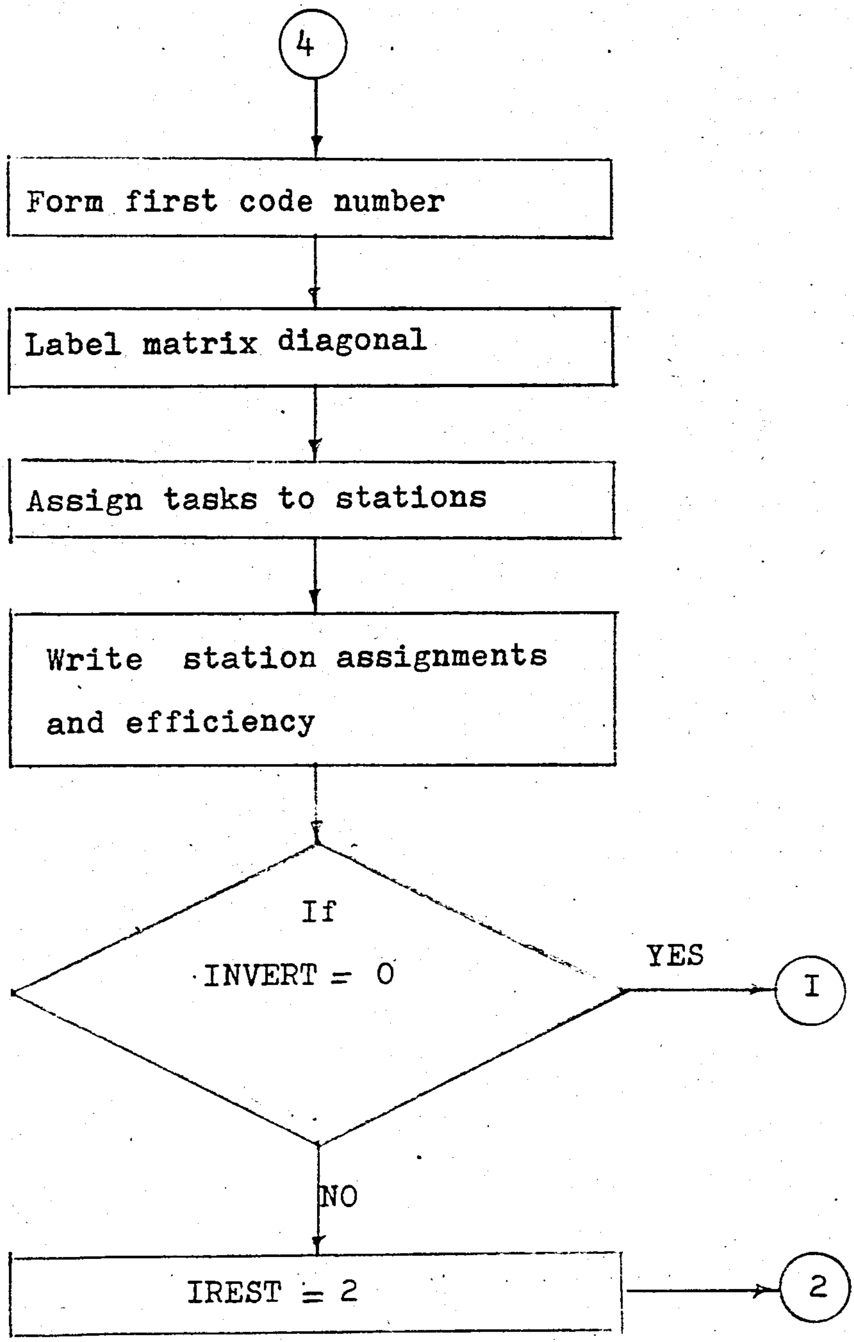


Figure III.5. Cont.



x) Replace the first Code Number with the last Code Number corresponding to the previous result. (This eliminates from further consideration of the elements already selected).

xi) Repeat the previous steps until all the elements have been assigned. (Code Number $C_i$ is entirely negative).

It is easily seen that this precedence matrix method is very mechanical and because of this, it is easily suited as a computer technique, known as HOFFMANN's TECHNIQUE.

The computer listing and the output for the following example is given in Appendix A.

Figure III.6. Precedence Diagram for the Example of HOFFMANN's TECHNIQUE

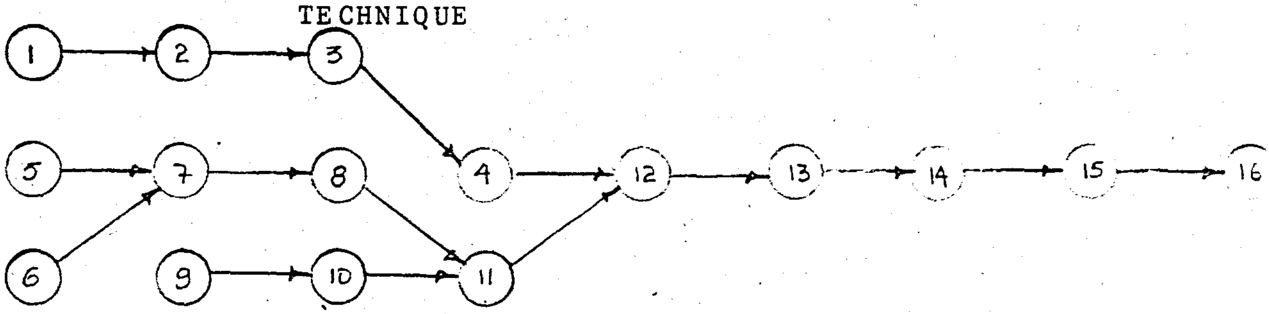



TABLE III.10- Data of the Precedence Diagram Given in Fig. III.6.

| Task | Times(mins.) |
|---|---|
| M01 | 0.062 |
| M02 | 0.065 |
| M03 | 0.227 |
| M04 | 0.236 |
| M05 | 0.088 |
| M06 | 0.125 |
| M07 | 0.063 |
| M08 | 0.057 |
| M09 | 0.119 |
| M10 | 0.118 |
| M11 | 0.082 |
| M12 | 0.319 |
| M13 | 0.288 |
| M14 | 0.181 |
| M15 | 0.201 |
| M16 | 0.231 |

III.1.6. Program for Random Generation of Solutions - ARCUS(1)

This one of the most commonly used technique generates random feasible solutions and selects the one with minimum number of work stations.

Description of the technique:

For a given cycle time this method finds the minimum number of work stations (possible) in the following manner:

i) Select a task i. (Randomly)

ii) Check the precedence for this task i.
   a) If task cannot be assigned goto (i)
   b) If task can be assigned goto (iii)

iii) Add task i's elemental time to Station Work Content.
   a) If S.W.C. > Cycle time, proceed to (iv)
   b) If S.W.C. ≤ Cycle time, assign task i to the station and proceed to (v).

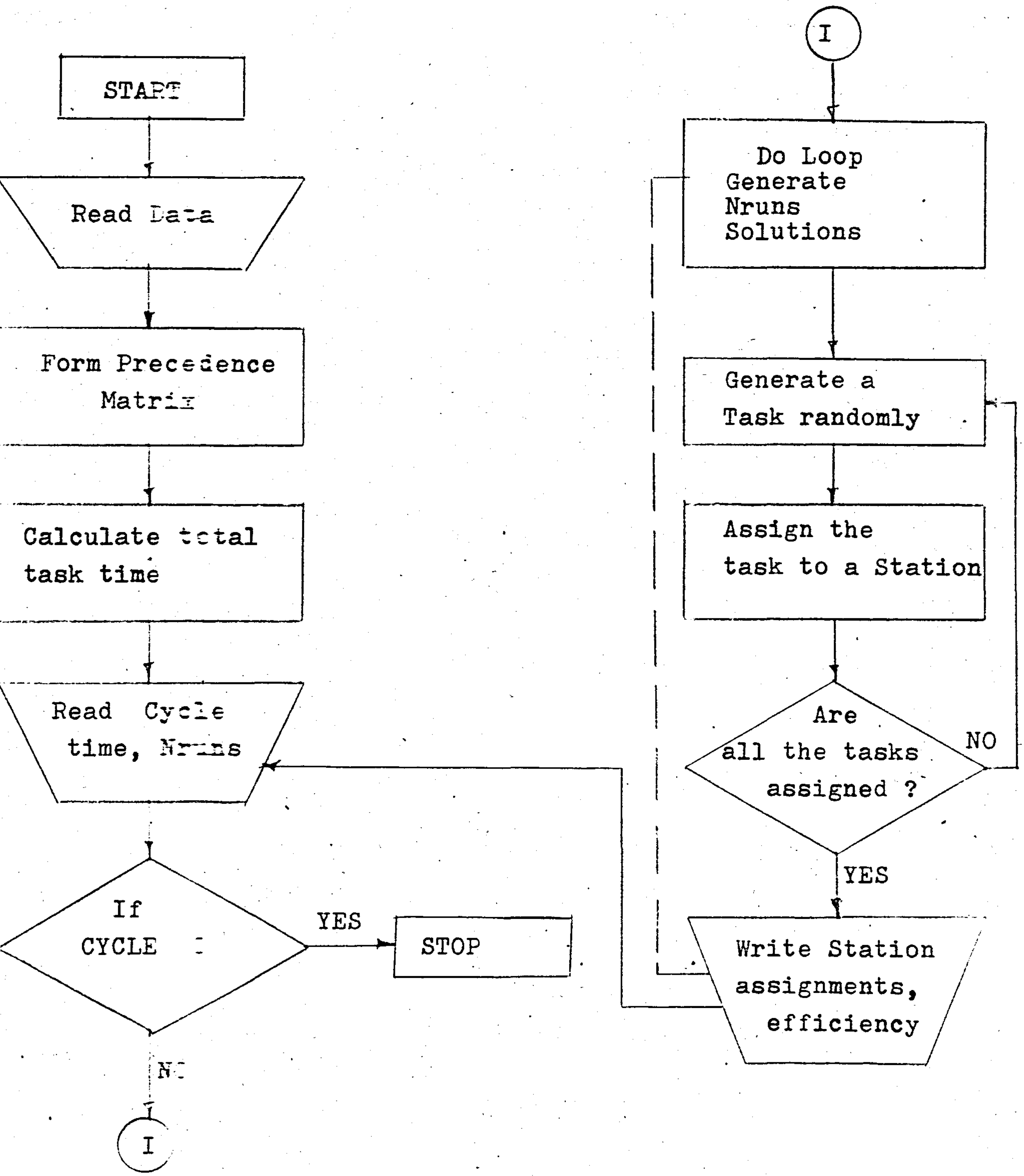
iv) Assign task i to next station and calculate the idle time (slack time)

v) Adjust the precedence matrix for the assignment of task i.

vi) Repeat steps (i-v) until all the tasks are assigned.

Using the same example problem as in last section, the FORTRAN IV Listing and the output of the run is presented in Appendix B.

Figure III.7. Flow Diagram for Random Generetion of Solutions



## III.2. Stochactic Assembly Line-Balancing Algorithms

It was noted, in the introduction, that the task times are in fact represented by some distribution, instead of as constant values. The stochastic nature of the work station times has some interesting implications for the design of line systems and for line-balancing.

### III.2.1. Two Parameter Assembly Line Balancing Model- D.BRENNECKE(2)

Theoretical foundation of the Two Parameter Model of Brennecke is three of the theorems in statistics, which deal with the distributions of sums and differences;

i) Given the distribution of a variable x and the distribution of another variable y, the expected value of the variable x + y is;

$$E(x+y) = E(x) + E(y)$$

ii) The variance of x+y is;

$$\sigma^2_{x+y} = \sigma^2_x + \sigma^2_y$$

iii) If the distributions of x and y are normal, the distribution of the sum or difference (x±y) is also normal.

Brennecke also uses two assumptions;

i) The task times are assumed to be independent of each other when they are combined.

ii) The task time distributions are sufficiently normal that the station time distributions may be considered normal.

Procedure:

The mean time for the station is the summation of the mean times of the elements and the standard deviation is the square-root of the summation of the variances of those elements that make up the stations. Suppose Brennecke's Model will be used to solve for 0.10 probability (Ten Percent) to exceed the cycle time. Then;

i) Select the controlling station which is the station with the highest time, equal to the station mean time plus one standard deviation.

ii) The standard deviation for the distribution of time values for this station is multiplied by the normal deviate corresponding to the 0.10 tail of the normal curve. This is the limiting value since it is known that the probability of actual time values exceeding cycle time cannot be less than 0.10.

iii) Compute z, the number of standard deviations, between the mean time values of the other stations and the cycle time.

$$z = \frac{\text{station time (j) - cycle time}}{\text{standard deviation (j)}}$$

iv) Obtain the corresponding probability values for each z from the table of Standardized Normal Distribution Function.

v) The figure that is obtained by summing these probabilities gives the probability that the actual operation time will exceed the cycle time.

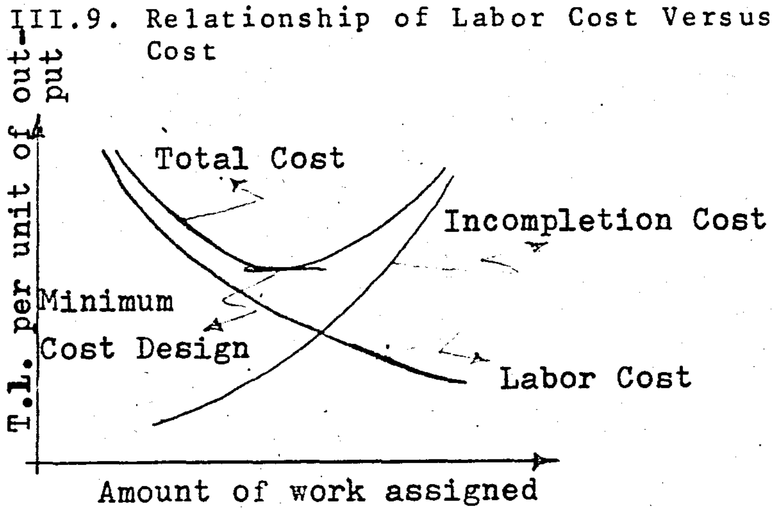
vi) Steps iii-v is repeated until designed 0.10 probability is obtained.

The optimal solution will be that solution which utilizes the least number of work stations. If there is a tie, the one with the smallest cycle time is selected. The flow chart is given in Figure III.8.

### III.2.2. A Cost Oriented Approach to Stochactic A L B

This heuristic approach is designed to reduce expected labor and incompletion costs. Labor cost is defined as; operating costs affected by line balancing are the cost of manning the line and the incompletion cost is; the cost arising from tasks not being finished as units move down the line. These two costs are inversely related. The more work we assign to workers the fewer workers needed. As a result of this incompletion cost increases. Figure III.9. shows how these costs vary per unit of output with the amount of work assigned to the worker.

Figure III.9. Relationship of Labor Cost Versus Incompletion Cost




Now the sum of these costs will be minimized.

Figure III.8. Flow Chart for Two Parameter ALB Model

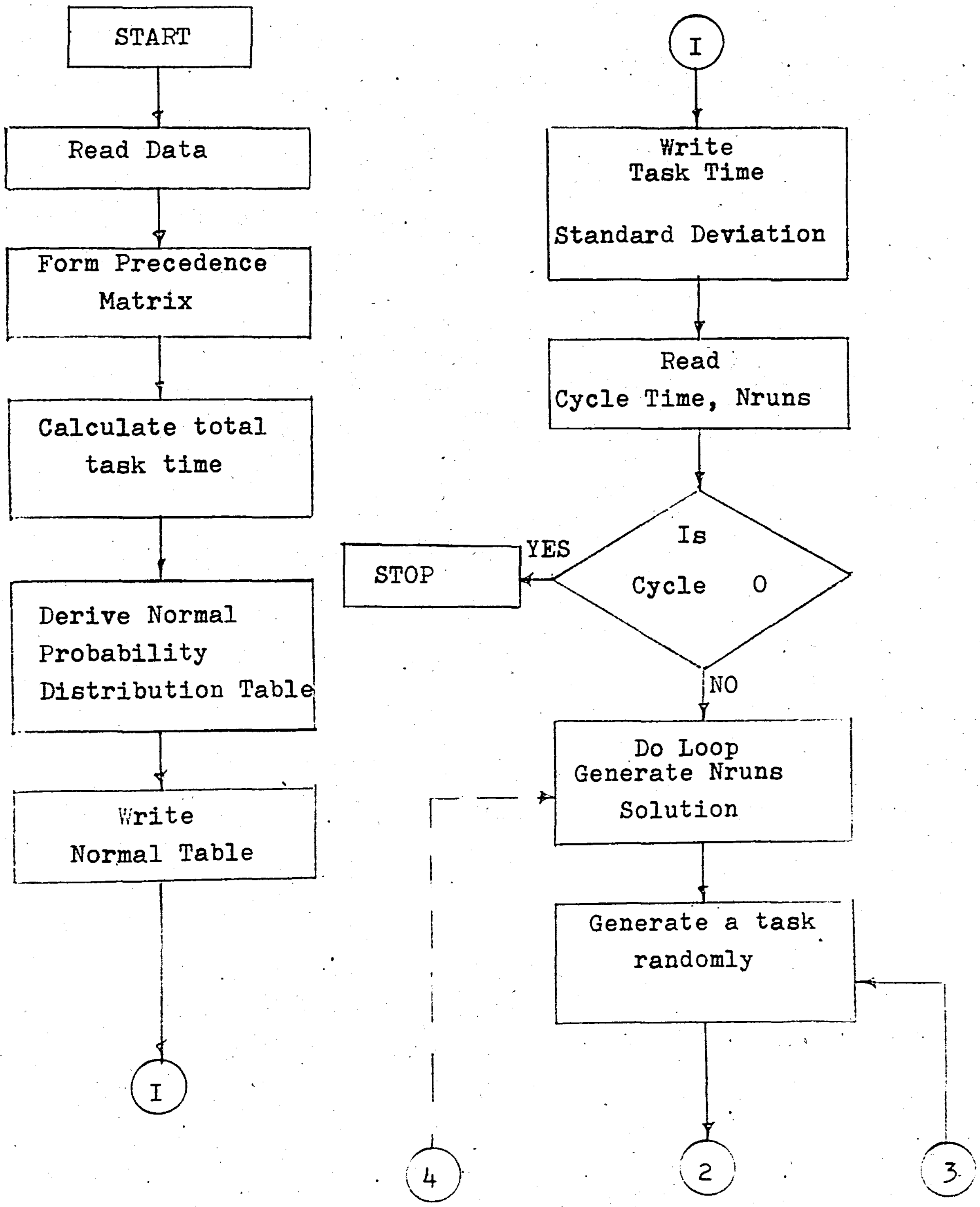
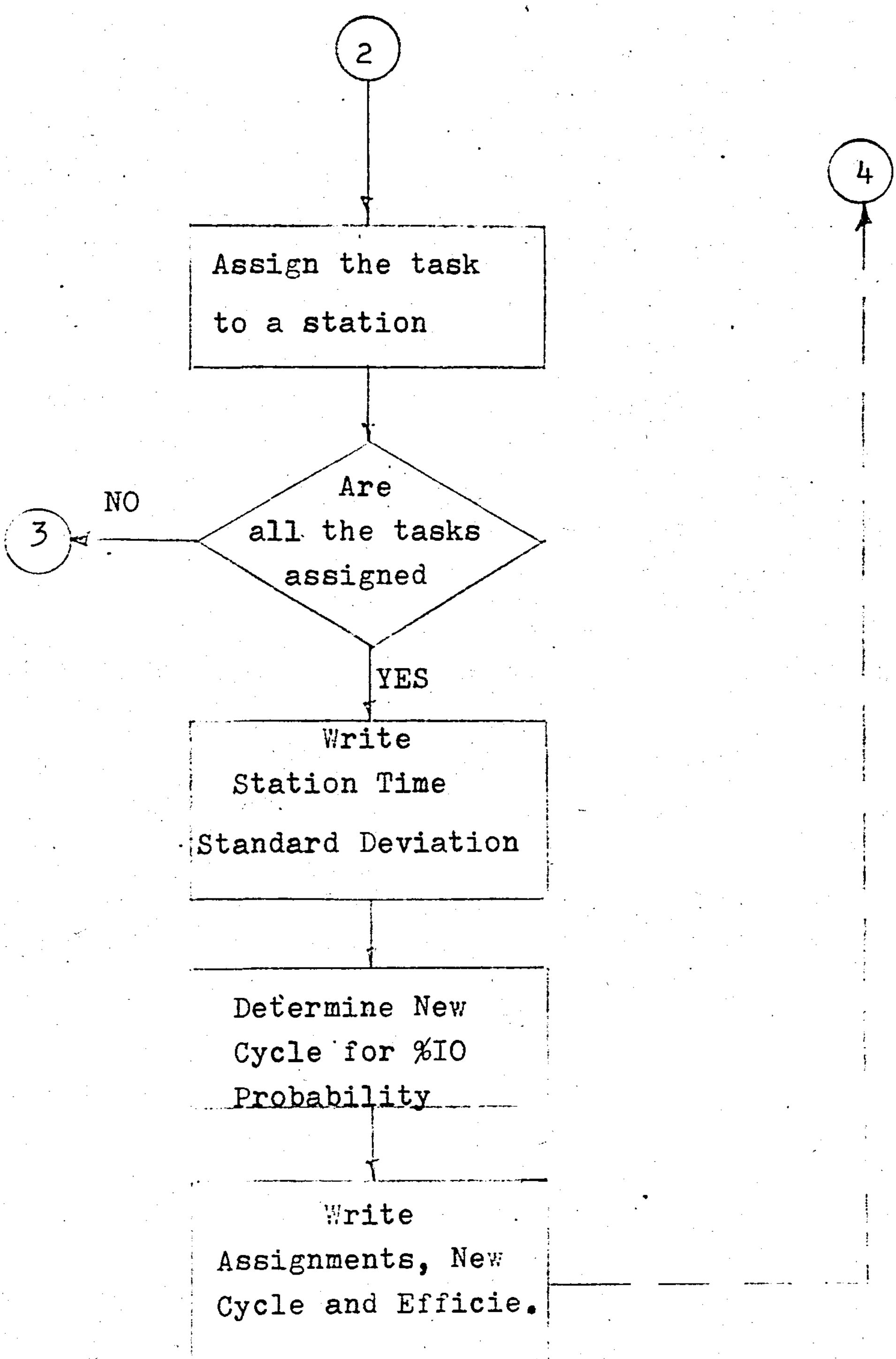


Figure.III.8. Cont.



For line Balancing with stochastic task times several techniques have been developed. Ignall is one of them, stated, to group the tasks into work stations so that the sum of the expected task times does not exceed some specified percentage of the cycle time, for instance % 90. Moodie and Young, and Brennecke also developed techniques.

But the techniques they developed leaves two critical questions unanswered.

i) Up to what percent of the cycle time should work stations be packed?

ii) Should this percentage be the same for all stations?

By consideration of Labour and incompletion costs these questions could be answered profitably.

In order to present the proposed approach as clearly as possible Kottas and Lau have used certain simplifying assumptions.

Assumptions:

i) Only restrictions while assigning tasks are cycle time and precedence relationships.

ii) Each line worker is paid the same wage regardless of his assignments.

iii) A task can only begin if all its predecessors have been completed.

iv) Each task time K is independent of each other and defined by $\mu_K$ and $\sigma_K$.

v) All unfinished tasks are completed off the line for a cost of $I_K$. $I_K$ is not a function of what fraction of the task K was completed on the line.

vi) Appropriate estimate of $\sigma_k$ is used; $s_k$

$$s_k = \left[ \sum_{i=1}^{n} (t_{ik} - \mu_k)^2 / (n-1) \right]^{1/2}$$

where

n- is the number of cycles of task k which where timed

$t_{i_k}$ - is the time for $i^{th}$ one of these cycles.

Starting from the input end, it decides which task should be performed first in a station, which tasks should follow and in what order, and when the station should be closed and a new one begin. It first determines the available tasks for assignment. Then it identifies which of these tasks are marginally desirable to perform next and from this group selects the one to be added on the basis of expected incomplete costs. A station is closed when there are no desirable tasks available.

Preliminary step in this technique is gathering the data as shown (STEP 1)*

i) cycle time,
ii) labour rate,

*STEPS are shown in Flow Chart. See Figure III.10.

Figure III.10. Flow Chart of the Technique.

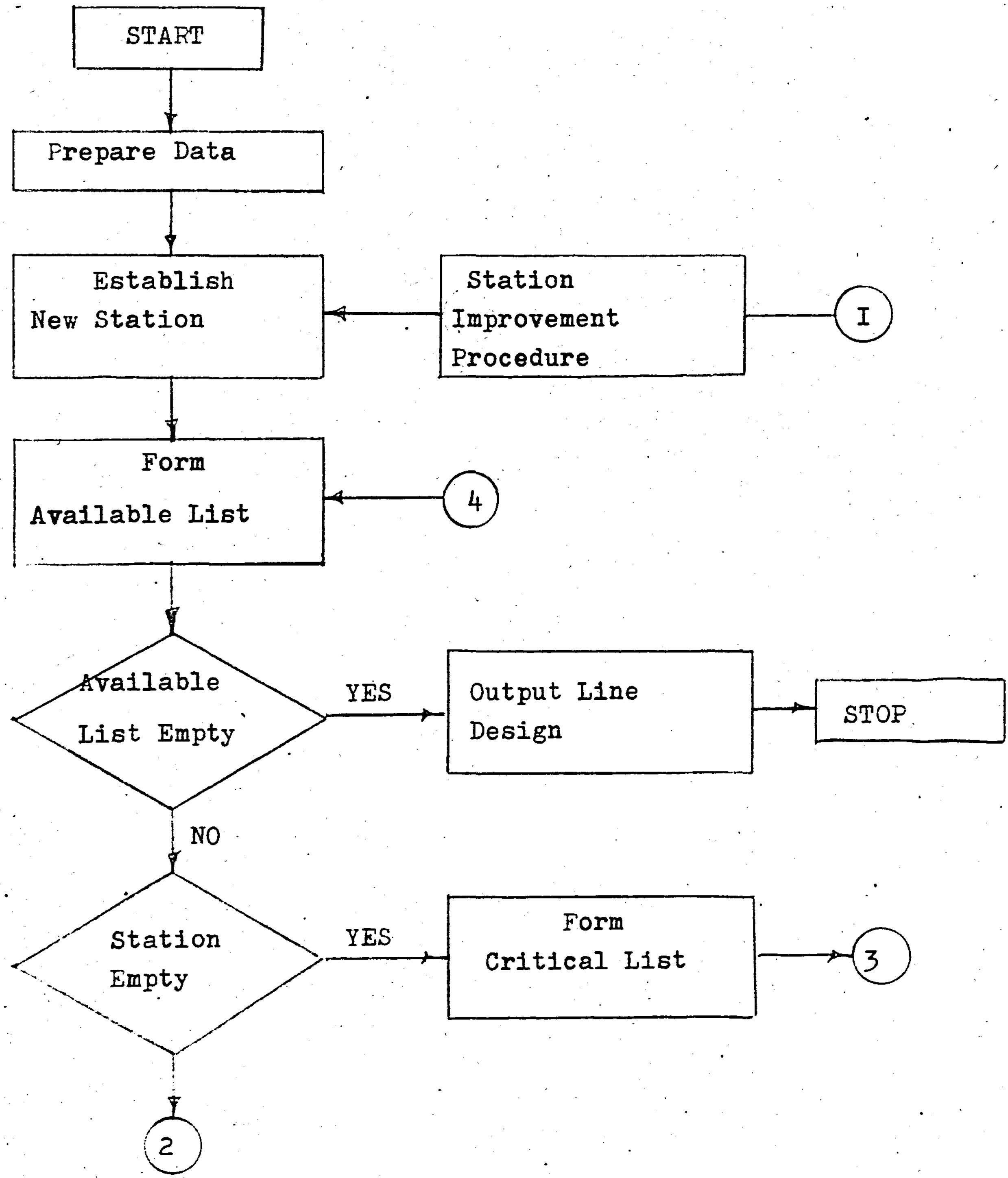
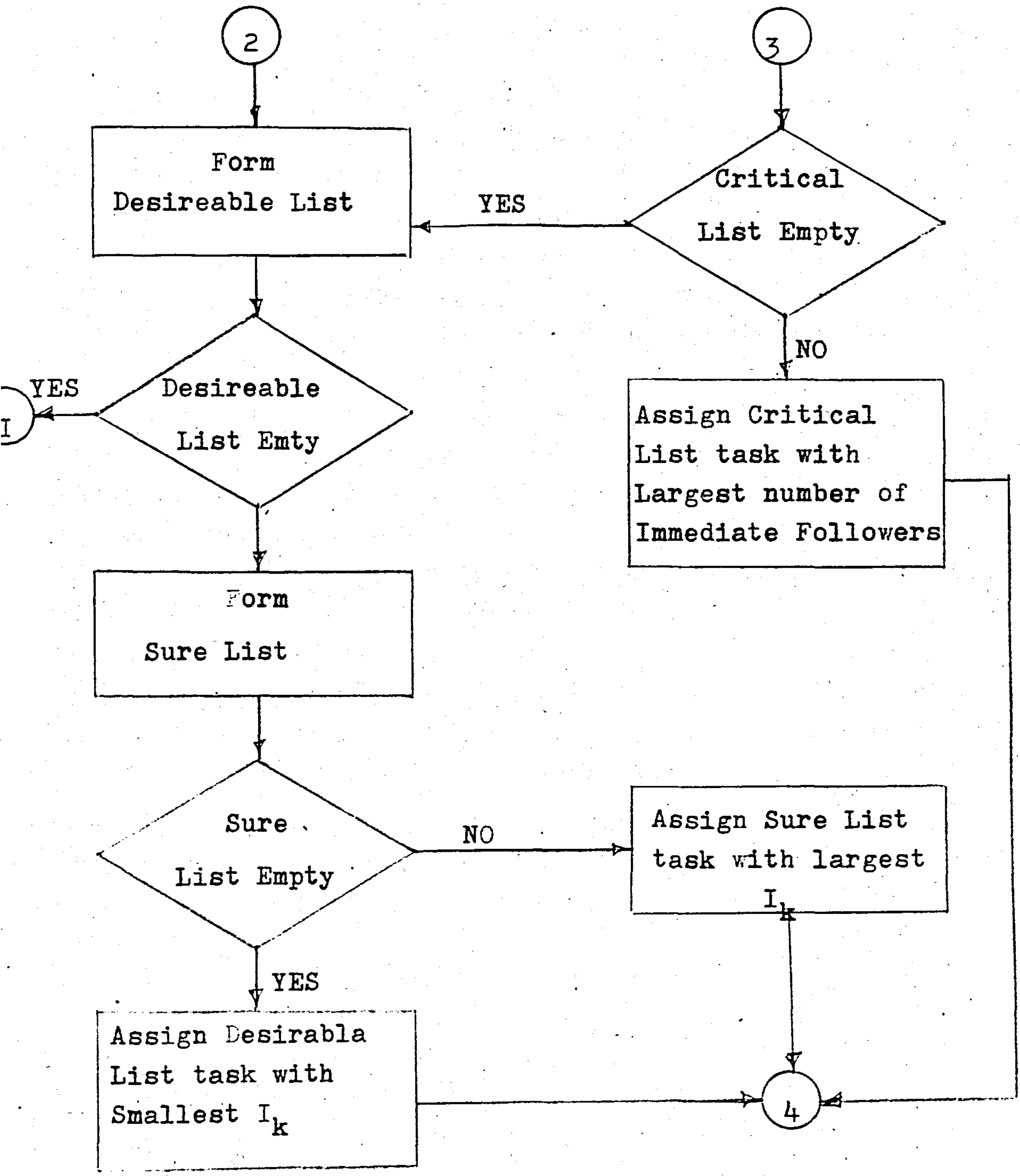


Figure III.IO. Cont.



iii) for each task;

a) $\mu_k$,

b) $\sigma_k$,

c) estimated cost to complete off the line,

d) immediate predecessors,

e) number of immediate followers.

Following step will be the forming of Available List (Step 3) It selects the tasks with no unassigned, immediate predecessors. This list is updated everytime a task is assigned. If we reached the (step 4) - YES the line design is completed.

For our purposes, a task is considered marginally desirable when its anticipated labour savings in the specific position under consideration is equal to or exceeds its expected incompletion cost (step 6) is then forming the desirable list from available list by using the definition of marginality.

The labour savings $L_k$; is determined by how much the labour cost of performing task k in our unestablished station will be reduced by performing it in the already existing station.

The incompletion cost expected as a result of assigning task k next in the station under consideration is the cost $I_k$ stemming from not completing task k in the line multiplied by the probability $p_k$ of not doing so within the cycle time C. $I_k$ is determined by both the cost $I_k'$ of completing task k itself off the line and the cost of finishing all its precedence related followers. For instance

$$I_1 = \sum_{k=1}^{11} I_k' = \$\ 3.05$$

$$I_{11} = I_{11}' = \$\ .20$$

as seen in Table III.11.

Calculation of $P_k$ is easy after all the assumption mentioned in the beginning. Note that task k cannot be completed until all the tasks in $s_k$, the set of task k and the tasks performed before it in the station, have been completed. As the distribution of the time needed to complete all the tasks in $s_k$ is normally distributed with mean $\sum_{i \in s_k} \mu_i$ and standard deviation $(\sum_{i \in s_k} \sigma_i^2)^{1/2}$ .

it follows that

$$P_k = 1 - F(z_k) \tag{1}$$

where $F(z_k)$ is the cumulative density function of normal distributed $z_k$ with $\mu = 0$ and $\sigma = 1$ and

$$z_k = (C - \sum_{i \in s_k} \mu_i)/(\sum_{i \in s_k} \sigma_i^2)^{1/2} \tag{2}$$

From our definition of marginality, the only tasks that will be placed on desirable list are those whose;

$$L_k \geq P_k \cdot I_k.$$

we could state this in terms of $z_k$ by using (1)

$$F(z_k) \geq 1 - L_k/I_k$$

thus

$L_k \geq P_k \cdot I_k$ is equivalent to $z_k \geq z_k^*$ where $z_k^*$ is the value below which the outcomes of a normally distributed random variable with $\mu$=0 and $\sigma$=1 have $1-L_k/I_k$ probability of occurring. In the case of task 1 in table I, $z_1^* = 1.29$. Since 1.29 is the value below which a normally distributed randon variable with mean, 0, and standard deviation, 1, has a

$1-L_1/I_1 = 1-0.3/3.05 = .9017$ probability of occuring.

Before going on to consider which one of the marginally desirable tasks to assign next to a station, it is important to understand the purpose of the concept of marginal desirability. It has two basic functions. First, it removes from consideration for assignment next to the current station those tasks which would move the line towards higher operating costs. Thus it eliminates those tasks which have higher expected incompletion costs than anticipated labour savings, and accepts all others for the desirable list.

Now let us move to the third part of the flow chart. Which one of the marginally desirable tasks to assign? The choice is made on the basis of their $P_k$ and $I_k$ with the goal to minimize the total expected incompleted cost of the station. If there is still a tie then we look at the immediate followers.

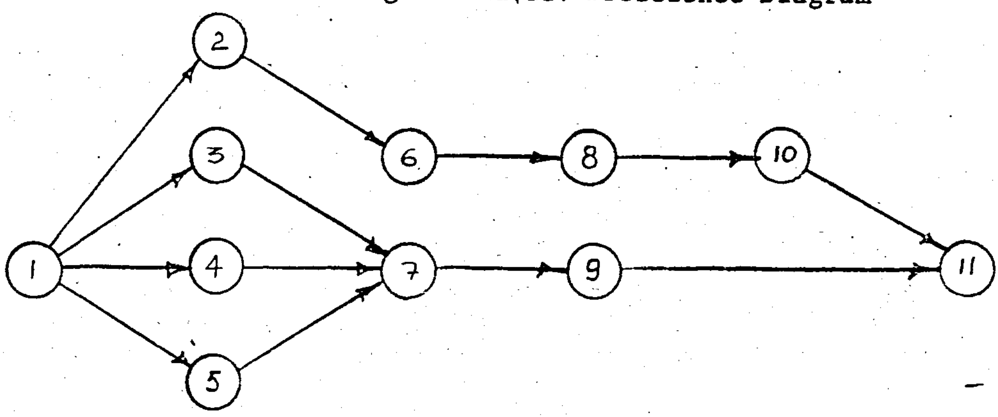
Part of the station can be filled by tasks with virtual certainty of completion. This set of tasks are included in sure list (Step 8). $P_k < 0.005$ or equivalently whose $z_k > 2.575$.

Once however, the station has been filled to the point where none of the desirable list candidates can be assigned without risking incompletion (the sure list is empty, $P_k \geq 0.005$ and $z_k \leq 2.575$) tasks are added on the basis of lowest $I_k$ until there are no more marginally desirable tasks available. (Step 9 - YES, step 11).

These is one more infrequent occurrance to discuss. It could conceivably happen that one or more of the available list tasks would not be marginally desirable to assign to an empty station. Such tasks comprise the ciritical list (STEP 5 - YES STEP 13, STEP 14-NO). Since any critical list task can never be made more desirable than by performing it first in some station, it must be assigned to an empty station.

Example;

Figure III.11. Precedence Diagram




Step 1 is summarised in Table III.11 Cycle time is 20 minutes.

To clarify the use of table III.11 let us examine in more detail the development of the first station.

Task 1 is determined to be the only task on available list (step 3).

(Step 4) No.

(Step 5) Yes. Form critical list (available list tasks whose $z_k < z_k^*$) Hence critical list is empty.

(Step 14) Yes. Form desirable list (Available list tasks whose $z_k \geq z_k^*$) Hence task 1 is in this List.

(Step 7) No. Form sure list. (desirable list tasks whose $z_k > 2.575$) Hence task 1 is in this list.

(Step 9) No. (Step 10) assign the sure list task with largest $I_k$.

Hence it is assigned to station 1 by (step 10) with the assignment of task 1 procedure returns to step 3, to begin another cycle.

| | Sta.I | Sta.II | Sta.III | Sta.IV | Sta.V | Sta.VI |
|---|---|---|---|---|---|---|
| | 1.5 | 2,6,8 | 3,10 | 4 | 7,9 | |
| Total sta. time | 16 | 20 | 18 | 18 | 16 | 6 |

TABLE III.11. Data for the example problem. $L_k$ is based on labor rate of \$3.00/hour Cycle time C is 20 minutes.

| Task | k min/unit | $\sigma_k^2$ | # of imme. pred. | imme. followers | $L_k$ \$/unit | $I'_k$ \$/unit | $I_k$ \$/unit | $Z^*_k$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 12 | 2.4 | 0 | 2,3,4,5 | 0.60 | 0.30 | 3.05 | 1.29 |
| 2 | 4 | 0.8 | 1 | 6 | 0.20 | 0.10 | 1.15 | 1.36 |
| 3 | 8 | 2.0 | 1 | 7 | 0.40 | 0.25 | 1.90 | 0.80 |
| 4 | 18 | 10.0 | 1 | 7 | 0.90 | 0.70 | 2.80 | 0.46 |
| 5 | 4 | 0.8 | 1 | 7 | 0.20 | 0.15 | 1.70 | 1.19 |
| 6 | 4 | 0.8 | 1 | 8 | 0.20 | 0.15 | 2.10 | 1.31 |
| 7 | 6 | 1.2 | 3 | 9 | 0.30 | 0.20 | 1.40 | 0.79 |
| 8 | 12 | 2.4 | 1 | 10 | 0.60 | 0.40 | 1.80 | 0.43 |
| 9 | 10 | 2.0 | 1 | 11 | 0.50 | 0.30 | 1.00 | 0.00 |
| 10 | 10 | 2.0 | 1 | 11 | 0.50 | 0.30 | 1.00 | 0.00 |
| 11 | 6 | 3.6 | 2 | - | 0.30 | 0.20 | 0.40 | -.67 |

### III.2.3. A Preference Order Dynamic Program For SALB(10)

In this technique task times are again stochastic times. It is wanted to assign the tasks to a minimum number of stations, given the cycle time. Dynamic programming Approach is used.

Problem Formulation:

Assume that $t_i$, task times, are independently distributed random variables with respective $F_i$, distribution functions. Let C be the cycle time $A_n$ - be the collection of tasks assisnged to station n, $\tau_n$ - station work content at station n, it is a random variable and

$$\tau_n = \sum_{i \in A_n} t_i.$$

In the formulation we also have,

$P(\tau_n \leq c) \geq \alpha$ for all n, where $\alpha$ is the given lower bound

$0 < \alpha \leq 1$

The definition of feasible set is followed from Held and Karp's Dyanmic Programming Model.

Feasible set is subset of tasks

$S \subset (1,2,\ldots,J)$ in which there exists no pair (i,j) such that

i) $i \notin s$
ii) $j \in s$ and,
iii) i precedes j.

A feasible set will be considered as a state in this Dynamic Programming approach.

The tasks without any predecessors are placed in stage 1 and are considered "marked". An "immediate follower" of a state S is defined as a task which is an immediate successor of at least one of the tasks in S. In stage say k, for State S, the unmarked immediate followers of S constructs the list called F(S). For each $H \subset F(S)$. $H \cup S$ is a state in stage k+1.

When all states in stage k have been considered, each task in F(S), for all S in stage k is marked and the process is repeated for stage k + 1.

When all tasks are marked the construction is done.

When talking about Preference Order Dynamic Program, let T(S) be the optimal return function.

$$T(S) = (n, G_r)$$

n = minimum number of stations needed to accommodate all tasks in S

$G_r$ = distribution function for r, the sum of the task times of tasks assigned to the last station.

Assume a task $e \in S$, where S-e is a feasible set. We let $T(S-e) = (m, G_s)$

m = minimum number of stations needed for S - e

then we define,

$$\begin{aligned} \nabla(T(S-e), e) &= (m, G_{s+e}) && \text{if } G^{-1}_{s+e}(\alpha) \leq c, \\ &= (m+1, F_e) && \text{otherwise.} \end{aligned}$$

What this means is as simple as this; place task e in the last station for S - e if its inclusion does not result in a violation of the probability constraint on the station work content, otherwise create a new station to include e.

Let I be the set of all positive Integers and g be the set of all distribution functions defined on $R^+$ and

$$\varepsilon = \{(n,G) \mid n \in I,\ G \in g\}$$

A preference ordering operator $\perp$ is a mapping from $\varepsilon \times \varepsilon \times \ldots \times \varepsilon$ to $\varepsilon$ with

$$(n_i^*,\ G_i^*) = \perp \{(n_1,G_1),(n_2,G_2),\ldots,(n_q,G_q)\}$$

The operator $\perp$ chooses the i th doublet $(n_i^*,\ G_i^*)$ using the following criterian

$$G^{-1}_{r_i^*+t}(\alpha) \leq G^{-1}_{r_i+t}(\alpha) \quad \text{for all } i \in K \text{ and any } F_t \in g \tag{1}$$

then the optimal return fucntion recursively,

$$T(s) = \underset{\substack{e \in S \\ S-e \text{ feasible}}}{\perp} \{\nabla(T(s-e),e)\}$$

Then for normal variates criterion (1) reduces to;

i* th triplet is selected if
$M_i^* \leq M_i$ and $V_i^* \leq V_i$ for all $i \in K$

when $M_i^* \leq M_i$ and $V_i^* > V_i$ inequality (1) becomes

$$Z_\alpha \leq (M_i - M_i^*)/(V_i^{*1/2} - V_i^{1/2}) \text{ for all } i \in K$$

where $M_i = \mu_{r_i}$ and $V_i = \sigma^2_{r_i}$ in $T(S) = (n,\ \mu_r,\ \sigma_r^2)$

$z_\alpha$ is the 100 $\alpha$ th percentile of the standardized normal distribution.

TABLE III- 12. Example

| Task i | Direct predecessor | $\mu_i$ | $\sigma_i^2$ |
|---|---|---|---|
| 1 | - | 6 | 1.2 |
| 2 | 1 | 2 | 0.4 |
| 3 | 1 | 4 | 1.0 |
| 4 | 1 | 9 | 5.0 |
| 5 | 1 | 2 | 0.4 |
| 6 | 2 | 2 | 0.4 |
| 7 | 3,4,5 | 3 | 0.6 |
| 8 | 6 | 6 | 1.2 |
| 9 | 7 | 5 | 1.0 |
| 10 | 8 | 5 | 1.0 |
| 11 | 9,10 | 3 | 1.8 |

In this illustrative example task times are assumed to be independent normal variates, the cycle time C is set at 10 and the precedence relations are identical to those given in Jackson's well-known example*. The data are shown in Table III.12. states generated and Listed in Table III.13.

If we require that each station there be at least a 90 % chance of completing the work on each unit, then the problem is equivalent to Kottas and Lau's** with $z_{0.90}=1.282$. Procedure yields several optimal solutions, one is;

1,5:4:2,3,6:7,9:8:10:11

* JACKSON,J.R. "A computing Procedure for a Line Balancing Problem" Management Science, Vol.2, No.3, 1956.

**KOTTAS,J.F. and LAU,H.S. - Chapter V.

TABLE III.13. Computation Results for the Illustrative Example

| Stage | Marked Tasks | State S | Unmarked Immediate Followers F (S) | Optimal Decision e | Optimal Return T(S) |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 2,3,4,5 | 1 | 1,6,1.2 |
| 2 | 2,3,4,5 | 1,2 | 6 | 2 | 1,8,1.6 |
| | | 1,3 | | 3 | 2,4,1 |
| | | 1,4 | | 4 | 2,9,5 |
| | | 1,5 | | 5 | 1,8,1.6 |
| | | 1,2,3 | 6 | 3 | 2,4,1 |
| | | 1,2,4 | 6 | 4 | 2,9,5 |
| | | 1,2,5 | 6 | 2,5 | 2,2,0.4 |
| | | 1,3,4 | | 3 | 3,4,1 |
| | | 1,3,5 | | 3 | 2,4,1 |
| | | 1,4,5 | | 4 | 2,9,5 |
| | | 1,2,3,4 | 6 | 3 | 3,4,1 |
| | | 1,2,3,5 | 6 | 2,3,5 | 2,6,1.4 |
| | | 1,2,4,5 | 6 | 2,5 | 3,2,0.4 |
| | | 1,3,4,5 | 7 | 3 | 3,4,1 |
| | | 1,2,3,4,5 | 6,7 | 2,3,5 | 3,6,1.4 |
| 3 | 6,7 | 1,2,6 | 8 | 6 | 2,2,0.4 |
| | | 1,2,3,6 | 8 | 3,6 | 2,6,1.4 |
| | | 1,2,4,6 | 8 | 6 | 3,2,0.4 |
| | | 1,2,5,6 | 8 | 5,6 | 2,4,0.8 |
| | | 1,2,3,4,6 | 8 | 3,6 | 3,6,1.4 |
| | | 1,2,3,5,6 | 8 | 3,5,6 | 2,8,1.8 |
| | | 1,2,4,5,6 | 8 | 5,6 | 3,4,0.8 |
| | | 1,3,4,5,7 | 9 | 7 | 3,7,1.6 |
| | | 1,2,3,4,5,6 | 8 | 3,5,6 | 3,8,1.8 |
| | | 1,2,3,4,5,7 | 9 | 2 | 4,2,0.4 |
| | | 1,2,3,4,5,6,7 | 8,9 | 7 | 4,3,0.6 |
| 4 | 8,9 | 1,2,6,8 | 10 | 8 | 2,8,1.6 |
| | | 1,2,3,6,8 | 10 | 3 | 3,4,1 |
| | | 1,2,4,6,8 | 10 | 8 | 3,8,1.6 |
| | | 1,2,5,6,8 | 10 | 5 | 3,2,0.4 |
| | | 1,2,3,4,6,8 | 10 | 3 | 4,4,1 |
| | | 1,2,3,5,6,8 | 10 | 8 | 3,6,1.2 |
| | | 1,2,4,5,6,8 | 10 | 5 | 4,2,0.4 |
| | | 1,3,4,5,7,9 | | 9 | 4,5,1 |
| | | 1,2,3,4,5,6,8 | 10 | 8 | 4,6,1.2 |
| | | 1,2,3,4,5,7,9 | | 2,9 | 4,7,1.4 |
| | | 1,2,3,4,5,6,7,8 | 10 | 7 | 5,3,0.6 |
| | | 1,2,3,4,5,6,7,9 | | 9 | 4,8,1.6 |
| | | 1,2,3,4,5,6,7,8,9 | 10 | 8 | 5,6,1.2 |
| | | 1,2,6,8,10 | | 10 | 3,5,1 |
| | | 1,2,3,6,8,10 | | 3 | 4,4,1 |
| | | 1,2,4,6,8,10 | | 10 | 4,5,1 |

TABLE III.13. Continued.

| Stage | Marked Tasks | State S | Unmarked Immediate Followers F (S) | Optimal Decision e | Optimal Return T (S) |
|---|---|---|---|---|---|
| | | 1,2,5,6,8,10 | | 5,10 | 3,7,1.4 |
| | | 1,2,3,4,6,8,10 | | 3 | 5,4,1 |
| | | 1,2,3,5,6,8,10 | | 3 | 4,4,1 |
| | | 1,2,4,5,6,8,10 | | 5,10 | 4,7,1.4 |
| | | 1,2,3,4,5,6,8,10 | | 3 | 5,4,1 |
| | | 1,2,3,4,5,6,7,8,10 | | 7 | 5,7,1.6 |
| | | 1,2,3,4,5,6,7,8,9,10 | 11 | 9,10 | 6,5,1 |
| | | 1,2,3,4,5,6,7,8,9,10,11 | | 11 | 7,3,1.8 |

## III.3. The Proposed Algorithm for Assembly Line Balancing

Let $t_{max}$ be the largest task time in the precedence diagram. As usual, C is the cycle time. First part of the procedure is about comparing the C against the $t_{max}$. As known from previous chapters $C \geq t_i$ for all i. Next the check is made for following inequalities.

$$1. \; t_{max} \leq C \leq 2. \; t_{max}$$

$$2. \; t_{max} \leq C \leq 3. \; t_{max}$$

$$3. \; t_{max} \leq C \leq 4. \; t_{max}$$

$$\vdots \qquad \vdots \quad \vdots$$

If first one is satisfied then first entry to any station will be the largest candidate available and following entries will be smallest candidates. If second inequality is satisfied then the first two entires to any station will be the largest candidate and followed by the smallest candidates. Then the general form is;

$$(n) \; . \; t_{max} \leq C \leq (n+1) \; . \; t_{max} \qquad \text{for} \quad n = 1,2,3,...$$

which ever value of n satisfies the inequality, proceed as following;

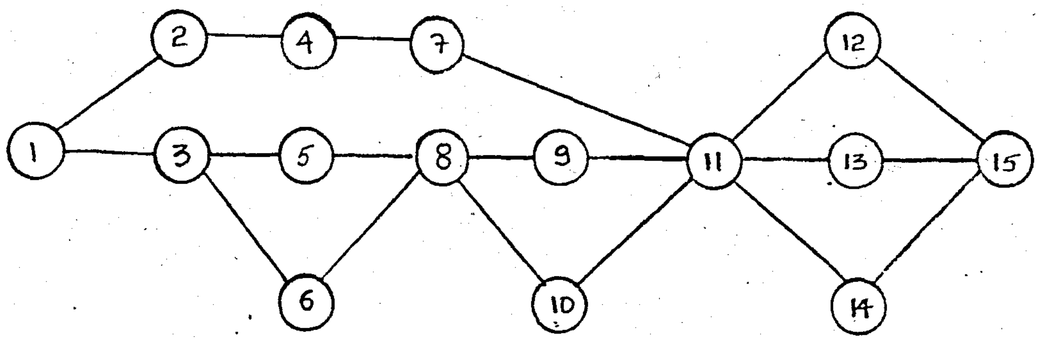
i) First n entries to any station will be the largest candidates available,

ii) Following entries will be the smallest candididates till the idle time is minimized for that station with the available candidates.

Example:

Suppose we are given an assembly line with 15 tasks with deterministic task times. Also we supplied with the precedence relations. Cycle time is given as 18 minutes.

Figure III.12. Precedence Diagram for the Example Problem




C = 18 $t_{max}$ = 9 from table III.14

$2.t_{max} \leq C \leq 3.t_{max}$ n = 2

TABLE III.14. Data for the Example

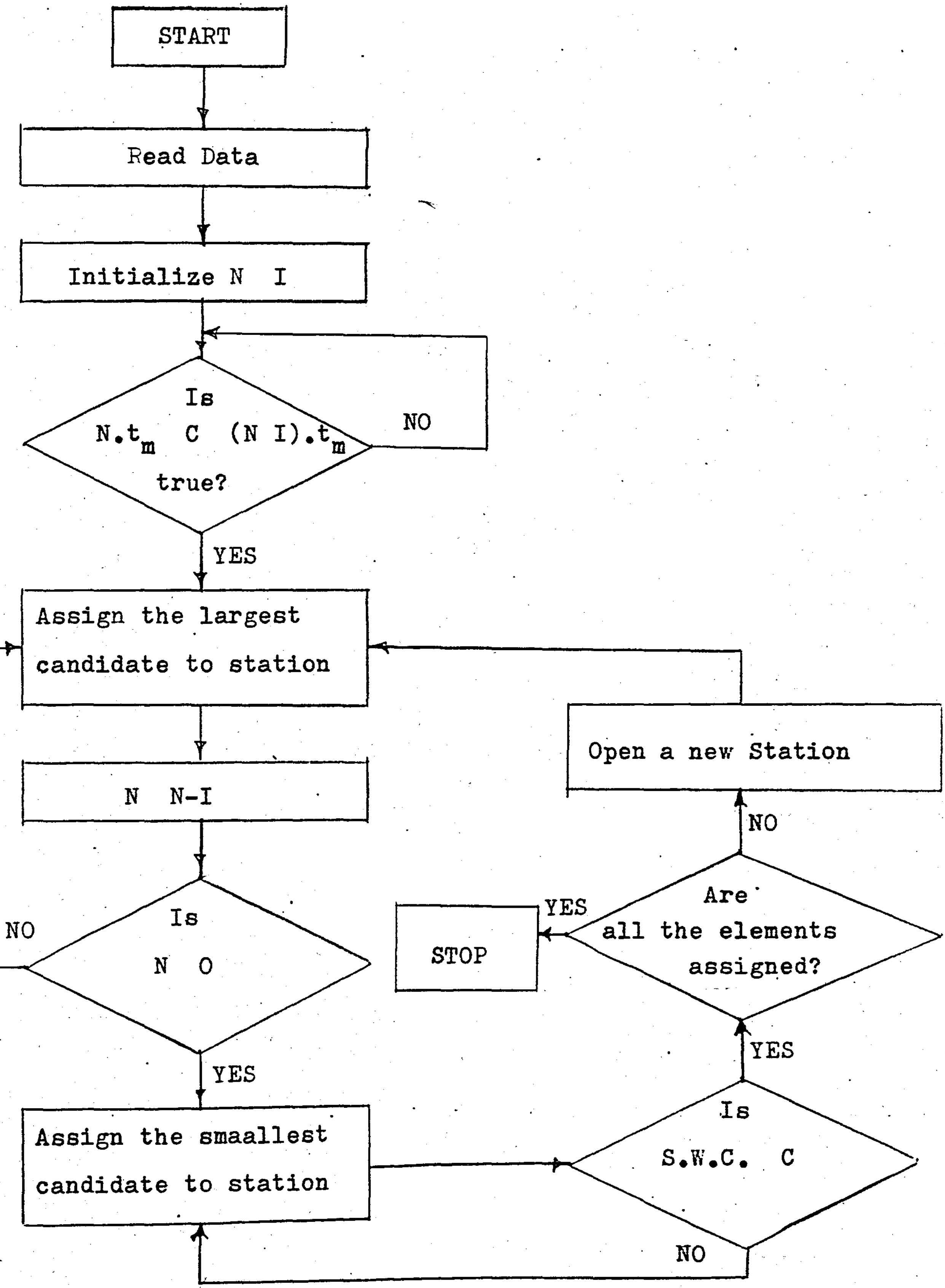
| Tasks | Times | Predeccesors |
|---|---|---|
| 1 | 5 | - |
| 2 | 7 | 1 |
| 3 | 8 | 1 |
| 4 | 2 | 2 |
| 5 | 3 | 3 |
| 6 | 9 | 3 |
| 7 | 6 | 4,5 |
| 8 | 5 | 5,6 |
| 9 | 4 | 8 |
| 10 | 5 | 8 |
| 11 | 7 | 7,9,10 |
| 12 | 7 | 11 |
| 13 | 3 | 11 |
| 14 | 2 | 11 |
| 15 | 8 | 12,13,14 |

The proposed algorithm yields a Balance loss of 10 % with the following solution;

TABLE III.15. Solution to the Problem

| Station | Elements | S.W.C. |
|---|---|---|
| I | 1,3,5 | 16 |
| II | 2,6,4 | 18 |
| III | 7,8,9 | 15 |
| IV | 10,11,13,14 | 17 |
| V | 12,15 | 15 |

Figure III.I3. Flow Chart of the Proposed Algorithm



# CHAPTER IV

# ASSEMBLY LINE BALANCING PROBLEM AT ÖZKÖSEOĞLU ISI SANAYİİ VE TİCARET A.Ş.

Özköseoğlu Isı Sanayii ve Ticaret A.Ş. as a member of Özköseoğlu group of companies, has been established in 1962 and highly participated Turkish Industrial Development up to date. Holding the licences of worlds well known companies, Özköseoğlu Isı Sanayii ve Ticaret A.Ş. manufactures and supplies all kinds of Industrial furnaces, burners, painting and drying systems, lime kilns, petrochemical plants, cement plants and complete automatic bread factories and has a good name on heat technology. During the last years she has extended her services to the international plane.

## IV.1. Description of the Problem

Inside the Özköseoğlu Plant at Yenibosna there is an assemly line in order to assemble the 34 parts of ELM 50 Flour Sifting and Lossening Machine. The parts list of ELM 50 is given in table IV.1. And ports are shown in Figure IV.1.

## IV.2. Required Data Versus Obtained Data

Before going into the data obtained from Özköseoğlu A.Ş., once more, information required for line balancing needs to be pointed out in order to achieve a good balance. It is first necessary to obtain certain data from various sources whether it is assembly or fabrication. The following minimum information is necessary in any case:

i) Production Volume

ii) List of operations and their sequence

iii) Times required to complete each operation as well as the elemental time values

The production volume should be determined by the sales or marketing group. The list of operations and their sequence should be established prior to considering the line-balancing problem; (by Research and Development Department together with Planning Department) otherwise the problem becomes excessively complex.

The following data has been obtained as described above plus the help of foreman of the ELM-50 Department and from observations in the factory.

List of subassemblies and precedence diagram of these subassemblies are given in table IV.2 and figure IV.1 respectively.

Figure 1. ELM-50 SIFTING and LOOSENING MACHINE

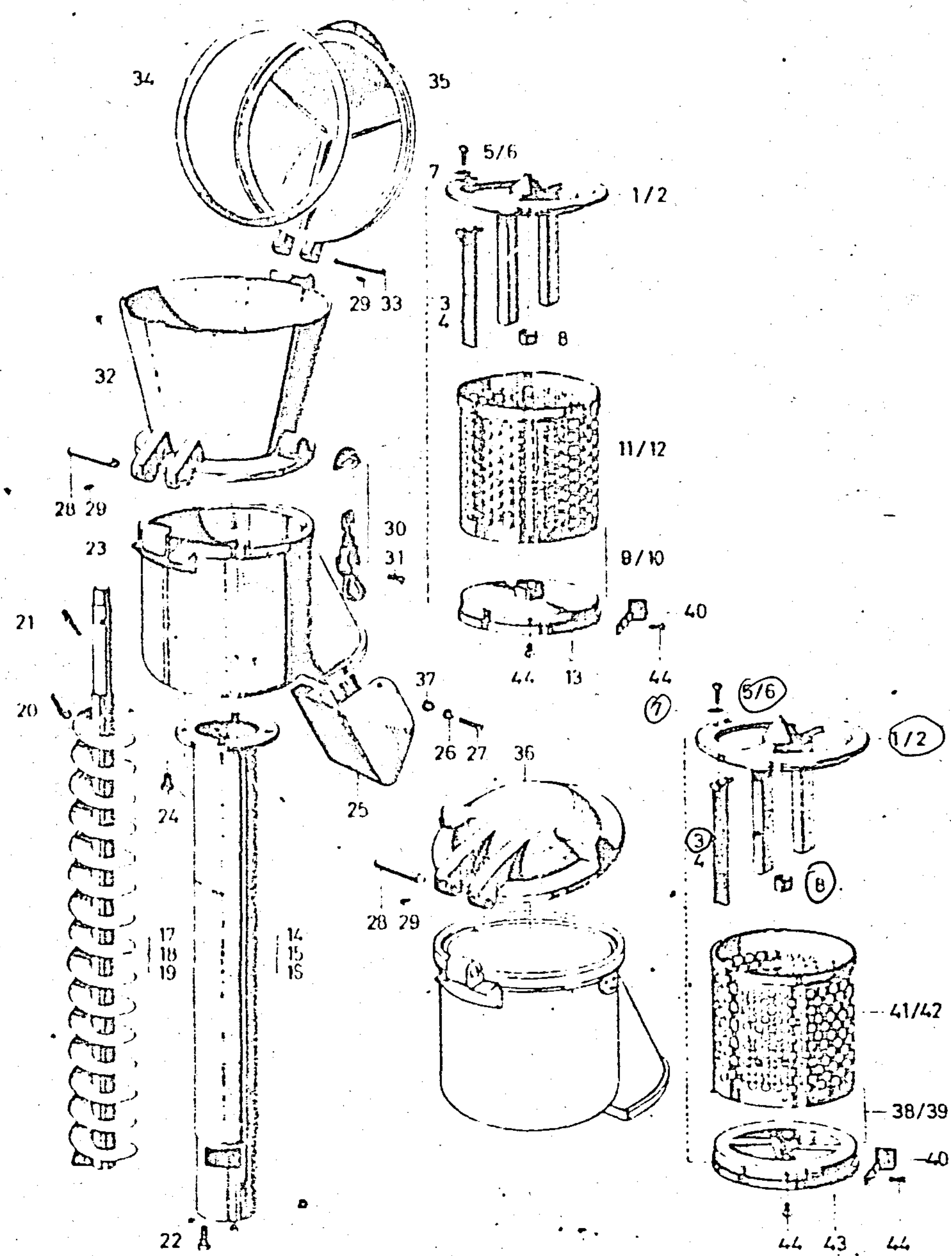


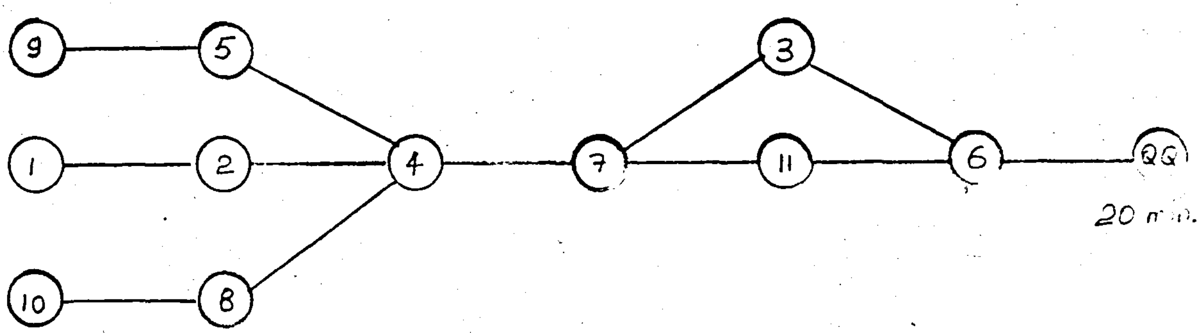
TABLE IV.1. Parts List for ELM-50 SIFTING MACHINE

| Part No | Part Name | Note |
|---|---|---|
| 1 | 3-prong | Steel |
| 3 | Screw-on prong | Steel |
| 5 | Inside Hexagonal Screw | M5x10 |
| 6 | Inside Hexagonal Screw | M5x10 |
| 7 | Spring-Washer | 5 |
| 8 | Ball bearing | SKF 6203-2RS |
| 9 | Grating Drum Steel | Grating Steel Drum base-alu. |
| 10 | Grating drum, stainless | Grating drum, Base alu., Grating cylinder |
| 11 | Grating cylinder with reinforcement | Steel |
| 12 | Grating cylinder with reinforcement | V2A |
| 13 | Drum-base | |
| 20 | Socket groved pin | 8Øq32 |
| 21 | Cylindrical pin | 10Øx50 |
| 22 | Inside Hexagonal Screw | M8x25 |
| 23 | Sifting drum housing | |
| 24 | Inside hexagonal screw | M10x20 |
| 25 | Delivery chute | |
| 26 | Washer | 6 |
| 27 | Round-headed screw | M6x20 |
| 28 | Bolt | 12Øx82 |
| 29 | Threaded pin | M6x10 |
| 30 | Fastening device complete | Brass |
| 31 | Oval-headed countersunk screw | M5x10 |
| 32 | Top feed-hopper | |
| 33 | Bolt | 10Øx100 |
| 34 | Jointing | 2x306/282 Ø |
| 35 | Lid | |
| 36 | Lid | |
| 37 | Nut | |
| 38 | Sifting drum, steel | Sifting drum base-alu. |
| 40 | Scraper | |
| 42 | Sifting cylinder with reinfor. | V2A |
| 43 | Drum-base | |
| 44 | Round-headed screw | M5x10 |

TABLE IV.2. List of Subassemblies

| Subassembly Number | Times (min) | Part Numbers |
|---|---|---|
| S.Ass.1. | 15 | 1,3,5,6 |
| S.Ass.2. | 22 | 38,40,42,43,44 |
| S.Ass.3. | 12 | 25,26,27,37 |
| S.Ass.4. | 14 | 28,29,36 |
| S.Ass.5. | 17 | 7,8,9,10 |
| S.Ass.6. | 12 | 30,31 |
| S.Ass.7. | 14 | 32,23 |
| S.Ass.8. | 6 | 24,22 |
| S.Ass.9. | 15 | 11,12,13 |
| S.Ass.10. | 11 | 20,21 |
| S.Ass.11 | 14 | 33,34,35 |

Figure IV.2. Precedence Diagram for ELM 50




Since the yearly demand for this ELM-50 FLOUR SIFTING and LOOSENING MACHINE is given as 2000 units from the Marketing Department, we need to produce one every hour for 255 working days in a year.

255 . 8 = 2040 units/year.

Then the cycle time for this assembly line balancing problem is dictated as 60 minutes by the demand. In the following sections this line will be balanced by using five of the algorithms mentioned in chapter III.

## IV.3. Solution by Using ARCUS' Technique

The solution by this most commonly used technique is summarized in table IV.3. These results will be discussed with the results of next section at the end of this chapter.

Table IV.3. Solutions Generated by Arcus' Technique

Sequence Number 1 Cycle Time 60.000

| Statien Number | | | | | Station Time | Slack |
|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 2 | 8 | 54.000 | 6.000 |
| 2 | 9 | 5 | 4 | 7 | 60.000 | .000 |
| 3 | 11 | 3 | 6 | 12 | 58.000 | 2.000 |

Total Slack Time = 8.000 Efficiency = 95.35 percent

Sequence Number 2 Cycle Time 60.000

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 2 | 10 | 0 | 54.000 | 6.000 |
| 3 | 9 | 5 | 4 | 7 | 60.000 | .000 |
| 4 | 3 | 11 | 6 | 12 | 58.000 | 2.000 |

Total Slack Time = 14.000 Efficiency = 91.86 percent

## IV.4. Solution by Using Proposed Algorithm

The $t_{max}$ in this problem is 22 minutes from table IV.2 and the cycle time is 60 minutes. Then the check is made for n;

for n=2, $n.t_{max} \leq C \leq (n+1) . t_{max}$ is

satisfied. Then it follows,

i) First (n=2) entries to any station will be the largest candidate available;

ii) Following entries will be the smallest candidates till the idle time is minimized for the station under consideration with the available candidates.

Solution by using these is given in Table IV.4.

TABLE IV.4. Solution by Proposed Algorithm

| n | Available Tasks (Feasible Tasks) | Largest/Smallest Task | Assigned Task | S.W.C. (Cumulative) |
|---|---|---|---|---|
| 1 | 9,1,10 | t(1), t(9) | t(1) =15 | 15 |
| 2 | 9,2,10 | t(2) | t(2) =22 | 37 |
| 3 | 9,10 | t(10) | t(10)=11 | 48 |
| 4 | 9,8 | t(8) | t(8) =6 | 54 |
| 5 | 9 | t(9) | t(9) =15 | 69 |

STOP 69 > 60

STATION NUMBER 1 = 1,2,10,8 with S.W.C. 54

| n | Available Tasks (Feasible Tasks) | Largest/Smallest Task | Assigned Task | S.W.C. (Cumulative) |
|---|---|---|---|---|
| 1 | 9 | t(9) | t(9) =15 | 15 |
| 2 | 5 | t(5) | t(5) =17 | 32 |
| 3 | 4 | t(4) | t(4) =14 | 46 |
| 4 | 7 | t(7) | t(7) =14 | 60 |

STOP 60 $\geq$ 60

STATION NUMBER 2 = 9,5,4,7 with S.W.C. 60

TABLE IV.4. (Cont.)

| n | Available Tasks (Feasible Tasks) | Largest/Smallest Task | Assigned Task | S.W.C. (Cumulative) |
|---|---|---|---|---|
| 1 | 3,11 | t(11) | t(11)=14 | 14 |
| 2 | 3 | t(3) | t(3)=12 | 26 |
| 3 | 6 | t(6) | t(6)=12 | 38 |
| 4 | Q.Q. | t(Q.Q.) | t(Q.Q.)=20 | 58 |

STOP ALL THE ELEMENTS ARE ASSIGNED.

STATION NUMBER 3 = 11,3,6,Q.Q with S.W.C. 58

The above solution yields a balance efficiency of; 95.5 %.

## IV.5. Solution by Using Kilbridge and Wester Technique

The solution by this Kilbridge and Wester technique, which was explained in detail in section III.1.1., is summarized in following manner;

| Station number | 1 | 2 | 3 |
|---|---|---|---|
| Elements | 1,9,10,5 | 2,8,4,7 | 11,3,6,QQ |
| Station Work Content | 58 | 56 | 58 |

Above solution yields an efficiency of 95.35 %.

## IV.6. Solution by Using Ranked Positional Weights

The Assembly line of ELM-50 Sifting Machine of Özköseoğlu Isı Sanayi A.Ş. was also tackled by using Rawked Positional Weights. Allocation of tasks to stations are;

| Station Number | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Elements | 1,9,2 | 5,10,8,4 | 7,11,3,6 | QQ |
| Station Work Content | 52 | 48 | 52 | 20 |

Which yields a balancing efficiency of 71.67 %.

## IV.7. Solution by Using COMSOAL Technique

An alternative solution was generated for the problem in hand by making use of COMSOAL technique. This approach gave the same efficiency and solution as Ranked Positional Weights approach.

| Station Number | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Elements | 1,2,9 | 5,10,8,4 | 7,11,3,6 | QQ |
| Station Work Content | 52 | 48 | 52 | 20 |

Balance Efficiency = 71.67 %

## IV.8. Proposed Algorithm Versus ARCUS', R.P.W., Kilbridge and Wester and COMSOAL

The Arcus' technique together with Kılbridge and Wester technique gave a balancing efficiency of 95.35 %, while COMSOAL and R.P.W was giving the same efficiency of 71.67 %.

The Proposed algorithm has proved its better performance against R.P.W. and COMSOAL by achieving the same efficiency with ARCUS' and K ξ W's algorithms. On the other hand the Proposed Algorithm has shown its quickness while achieving this efficiency compared with two of these better performed techniques, namely Arcus' and K ξ W' techniques.

| | The Proposed Algorithm | Kilbridge and Wester | Arcus' | R.P.W. | COMSOAL |
|---|---|---|---|---|---|
| % | 95.35 | 95.35 | 95.35 | 71.67 | 71.67 |

# CHAPTER V
# CONCLUSIONS

In the introduction of this thesis line-balancing problem was introduced with different approaches as, simulation, queueing theory, etc., and it was pointed out that these approaches were only able to give insight to the structure of the problem. Then in following chapter basic concepts and measure of effectiveness of the line-balancing process were defined. The exact model was presented by using Mixed-Integer Programming. At the end of the chapter, the difficulties of finding a solution to such a formulation was mentioned. Chapter III was solely devoted to a comprehensive literature survey of Assembly-Line-Balancing Techniques. Starting from the simplest and the earliest algorithms, the most mechanized and computerized algorithms are covered and reviewed by means of hypothetical examples. Flow charts for three of the algorithms and computer programs in FORTRAN IV are written and run for sample problems, which are given in Appendix as scheduling tools. Thus one of the objectives have been achieved by giving quick references to the readers and practitioners of production planning and control function in industry. Using the content of chapters II and III. one can evaluate and select the most appropriate algorithm to his problem.

The basic contribution of this thesis is the Proposed Algorithm which was developed during this research. This Proposed

Algorithm offors manual technique to achieve balances with reasonable level of efficiency by savings in computation time. This way practitioners can tackle large scale balancing problems with a minimum amount of theoretical background.

The Proposed Algorithm is also applied to a real case taken from a manufacturing company which makes Industrial Furnaces, in order to compare its performance against the most commonly used techniques as ARCUS', Kilbridge and Wester, Ranked Positional Weights and COMSOAL techniques. This case study was taken from the assembly line of ELM-50 Flour Sifting and Loosening Machine at ÖZKÖSEOĞLU ISI SANAYİİ ve TİCARET A.Ş. Detailed explanation of the problem was covered in Chapter IV. It was observed that for this size of problem, the Proposed Algorithm performed equally efficient, with ARCUS' and K&W's techniques which also gave the best efficiency among the rast of the techniques. Finally, the study and analytical discussion of existing line-balancing algorithms and the development of the Proposed Algorithm is giving the structure of this topic for the future researchers.

# REFERENCES


1- ARCUS,A.L., "COMSOAL-A Computer Method of Sequencing Operations for Assembly Lines" International Journal of Production Research, 4, 4 (1966).

2- BRENNECKE,D., "Two Parameter Assembly Line Balancing Model", Models and Analysis for Production Management, M.P. Hottenstein, Ed., International Textbook Co., Scranton Pens., 1968.

3- CARSON, BOLZ, AND YOUNG, "Production Handbook", The Ronald Press Company, New York, 1972.

4- CONRAD,R., "Setting the Pace", Medical Research Council, APU-232-55, London: Applied Psychology Research Unit, 1955.

5- GUTJAHR,A.L. AND NEMHAUSER,G.L., "An Algorithm for the Line Balancing Problem", Management Science, Vol.11, No 2, 1964.

6- HELD,M. AND KARP,R.M., "A Dynamic Programming Approach to Sequencing Problems", J.Soc.Indust.Appl.Math., 10, 1962.

7- HOTTENSTEIN,M.P., "A Review of Assembly Line Balancing", International Textbook Co., Screnton, Pennsylvania, 1968.

8- HUNT,G.C., "Segmential Arrays of Waiting Lines", Operations Research, 4, 1956.

9- JACKSON,J.R., "A Computing Procedure for a Line Balancing Problem", Management Science, Vol.2, No 3, 1956.

10- KAO,P.C., "A Preference Order Dynamic Program for Stochastic Assembly Line Balancing", Management Science, Vol. 22, No 10, June 1976.

11- KILBRIDGE,M.D. and WESTER,L., "A Heuristic Method of Assembly Line Balancing", Journal of Industrial Engineering, XII, 4 (1961).

12- KOTTAS,F.J. and LAU,H.S., "A Cost-Oriented Approach t. Stochastic Line Balancing", AIIE Transactions, Vol.5, No 2, 1973.

13- MASTOR,A.A., "An Experimental Investigation an Comparative Evlauation of Production Line Balancing Techniques", Management Science, Vol.16, No 11, July, 1970.

14- MIZE,J.H., WHITE,C.R. and BROOKS,G.H., "Operations Planning and Control", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

15- REEVE,N.R. and THOMAS,W.H., "Balancing Stochastic Assembly Lines", AIIE Transactions, Sept. 1973.

16- SPHICAS,G.P., SILVERMAN,F.N.: "Deterministic Equivalents for Stochastic Assembly Line Balancing", AIIE Transactions, Vol.8, Dec., 1975.

17- TAYLOR,B.W., "Addressing the Production Balancing Problem" Virginia Polytechnic Int. and State University, Blocksburg, Va 24061.

18- THOMOPOULOS,N.T., "Mixed Model Line Balancing with Smoothed Station Assignments", Management Science, Vol. 16, No 9, May, 1970.

19- WILD,R., "Mass-Production Management", Management Center, University of Bradford, Wiley, 1971.

20- ZIMMERMANN,H.J. and SOVEREIGN,M.G., "Quantitative Models for Production Management", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

# APPENDIX - I
# TERMINOLOGY IN ASSEMBLY LINE BALANCING

-A-

Assembly Bill of Materials; a Bill of Materials formatted to show all of the components required in an assembly operation.

Assembly Line; a sequence of operations with a common cycle time through which a unit of product is assembled. The arrangement of machines, equipments material and workers which permits the work in process to progress sequentially from one operation to another until the product (or product component) has been assembled.

-B-

Balance; the degree to which the station times of each operation in an assembly line approach the cycle time.

Batch Manufacturing; used in similar-process industries. Products manufactured are kept in different lots as they pass through the process. They are placed in these lots or batches, according to size, type, colour, issue, thickness, etc. Exactly right number, or at least a certain minimum of each must be made.

-C-

Christmas Tree; a graphic product structure chart showing how the assembly is made up of subassemblies, the subassemblies made up of lower level components, etc.

Critical Path; the sequence of jobs or activities, in a Network Analysis project, such that the total duration of the project is equal to the sum of the durations of individual jobs in the sequence. There is no time delay or slack in an activity along the critical path.

Cycle; the complete sequence of activities, operations and machines or process times required to complete one segment, unit, or batch of work.

-D-

Delay (Balance Delay); the idle time of one or more operations in a series due to imperfect balancing.

-E-

Element time; the time to perform a given element. May refer to the observed, average, selected, normal or standard time.

-F-

Final Assembly; the highest or "zero level" assembled product. Frequently used as a name for the manufacturing department where the product is assembled.

Flow Line; the path along which men or material travel in progressing through the plant.

-G-

GANTT Chart; a graphic representation on a time scale of the current relationship between actual and planned performance.

-H-

Heuristics; general method used to solve problems which defy solution by standard techniques. Examples are: "Sequencing" and "Line-balancing" problems. Literally heuristics means "serving to find out and encouraging further investigation." Techniques lead to solutions by trying "commonsense" rules and procedures rather than rigorous optimality criteria.

-I-

Idle Time; time which has been scheduled, when operators or machines are not producing because of lack of material, tooling etc. Part of process where one member is waiting for another member to complete his task.

Intermittent Production; a production system in which jobs pass through functional departments in lots.

-J-

Job Shop; an intermittent type manufacturing plant. the term job shop is frequently used to indicate a make-to-order plant but this is not correct usage.

-L-

Line-balancing; a technique for determining the product mix that can be run down an assembly line at the planned line rate. For example, if an automative assembly line happened to be scheduled one day with nothing but convertibles, some workers would be standing idle while others would not be able to keep pace with the line. This is an attempt to make the work time at each station as close to the cycle time for the product as possible.

Loads; this is the amount of scheduled work ahead of a manufacturing facility, usually expressed in terms of hours of work.

-M-

Manufacturing Bill of Materials; the bill of materials organized into a form that is useful to the manufacturing departments. An engineering bill, for example, might show a simple parts list of all of the components that make up a particular assembly, where the manufacturing bill of materials might be shown in indented form so that the levels of components can easily be identified.

Manufacturing Cycle; the amount of time that is required from the penetration of a manufacturing order to the completion of the order.

Master Parts List; the authoritative parts list from which all other format variations and copies are derived.

Mechanization; the act of process of using power-driven machinery to perform specific operations or functions usually with the intent or improving productivity and/or quality on the work performed.

-N-

Network Analysis; a technique of analysis useful in planning a project that consists of showing the sequence of activities, and their interrelationship within a network of activities making up a project. By computing the cumulative time for each part or path through the network, from the starting event to terminal event, the extent or cost of the Critical Path is determined.

-O-

Operation List; history or progress through the cycle of operations list of operations sequences, much like route sheet.

-P-

Parts List; a listing of all the parts that go into a product usually in Product Summary format. A tabulation of all the parts included in any unit to be manufactured.

PERT; (Program Evaluation and Review Technique) this is a project planning technique like the Critical Path Method but more sophisticated since it involves obtaining a pessimistic, most likely, and optimistic time for each activity from which the most likely completion time for the project along the critical path is computed.

Predetermined Time System; a organized body of information procedures and techniques employed in the study and evaluation of manual work elements. The system is expressed in terms of the motions used, their general and specific nature, the conditions under which they occur, and their previously determined performance times.

Production Cycle; the lead time to produce a product.

Production Rates; the quantity of production usually expressed in units, hours, or some other broad measure.

Productivity; 1. the ratio of output to total inputs.
2. the ratio of actual production to standard production applicable to either an individual worker or a group of workers.

-Q-

QUEUE; a sequence of elements, one waiting behind the other. In other words, a waiting line.

Queueing Theory; the mathematical theory relating the way in which elements arrive into a queue and are serviced. Deals with the problem of providing adequate service facilities to handle an arriving stream of things, or people, requiring service of some kind.

-S-

Safety factor; a constant which is multiplied by the standard deviation of forecast during the lead time and which is chosen to provide a particular level of service and total safety stock.

Scheduling; deciding the precise use of manufacturing facilities at each instant of time.

Sequencing; determining the order in which a manufacturing facility is to process a number of different jobs in order to achieve certain objectives.

Slack time; the amount of time between the scheduled due date for a job and the estimated completion date. If the job is to be completed ahead of schedule, it is said to have slack time; otherwise, negative slack time.

Standart time; a unit time value for the accomplishment of a work task as determined by the proper application of appropriate work measurement techniques. Generally established by applying appropriate allowance to normal time.

Subassembly; an assembly which is used at a higher level to make up another assembly.

-W-

Wait Time; (Dead time) the time that a job spends waiting to be moved or waiting to be worked on in the shop.

Workplace; (Work station), a specific area, usually in a fixed, defined location, used for the performance of a work task including auxiliary area for machinery and materials.

# APPENDIX - II

## FORTRAN IV LISTING AND THE OUTPUT OF HOFFMANN'S ALGORITHM

A-II/1
(79)

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890**SEQ NO. 06
```

HOFFMANN'S

```
EEEEEEEEEEEE     TTTTTTTTTTTT      UU          UU      DDDDDDDDDD
EEEEEEEEEEEE     TTTTTTTTTTTT      UU          UU      DDDDDDDDDDD
EE                    TT           UU          UU      DD          DD
EE                    TT           UU          UU      DD          DD
EE                    TT           UU          UU      DD          DD
EEEEEEEE              TT           UU          UU      DD          DD
EEEEEEEE              TT           UU          UU      DD          DD
EE                    TT           UU          UU      DD          DD
EE                    TT           UU          UU      DD          DD
EE                    TT           UUU        UUU      DD          DD
EEEEEEEEEEEE          TT            UUUUUUUUUU         DDDDDDDDDDD
EEEEEEEEEEEE          TT              UUUUUUUU         DDDDDDDDDD
```

```
* * *  UNIVAC 1106 -- BOGAZICI  UNIVERSITESI  KOMPUTER  MERKEZI --ISTANBUL  VER. 33R3/BU7-1   SITE *  BU1106    TURKIYE  * * *

  RUNID * ETUD          USER ID *                PART NUMBER * 00      INPUT DEVICE * CR2          OUTPUT DEVICE * PR2

    FILE NAME * PRA000ETUD               CREATED AT:    12:00:01 MAY 02,1980              PRINTED AT:   12:04:15 MAY 02,1980

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890**SEQ NO. 06
ΔRUN,E ETUD,114-14-214, HESABI
ΔFTN,IS
FTN 7R1-1 05/02/80-12:00
          DIMENSION MATRIX(16,16),KODE(15,16),JOBS(30),STIME(50),
         1NCLEM(14),NELEM(14),ETIME(16),NAME(10)
          WRITE(6,501)
      501 FORMAT (1H1)
      301 LO=15
          MO= 0
          NO= 0
          READ (5,13) NAME
       13 FORMAT (10A4)
      302 WRITE (6,1002) NAME
     1002 FORMAT(,0,,,ASSEMBLY LINE BALANCE FOR,,5X,10A4,//)
          DO 200 N = 1, 16
1         DO 200 M = 1, 16
2     200 MATRIX(M,N) = 0
          DO 105 K = 1,30,2
1         READ (5,5) JOBS(K), JOBS(K+1)
1         IF (JOBS(K+1)) 105,10,105
1     105 MO=MO+2
        5 FORMAT (2I3)
       10 TETIME = 0.
          DO 106 K = 1,16
1         READ (5,12) ETIME(K)
1      12 FORMAT (F5.3,4X,I2)
1         IF (ETIME(K)) 106,1,106
1     106 NO = NO+1
        1 DO 92 I=1,MO,2
1         IF (JOBS(I)) 8,92,8
1       8 N = JOBS(I+1)
1         L=JOBS(I)
1         MATRIX(L,N) = 1
1      92 CONTINUE
          DO 9 N=1,NO
1       9 TETIME = TETIME + ETIME(N)
          MO = MO/2
          WRITE (6,111) TETIME, NO,MO
      111 FORMAT(, ,,,TOTAL ELEMENT TIME =,,F8.3,3X,,NUMBER OF ELEMENTS IS,,
         1 I4,3X,36HNUMBER OF PRECEDENCE RESTRICTIONS IS,I5)
       45 MSUM=0
          IREST = 1
          ISLACK = 0
```

A-II/2
(80)

```
        READ (5,12) CTIME, INVERT
        IF(CTIME) 300,97,300
    300 DO 201 N=1,LO
1       DO 201 M=1,NO
2   201 KODE(N,M)=0
        LOX1 =LO-1
        DO 204 N=1,LOX1
1       STIME(N) = 0.
1       NELEM(N) =0
1   204 NCLEM(N) = 0
        IF (INVERT) 209,210,209
    209 NI = (NO-1)/2
        NZ = 2
        ND = NO-1
        NO1 = NO + 1
        DO 206 I = 1,NI
1       DO 207 J = NZ, ND
2       MS = MATRIX(I,J)
2       NO1J = NO1-J
2       NO1I = NO1-I
2       MATRIX(I,J) = MATRIX(NO1J,NO1I)
2   207 MATRIX(NO1J,NO1I) = MS
1       ND = ND-1
1   206 NZ = NZ+1
        DO 213 I = 1, NI
1       ETS = ETIME(I)
1       NO1I = NO1-I
1       ETIME(I) = ETIME(NO1I)
1   213 ETIME(NO1I) = ETS
        GO TO (210,45),IREST
    210 DO 6 L =1,NO
1       DO 7 N=1,NO
2     7 MSUM = MSUM + MATRIX(N,L)
1       KODE(1,L) = MSUM
1     6 MSUM = 0
        NOP = 1
        DO 4 N=1,NO
1     4 MATRIX(N,N) = 16000
     11 WRITE (6,41) CTIME
     41 FORMAT (,1,,33X,,ASSEMBLY LINE BALANCE,/5X,,STATION,,20X,
       117HCYCLE TIME NUMBER,F7.4,21X,5HSLACK/5X,8H NUMBER 63X,4HTIME//)
     33 STIME(1) = CTIME
        STIME(LO) = CTIME
        I=1
        N=1
     18 DO 14 J=N,NO
1       IF (KODE(I,J))14,16,14
1    16 IF (STIME(I)-ETIME(J)) 14,17,17
1    17 DO 20 K=1,NO
2    20 KODE(I+1,K) = KODE(I,K)-MATRIX(J,K)
1       NELEM(I)=J
1       STIME(I+1)=STIME(I)-ETIME(J)
1       I=I+1
1    14 CONTINUE
        IF (I-1) 96,31,15
     15 IF(STIME(LO)-STIME(I))21,22,23
     23 DO 24 J=1,NO
1    24 KODE(LO,J)=KODE(I,J)
        STIME(LO)=STIME(I)
        LOX1 = LO-1
        DO 25 K= 1,LOX1
```

A-II/3
(81)

```
1        25 NCLEM(K) = NELEM(K)
         21 NELEM(I) = 0
         22 I = I-1
            N = NELEM(I) + 1
            GO TO 18
         31 IF (STIME(LO)-CTIME) 26,99,97
         26 DO 28 J = 1,NO
1        28 KODE(1,J) = KODE(LO,J)
            IF(INVERT) 211,212,211
        211 DO 214 K= 1,14
1           IF (NCLEM(K)) 208,214,208
1       208 NCLEM(K) = NO1-NCLEM(K)
1       214 CONTINUE
        212 WRITE (6,30)  NOP,(NCLEM(K),K=1,14),STIME(LO)
         30 FORMAT(I10,I10,13I4,F8.4)
            NOP = NOP + 1
            TSLACK = TSLACK + STIME(LO)
            GO TO 33
         96 WRITE (6,110) STIME(LO)
        110 FORMAT (, ,,,ERROR, I IS MINUS,STIME,,F6.4)
         97 STOP
         99 EFF = (1.0-(TSLACK/TETIME))*100.0
            WRITE (6,3) TSLACK, EFF
          3 FORMAT (1H /5X,23HTOTAL SLACK TIME NUMBER F7.4,24X,
           118HEFFICIENCY NUMBER F6.2,9H PER CENT//)
            DO 2 N=1,NO
1         2 MATRIX(N,N) = 0
            IF (INVERT) 215,45,215
        215 IREST = 2
            GO TO 209
            END

END FTN 472 IBANK 891 DBANK
ΔXQT
MAP28R2 72R1U1 05/02/80 12:01:16

ADDRESS LIMITS     001000 033057     13360 IBANK WORDS DECIMAL
                   040000 044357      2288 DBANK WORDS DECIMAL
STARTING ADDRESS   032130

              SEGMENT $MAIN$          001000 033057     040000 044357

TABLE$/SYS72          $(1)    001000 001177
F2FARR                $(1)    001200 002114
F2ACTIV$              $(1)    002115 002131    $(2)     040000 040002
                      $(3)    002132 002146
                      $(5)    002147 002147
F2TABX                                         $(2)     040003 040314
F2RTRN$                                        $(2)     040315 040315
F2FCA                                          $(0)     040316 040322
F2FRT                                          $(034)   040323 040364
F2CDCD$                                        $(2)     040365 040432
F2OUT                 $(1)    002150 005151
FDASC$/SYS73R1Q8      $(1)    005152 005346
F2INP                 $(1)    005347 007524
F2NMLT                $(1)    007525 012315
F2NFMT                $(1)    012316 012644
```

A-II/
(82

```
F2SAR                          $(1)     012645 013002
F2FMT                          $(1)     013003 015125
C2ERR                          $(1)     015126 016562
C2DSDF                         $(1)     016563 017540
C2ANSI                         $(1)     017541 024106
C2SSDF                         $(1)     024107 026425
F2CLOSE                        $(1)     026426 026457     $(0)     040433 040435
PMD$COM(COMMONBLOCK)                                               040436 040436
MOERO$(COMMONBLOCK)                                                040437 040442
F2CON                                                     $(2)     040443 042164
                                                          $(034)   MOERO$
                                                          $(036)   PMD$COM
F2SCT                                                     $(2)     042165 042436
ERU$/SYS73R1
FORCOM$/FORFT4                                            $(2)     042437 042444
CERU$
F2EXIT                                                    $(2)     042445 042500
F2FIM                          $(1)     026460 032127
F2INIT                                                    $(0)     042501 042501
                                                          $(2)     042502 042564
NAME$                          $(1)     032130 033057     $(0)     042565 044004
                                                          $(4)     044005 044206
                                                          $(6)     044207 044207
                                                          $(010)   044210 044355
                                                          $(012)   044356 044357

SYS$*RLIB$. LEVEL 73R1
END MAP
```

A-II/5 (83)

ASSEMBLY LINE BALANCE FOR KARAYALCIN ALPASLAN

TOTAL ELEMENT TIME = 2.462 NUMBER OF ELEMENTS IS 16

NUMBER OF PRECEDENCE RESTRICTIONS IS 15

A-II/6 (84)

```
                              ASSEMBLY LINE BALANCE
   STATION                   CYCLE TIME NUMBER  .9010                        SLACK
   NUMBER                                                                    TIME

      1         16  15  14  13   0   0   0   0   0   0   0   0   0   0    .0000
      2         12  11  10   9   8   7   6   0   0   0   0   0   0   0    .0180
      3          5   4   3   2   1   0   0   0   0   0   0   0   0   0    .2230
   TOTAL SLACK TIME NUMBER  .2410                    EFFICIENCY NUMBER  90.21 PER CENT


 FTN ERR ON UNIT-5       ATTEMPTED TO READ PAST AN END-OF-FILE
 ERR MODE  ERR-TYPE: 03     ERR-CODE: 00
 ERROR ADDRESS:  031650     BDI:  000004
 ER EABT$       ABORT ADR: 042137  BDI:200005
 PROGRAM INITIATED INTERUPT: EABT$.

   X                      000000 027064  000000 027065  000000 000000  000000 040333  000000 044672  000000 044646  000000 044634
        000000 042165     000000 044630  000000 044124  000000 032313  400000 000000  020001 000040  000000 000007  000000 000000

   A    400000 000000     020001 000040  000000 000007  000000 000000  000000 005347  000000 000001  000000 000012  143000 000000
        175402 030447     000000 000000  000000 000021  000000 000000  000000 000000  000000 000000  000000 000000  000001 000000
        000220 777714     462200 000000

   R                      000000 000000  777777 777776  777777 777776  777777 777776  777777 777776  000000 010001  000001 004311
        000010 000000     001750 000000  000000 000000  000000 000000  000000 000000  000000 000000  000000 000000  000000 042165
  ERR$  TYPE 03 CODE 00 CONT 12       REENT ADR:  041213 BDI: 200005
 USER DID ER ERR$.

   X                      000000 027064  000000 027065  000000 000000  000000 040333  000000 044672  000000 044646  000000 044634
        000000 042165     000000 044630  000000 044124  000000 032313  400000 000000  020001 000040  000000 000007  000000 000000

   A    400000 000000     020001 000040  000000 000007  000000 000000  000000 005347  000000 000001  000000 000012  143000 000000
        175402 030447     000000 000000  000000 000021  000000 000000  000000 000000  000000 000000  000000 000000  000001 000000
        000220 777714     462200 000000

   R                      000000 000000  777777 777776  777777 777776  777777 777776  777777 777776  000000 010001  000001 004311
        000010 000000     001750 000000  000000 000000  000000 000000  000000 000000  000000 000000  000000 000000  000000 042165
 RUNSTREAM ANALYSIS TERMINATED
```

A-II/7
(85)

```
RUNID: ETUD      ACCT: 114-14-214      PROJECT: HESABI
   ETUD     ABORT
TIME:     TOTAL: 00:00:43.568      CBSUPS:  000002041
          CPU:   00:00:11.097      I/O:  00:00:21.516
          CC/ER: 00:00:10.954      WAIT: 00:00:00.050
SUAS USED:        58.28TL   SUAS REMAINING:      0.00TL
ABOVE CHARGE CALCULATED AT FOLLOWING RATES -
1 CBSUP              = 0.02TL
1 CARD READ          = 0.05TL
1 CARD PUNCHED       = 0.40TL
1 PAGE PRINTED       = 1.50TL
1 TAPE I/O MINUTE    = 1.50TL
IMAGES  READ:   169       PAGES:  7
START:    12:00:01 MAY 02,1980     FIN:  12:02:15 MAY 02,1980
```

# APPENDIX - III

## FORTRAN IV LISTING AND THE OUTPUT OF ARCUS' ALGORTHM

A.III/1
(87)

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890**SEQ NO. 02
      AAAAAAAA      LL             CCCCCCCC       00000000     LL            EEEEEEEEEEEE
     AAAAAAAAAA     LL            CCCCCCCCCC     0000000000    LL            EEEEEEEEEEEE
    AA      AA      LL           CC        CC    00      00    LL            EE
    AA      AA      LL           CC        CC    00      00    LL            EE
    AA      AA      LL           CC              00      00    LL            EE
    AAAAAAAAAAAA    LL           CC              00      00    LL            EEEEEEE
    AAAAAAAAAAAA    LL           CC              00      00    LL            EEEEEEE
    AA      AA      LL           CC              00      00    LL            EE
    AA      AA      LL           CC        CC    00      00    LL            EE
    AA      AA      LL           CC        CC    00      00    LL            EE
    AA      AA      LLLLLLLLLLLL  CCCCCCCCCC     0000000000    LLLLLLLLLLLL  EEEEEEEEEEEE
    AA      AA      LLLLLLLLLLLL   CCCCCCCC       00000000     LLLLLLLLLLLL  EEEEEEEEEEEE
*  *  *  UNIVAC 1106 -- BOGAZICI  UNIVERSITESI  KOMPUTER  MERKEZI --ISTANBUL  VER.  33R3/BU7     SITE * 01106     TURKIYE  *  *  *

   RUNID * ALCOLE        USER ID *                 PART NUMBER * 00      INPUT DEVICE * CR2          OUTPUT DEVICE * PR2

   FILE NAME * PRA000ALCOLE            CREATED AT:   11:38:09 JUL 01,1980            PRINTED AT:   11:55:04 JUL 01,1980

123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890**SEQ NO. 02
ΔRUN,E ALCOLE,111-15-216,ALB,10,100
ΔFTN,IS
FTN 7R1-1 07/01/80-11:38
                    DIMENSION MATRIX(16,16),TIME(16),NUNIT(50,14),STATIM(50),
                   1SLAK(50),NPREC1(16),NPREC2(16),JOBS(30),NAME(10),NLIST(20)
                    DO 9901 I=1,30
  1            9901 JOBS(I)=0
                    DO 9902 I=1,50
  1                 STATIM(I)=0.0
  1                 SLAK(I)=0.0
  1                 DO 9902 J=1,14
  2            9902 NUNIT(I,J)=0
                    XI=0.123456
                    NCOUNT=1
                    MO=0
                    NO=0
                    READ(5,913) NAME
               913  FORMAT(10A4)
               302  WRITE(6,1002) NAME
              1002  FORMAT(' ','ASSEMBLY LINE BALANCING FOR    ',5X,10A4,//)
                    DO 200 M=1,16
  1                 TIME(M)=0.0
  1                 NPREC1(M)=0
  1                 NPREC2(M)=0
  1                 DO 200 N=1,16
  2             200 MATRIX(M,N)=0
                    DO 105 K=1,30,2
  1                 READ(5,5) JOBS(K),JOBS(K+1)
  1                 IF(JOBS(K+1))105,10,105
  1             105 MO=MO+2
                  5 FORMAT(2I3)
                 10 DO 92 I=1,MO,2
  1                 IF(JOBS(I)) 8,92,8
  1               8 L=JOBS(I+1)
  1                 N=JOBS(I)
  1                 MATRIX(N,L)=1
  1              92 CONTINUE
                    DO 113 K=1,16
  1                 READ(5,912) TIME(K)
  1                 IF(TIME(K))113,1,113
  1             113 NO=NO+1
                912 FORMAT(F5.3)
                  1 NTASK=NO
```

C55OK.

```
          BTASK=NO
          DO 98 J=1,NTASK
1         TETIME=0.0
1         DO 5054 I=1,NTASK
2    5054 TETIME=TIME(I)+TETIME
1         NPREC1(J)=0
1         DO 98 I=1,NTASK
2      98 NPREC1(J)=MATRIX(I,J)+NPREC1(J)
       22 READ(5,23) CYCLE,NRUNS
       23 FORMAT(F5.3,3X,I3)
          IF(CYCLE) 24,25,24
       24 M=0
          NO=0
          KBEFOR=100
          NCOUNT=1
          SMALL=0.0
          DO 1001 NX=1,NRUNS
1      41 DO 95 J=1,NTASK
2      95 NPREC2(J)=NPREC1(J)
1         M=0
1         STATIM(I)=0.0
1         K=1
1         WRITE(6,46) NCOUNT
1      46 FORMAT(, ,,, SEQUENCE NUMBER  ,,I3)
1         WRITE(6,26) CYCLE
1      26 FORMAT(, ,,6X,7HSTATION,23X,13HCYCLE TIME    ,F6.3,20X,7HSTATION,
1        14X,5HSLACK,/,6X,6HNUMBER,64X,4HTIME,/)
1         NO=0
1      13 XII=XI* .23
1         IXJ=XII*100.
1         XIJ=IXJ
1         XI=(XII-XIJ/100.)*100.
1         IXI=XI*100.
1         XIK=IXI
1         XKK=XI-(XI-XIK/100.)
1         NUMA=0
1         DO 2031 I=1,NTASK
2         IF(NPREC2(I)) 2031,2032,2031
2    2032 IF(NUMA-20) 5061,2031,2031
2    5061 NUMA=NUMA+1
2         NLIST(NUMA)=I
2    2031 CONTINUE
1         ANUM=NUMA
1         NUMB=(XKK*ANUM)+1.0
1         NRAND=NLIST(NUMB)
1         IF(XI) 2022,2022,45
1    2022 WRITE(6,2021) NRAND
1    2021 FORMAT(, ,,, NRAND,,I5)
1         GO TO 25
1      45 IF(NRAND-NTASK) 12,12,13
1      12 IF(NPREC2(NRAND)) 13,14,13
1      14 STATIM(K)=STATIM(K)+TIME(NRAND)
1         IF(STATIM(K)-CYCLE) 27,27,28
1      27 M=M+1
1         NUNIT(K,M)=NRAND
1         SLAK(K)=CYCLE-STATIM(K)
1         GO TO 29
1      28 STATIM(K)=STATIM(K)-TIME(NRAND)
1         SLAK(K)=CYCLE-STATIM(K)
1         K=K+1
1         STATIM(K)=TIME(NRAND)
```

A.III/3
(89)

```
1        SLAK(K)=CYCLE-STATIM(K)
1        M=1
1        NUNIT(K,M)=NRAND
1     29 NPREC2(NRAND)=-1
1        DO 91 J=1,NTASK
2     91 NPREC2(J)=NPREC2(J)-MATRIX(NRAND,J)
1        NO=NO+1
1        IF(NO-NTASK) 13,31,13
1     31 NCOUNT=NCOUNT+1
1        DO 93 I=1,K
2     93 WRITE(6,36) I,(NUNIT(I,M),M=1,14),STATIM(I),SLAK(I)
1     36 FORMAT(2I10,13I4,F10.3,F9.3)
1        TSLACK=0.0
1        DO 5055 I=1,K
2   5055 TSLACK=SLAK(I)+TSLACK
1        EFF=(1.0-(TSLACK/TETIME))*100.
1        WRITE(6,5053) TSLACK,EFF
1   5053 FORMAT(1H ,/,5X,18HTOTAL SLACK TIME = ,F6.3,24X,13HEFFICIENCY =
1       1F6.2,9H PER CENT,//)
1        DO 1009 I=1,K
2        DO 1009 J=1,14
3   1009 NUNIT(I,J)=0
1   1001 CONTINUE
         WRITE(6,500)
     500 FORMAT(,1,)
         GO TO 22
      25 STOP
         END

END FTN 417 IBANK 1410 DBANK
ΔXQT
MAP28R2 72R1U1 07/01/80 11:38:23

ADDRESS LIMITS      001000 032770      13305 IBANK WORDS DECIMAL
                    040000 045366       2807 DBANK WORDS DECIMAL
STARTING ADDRESS    032130

              SEGMENT $MAIN$          001000 032770      040000 045366

TABLE$/SYS72           $(1)    001000 001177
F2FARR                 $(1)    001200 002114
F2ACTIV$               $(1)    002115 002131      $(2)     040000 040002
                       $(3)    002132 002146
                       $(5)    002147 002147
F2TABX                                            $(2)     040003 040314
F2RTRN$                                           $(2)     040315 040315
F2FCA                                             $(0)     040316 040322
F2FRT                                             $(034)   040323 040364
F2CDCD$                                           $(2)     040365 040432
F2OUT                  $(1)    002150 005151
FDASC$/SYS73R1Q8       $(1)    005152 005346
F2INP                  $(1)    005347 007524
F2NMLT                 $(1)    007525 012315
F2NFMT                 $(1)    012316 012644
F2SAR                  $(1)    012645 013002
F2FMT                  $(1)    013003 015125
C2ERR                  $(1)    015126 016562
```

```
C2DSDF                    $(1)   016563 017540
C2ANSI                    $(1)   017541 024106
C2SSDF                    $(1)   024107 026425
F2CLOSE                   $(1)   026426 026457   $(0)     040433 040435
PMD$COM(COMMONBLOCK)                                      040436 040436
MOERO$(COMMONBLOCK)                                       040437 040442
F2CON                                            $(2)     040443 042164
                                                 $(034)   MOERO$
                                                 $(036)   PMD$COM
F2SCT                                            $(2)     042165 042436
ERU$/SYS73R1
FORCOM$/FORFTN                                   $(2)     042437 042444
CERU$
F2EXIT                                           $(2)     042445 042500
F2FIM                     $(1)   026460 032127
F2INIT                                           $(0)     042501 042501
                                                 $(2)     042502 042564
NAME$                     $(1)   032130 032770   $(0)     042565 045036
                                                 $(4)     045037 045246
                                                 $(6)     045247 045247
                                                 $(010)   045250 045361
                                                 $(012)   045362 045366

SYS$*RLIB$. LEVEL 73R1
END MAP
```

ASSEMBLY LINE BALANCING FOR KARAYALCIN ALPASLAN

SEQUENCE NUMBER 1

CYCLE TIME .901

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 5 | 1 | 2 | 3 | 10 | 6 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | .867 | .034 |
| 2 | 8 | 11 | 4 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .694 | .207 |
| 3 | 13 | 14 | 15 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .901 | .000 |

TOTAL SLACK TIME = .241 EFFICIENCY = 90.21 PER CENT

SEQUENCE NUMBER 2

CYCLE TIME .901

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .867 | .034 |
| 2 | 6 | 9 | 5 | 10 | 1 | 7 | 8 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | .697 | .204 |
| 3 | 3 | 11 | 4 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .864 | .037 |
| 4 | 13 | 14 | 15 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .901 | .000 |

TOTAL SLACK TIME = .275 EFFICIENCY = 88.83 PER CENT

SEQUENCE NUMBER 3

CYCLE TIME .901

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .867 | .034 |
| 2 | 5 | 1 | 9 | 10 | 6 | 2 | 7 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | .867 | .034 |
| 3 | 4 | 8 | 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .694 | .207 |
| 4 | 13 | 14 | 15 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | .901 | .000 |

TOTAL SLACK TIME = .275 EFFICIENCY = 88.83 PER CENT

# APPENDIX - IV

## SOLUTION TO THE CASE STUDY BY USING ARCUS' ALGORITHM

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901**SE  NO. 05

        AAAAAAAA      LL                 CCCCCCCC      00000000      LL              EEEEEEEEEEEE
       AAAAAAAAAA     LL                CCCCCCCCCC    0000000000     LL              EEEEEEEEEEEE
      AA        AA    LL               CC        CC   00        00   LL              EE
      AA        AA    LL               CC        CC   00        00   LL              EE
      AA        AA    LL               CC             00        00   LL              EE
      AAAAAAAAAAAA    LL               CC             00        00   LL              EEEEEEEE
      AAAAAAAAAAAA    LL               CC             00        00   LL              EEEEEEEE
      AA        AA    LL               CC             00        00   LL              EE
      AA        AA    LL               CC        CC   00        00   LL              EE
      AA        AA    LL               CC        CC   00        00   LL              EE
      AA        AA    LLLLLLLLLLLL      CCCCCCCCCC    0000000000     LLLLLLLLLLLL    EEEEEEEEEEEE
      AA        AA    LLLLLLLLLLLL       CCCCCCCC      00000000      LLLLLLLLLLLL    EEEEEEEEEEEE
*  *  *  UNIVAC 1106 -- BOGAZICI UNIVERSITESI KOMPUTER MERKEZI --ISTANBUL VER. 33R3/BU9-7  SITE * BU1106   TURKIYE   *  *

  RUNID * ALCOLE        USER ID *              PART NUMBER * 00     INPUT DEVICE * CR2         OUTPUT DEVICE * PR2

    FILE NAME * PRΔ000ALCOLE          CREATED AT:  14:33:00 JUL 14,1981            PRINTED AT:  15:48:24 JUL 14,198

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901**SE  NO. 05
ΔRUN,E ALCOLE,111-15-216,ALB,4
ΔFTN,IS
FTN 8R1  *07/14/81-14:33
                  DIMENSION MATRIX(12,12),TIME(12),NUNIT(50,14),STATIM(50)
                  DIMENSION SLAK(50),NPREC1(12),NPREC2(12),JOBS(30)
                  DIMENSION NAME(10),NLIST(20)
                  DO 9901 I=1,30
  1          9901 JOBS(I)=0
                  DO 9902 I=1,50
  1               STATIM(I)=0.0
  1               SLAK(I)=0.0
  1               DO 9902 J=1,14
  2          9902 NUNIT(I,J)=0
                  XI=0.123456
                  NCOUNT=1
                  MO=0
                  NO=0
                  READ(5,913) NAME
              913 FORMAT(10A4)
              302 WRITE(6,1002) NAME
             1002 FORMAT(, ,,,ASSEMBLY LINE BALANCING FOR   ,,5X,10A4,//)
                  DO 200 M=1,12
  1               TIME(M)=0.0
  1               NPREC1(M)=0
  1               NPREC2(M)=0
  1               DO 200 N=1,12
  2           200 MATRIX(M,N)=0
                  DO 105 K=1,23,2
  1               READ(5,5) JOBS(K),JOBS(K+1)
  1               IF(JOBS(K+1))105,10,105
              105 MO=MO+2
                5 FORMAT(2I3)
               10 DO 92 I=1,MO,2
                  IF(JOBS(I)) 8,92,8
                8 L=JOBS(I+1)
                  N=JOBS(I)
                  MATRIX(N,L)=1
  1            92 CONTINUE
                  DO 113 K=1,12
  1               READ(5,912) TIME(K)
  1               IF(TIME(K))113,1,113
  1           113 NO=NO+1
              912 FORMAT(F6.3)
```

A-IV/1
(92)

```
        1 NTASK=NO
          BTASK=NO
          DO 98 J=1,NTASK
1         TETIME=0.0
1         DO 5054 I=1,NTASK
2      5054 TETIME=TIME(I)+TETIME
1         NPREC1(J)=0
1         DO 98 I=1,NTASK
2        98 NPREC1(J)=MATRIX(I,J)+NPREC1(J)
         22 READ(5,23) CYCLE,NRUNS
         23 FORMAT(F6.3,3X,I2)
          IF(CYCLE) 24,25,24
         24 M=0
          NO=0
          KBEFOR=100
          NCOUNT=1
          SMALL=0.0
          DO 1001 NX=1,NRUNS
1        41 DO 95 J=1,NTASK
2        95 NPREC2(J)=NPREC1(J)
1         M=0
1         STATIM(I)=0.0
1         K=1
1         WRITE(6,46) NCOUNT
1        46 FORMAT(, ,,, SEQUENCE NUMBER  ,,I3)
1         WRITE(6,26) CYCLE
1        26 FORMAT(, ,,6X,7HSTATION,23X,13HCYCLE TIME    ,F6.3,20X,7HSTATIO
1        14X,5HSLACK,/,6X,6HNUMBER,64X,4HTIME,/)
1         NO=0
1        13 XII=XI* .23
1         IXJ=XII*100.
1         XIJ=IXJ
1         XI=(XII-XIJ/100.)*100.
1         IXI=XI*100.
1         XIK=IXI
1         XKK=XI-(XI-XIK/100.)
1         NUMA=0
1         DO 2031 I=1,NTASK
2         IF(NPREC2(I)) 2031,2032,2031
2      2032 IF(NUMA-20) 5061,2031,2031
2      5061 NUMA=NUMA+1
2         NLIST(NUMA)=I
2      2031 CONTINUE
1         ANUM=NUMA
1         NUMB=(XKK*ANUM)+1.0
1         NRAND=NLIST(NUMB)
1         IF(XI) 2022,2022,45
1      2022 WRITE(6,2021) NRAND
1      2021 FORMAT(, ,,, NRAND,,I5)
1         GO TO 25
1        45 IF(NRAND-NTASK) 12,12,13
1        12 IF(NPREC2(NRAND)) 13,14,13
1        14 STATIM(K)=STATIM(K)+TIME(NRAND)
1         IF(STATIM(K)-CYCLE) 27,27,28
1        27 M=M+1
1         NUNIT(K,M)=NRAND
1         SLAK(K)=CYCLE-STATIM(K)
1         GO TO 29
1        28 STATIM(K)=STATIM(K)-TIME(NRAND)
1         SLAK(K)=CYCLE-STATIM(K)
```

A-IV/3
(94)

```
1              K=K+1
1              STATIM(K)=TIME(NRAND)
1              SLAK(K)=CYCLE-STATIM(K)
1              M=1
1              NUNIT(K,M)=NRAND
1           29 NPREC2(NRAND)=-1
1              DO 91 J=1,NTASK
2           91 NPREC2(J)=NPREC2(J)-MATRIX(NRAND,J)
1              NO=NO+1
1              IF(NO-NTASK) 13,31,13
1           31 NCOUNT=NCOUNT+1
1              DO 93 I=1,K
2           93 WRITE(6,36) I,(NUNIT(I,M),M=1,14),STATIM(I),SLAK(I)
1           36 FORMAT(2I10,13I4,F10.3,F9.3)
1              TSLACK=0.0
1              DO 5055 I=1,K
2         5055 TSLACK=SLAK(I)+TSLACK
1              EFF=(1.0-(TSLACK/TETIME))*100.
1              WRITE(6,5053) TSLACK,EFF
1         5053 FORMAT(1H ,/,5X,18HTOTAL SLACK TIME = ,F6.3,24X,13HEFFICIENCY =
1             1F6.2,9H PER CENT,//)
1              DO 1009 I=1,K
2              DO 1009 J=1,14
3         1009 NUNIT(I,J)=0
1         1001 CONTINUE
               WRITE(6,500)
           500 FORMAT(,1,)
               GO TO 22
            25 STOP
               END

END FTN 417 IBANK 1287 DBANK
ΔXQT
MAP 30R1 S74T11 07/14/81 14:33:39


ADDRESS LIMITS      001000 002026        535 IBANK WORDS DECIMAL
                    040000 045513       2892 DBANK WORDS DECIMAL
STARTING ADDRESS    001166



            SEGMENT $MAIN$            001000 002026      040000 045513

M$PKT$                                            $(2)    040000 040011     20 NOV 78    17:09:58
F2RTRN$                                           $(2)    040012 040013     16 AUG 77    16:22:24
F2ACTIV$/FORFTN           $(1)    001000 001013   $(2)    040014 040016     12 JAN 78    11:01:29
                          $(3)    001014 001027
                          $(5)    001030 001030
F2TABX                                            $(2)    040017 040330     26 JUN 75    13:29:39
F2FCA                                             $(0)    040331 040335     03 JAN 75    07:43:40
FORCOM$/FORFTN                                    $(2)    040336 040343     23 JUL 75    12:16:44
F2CLOSE                   $(1)    001031 001062   $(0)    040344 040346     16 APR 81    22:00:30
CERU$                                                                       08 APR 81    11:56:55
PMD$COM(COMMONBLOCK)                                      040347 040347
MOERO$(COMMONBLOCK)                                       040350 040353
F2CON                                             $(2)    040354 042230     16 APR 81    22:00:44
                                                  $(034)  MOERO$
                                                  $(036)  PMD$COM
```

A-IV/4
(95)

```
F2FRT                                                  $(034)   042231 042272       27 JUL 78   12:58:44
MEMINTERFACE            $(1)    001063 001165          $(2)     042273 042320       20 NOV 78   17:10:37
F2SCT                                                  $(2)     042321 042622       20 NOV 78   17:06:44
F2INIT                                                 $(2)     042623 043002       16 APR 81   22:02:10
F2CDCD$                                                $(2)     043003 043050       19 DEC 74   14:35:02
ERU$/SYS74R1                                                                        20 DEC 78   17:30:56
F2EXIT                                                 $(2)     043051 043104       16 APR 81   22:01:01
F2IOENT                                                                             04 MAY 78   15:34:26
NAME$                   $(1)    001166 002026          $(0)     043105 045162       14 JUL 81   14:33:37
                                                       $(4)     045163 045505
                                                       $(6)     045506 045506
                                                       $(012)   045507 045513
```

COMMON BANKS REFERENCED

0400036 0400003 0400025 0400002 0400001
SYS$*RLIB$. LEVEL
END MAP. ERRORS: 0 TIME: 14.793 STORAGE: 18853/4/040777/074777
ASSEMBLY LINE BALANCING FOR KARAYALCIN ALPASLAN

SEQUENCE NUMBER 1

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 2 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 9 | 5 | 4 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60.000 | .000 |
| 3 | 11 | 3 | 6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 58.000 | 2.000 |

TOTAL SLACK TIME = 8.000 EFFICIENCY = 95.35 PER CENT

SEQUENCE NUMBER 2

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 2 | 10 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 3 | 9 | 5 | 4 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60.000 | .000 |
| 4 | 3 | 11 | 6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 58.000 | 2.000 |

TOTAL SLACK TIME =14.000 EFFICIENCY = 91.86 PER CENT

SEQUENCE NUMBER 3

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 10 | 8 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 3 | 9 | 5 | 4 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60.000 | .000 |
| 4 | 3 | 11 | 6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 58.000 | 2.[illegible]0 |

TOTAL SLACK TIME =14.000 EFFICIENCY = 91.86 PER CENT

SEQUENCE NUMBER 4

CYCLE TIME 60.000

| STATION NUMBER | STATION TIME | SLACK |
|---|---|---|

A-IV/5
(96)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 9 | 1 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 41.000 | 19.000 |
| 3 | 2 | 5 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 59.000 | 1.000 |
| 4 | 7 | 11 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

SEQUENCE NUMBER 5

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 9 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 3 | 10 | 5 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48.000 | 12.000 |
| 4 | 7 | 11 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

SEQUENCE NUMBER 6

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 2 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48.000 | 12.000 |
| 3 | 9 | 5 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 4 | 7 | 11 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

SEQUENCE NUMBER 7

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 2 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48.000 | 12.000 |
| 3 | 9 | 8 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 4 | 7 | 3 | 11 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

SEQUENCE NUMBER 8

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 9 | 1 | 10 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 47.000 | 13.000 |
| 3 | 2 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 53.000 | 7.000 |
| 4 | 7 | 3 | 11 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

A IV/6
(97)

SEQUENCE NUMBER 9

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 2 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48.000 | 12.000 |
| 3 | 9 | 5 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 4 | 7 | 11 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

SEQUENCE NUMBER 10

CYCLE TIME 60.000

| STATION NUMBER | | | | | | | | | | | | | | | STATION TIME | SLACK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54.000 | 6.000 |
| 2 | 1 | 9 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 47.000 | 13.000 |
| 3 | 2 | 10 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 53.000 | 7.000 |
| 4 | 7 | 11 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.000 | 8.000 |
| 5 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20.000 | 40.000 |

TOTAL SLACK TIME =74.000 EFFICIENCY = 56.98 PER CENT

A-IV/7 (98)

```
FTN ERR ON UNIT-5        ATTEMPTED TO READ PAST AN END-OF-FILE
ERR MODE  ERR-TYPE: 03     ERR-CODE: 00
ERROR ADDRESS:  006434     BDI:  500025
ER EABT$      ABORT ADR: 042134  BDI:200005

 X                000000 003075  000000 003076  000000 000017  000000 042241  000000 046117  000000 046073  000000 046061
    000000 042321 000000 046055  777700 045372  000004 001426  400000 000000  020001 001766  000000 000007  000000 000000

 A  400000 000000 020001 001766  000000 000007  000000 000000  000000 017176  000000 000002  000000 000004  777777 775011
    000000 000025 000000 000016  000000 001431  000000 000264  000000 400264  000000 045514  000000 000012  000000 000001
    000000 000000 000000 000000

 R                000000 000000  000000 000000  000000 000204  777777 777776  777777 777776  777777 777776  000000 000003
    000000 000001 000000 000061  000000 000012  000000 045513  000000 000002  000000 000000  000000 000000  000000 042321
 ERR$  TYPE 03 CODE 00 CONT 12      REENT ADR:  041202 BDI: 200005
USER EXECUTED ER ERR$.

 X                000000 003075  000000 003076  000000 000017  000000 042241  000000 046117  000000 046073  000000 046061
    000000 042321 000000 046055  777700 045372  000004 001426  400000 000000  020001 001766  000000 000007  000000 000000

 A  400000 000000 020001 001766  000000 000007  000000 000000  000000 017176  000000 000002  000000 000004  777777 775011
    000000 000025 000000 000016  000000 001431  000000 000264  000000 400264  000000 045514  000000 000012  000000 000001
    000000 000000 000000 000000

 R                000000 000000  000000 000000  000000 000204  777777 777776  777777 777776  777777 777776  000000 000003
    000000 000001 000000 000061  000000 000012  000000 045513  000000 000002  000000 000000  000000 000000  000000 042321
RUNSTREAM ANALYSIS TERMINATED
```

A-IV/8
(99)

```
RUNID: ALCOLE   ACCT: 111-15-216      PROJECT: ALB
    ALCOLE  ABORT
TIME:     TOTAL: 00:00:27.616        CBSUPS:  000001154
          CPU:   00:00:06.462        I/O:  00:00:09.971
          CC/ER: 00:00:11.183        WAIT: 00:00:00.000
SUAS USED:      137.44TL   SUAS REMAINING:       0.00TL
ABOVE CHARGE CALCULATED AT FOLLOWING RATES -
1 CBSUP              = 0.08TL
1 CARD READ          = 0.10TL
1 CARD PUNCHED       = 0.60TL
1 PAGE PRINTED       = 4.00TL
1 TAPE I/O MINUTE    = 6.00TL
1 DISK I/O MINUTE    = 6.00TL
1 TERMINAL MINUTE    = 5.00TL
IMAGES  READ:    160        PAGES:  8
START:     14:33:00 JUL 14,1981    FIN:  14:34:34 JUL 14,1981
```